

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению лабораторных работ
по дисциплине
«Моделирование на основе проблемно-ориентированных языков»
для студентов направления подготовки
02.03.01 Математика и компьютерные науки
направленность (профиль)
«Анализ данных и искусственный интеллект»

Лабораторная работа № 1

Запись арифметических выражений

ЦЕЛЬ РАБОТЫ: закрепление знаний по теоретическим основам алгоритмизации и программирования, приобретение навыков использования арифметических операций, функций и правил записи арифметических выражений согласно синтаксису языка JAVA

Теоретическая часть

Основные понятия алгоритмизации и программирования

1.1. Этапы решения задач на ЭВМ

Задачи, решаемые с помощью ЭВМ, можно классифицировать по различным критериям: по типу информации и информационным технологиям, по способу поиска решения (простые и переборные), по характеру целей (задачи оптимизации, управления, обучения, информационного поиска), по функциональному назначению, а также по уровню достижения цели и уровню их решения, в т. ч. по уровню автоматизации этапов их решения.

Задача становится разрешимой, если найдено правило, способ получения результата. В информатике такое правило называют алгоритмом.

Решение задачи на ЭВМ состоит из нескольких этапов, среди которых основными являются следующие:

1. Постановка задачи:

- сбор информации о задаче;
- формулировка условия задачи;
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин, структуры и т. п.).

2. Формализация (анализ и исследование задачи, модели, представление ее в виде уравнений, соотношений, ограничений и т.п.):

- анализ существующих аналогов задачи;
- анализ технических и программных средств;
- разработка математической модели;
- разработка структур данных.

Понятие моделирования

При решении задачи обычно исследуют не реальный объект, а его модель.

Модель— искусственно созданный объект, обладающий всеми существенными признаками реального объекта, явления или процесса.

Моделирование — это метод познания, состоящий в создании и исследовании моделей.

Цели моделирования:

- 1) понять сущность изучаемого объекта;

- 2) научиться управлять объектом и определять наилучшие способы управления;
- 3) прогнозировать прямые или косвенные последствия;
- 4) решать прикладные задачи.

Математическая модель – это система математических соотношений (данных) – формул, уравнений, неравенств и т.д. и отношений между ними, описывающих поведение объекта с некоторой степенью точности и отражающих существенные свойства моделируемого процесса (объекта или явления).

Порядок составления математической модели:

- выделить реальный объект, на котором будет основываться математическая модель, и из множества его свойств, закономерностей, внутренних связей, отдельных характеристик явления и параметров выделяем те, которые являются существенными для решаемой задачи, и отбросить несущественные;
- определить, что считать исходными данными и результатами;
- подобрать математический объект с тем же числом подобных параметров, отражающий суть реального объекта; записать математические соотношения, связывающие результаты с исходными данными.

3. Выбор метода решения.

4. Разработка алгоритма:

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма (блок-схемы, псевдокод и др.);
- выбор тестов и метода тестирования;
- проектирование алгоритма.

5. Программирование:

- выбор языка программирования;
- уточнение способов организации данных;
- запись алгоритма на выбранном языке программирования.

6. Тестирование, отладка и исправление обнаруженных ошибок:

- синтаксическая отладка;
- отладка семантики и логической структуры;
- тестовые расчёты и анализ результатов тестирования;
- совершенствование программы.

Отладка программы

Отладка программы – это процесс поиска и устранения ошибок в программе, производимый по результатам ее прогона на компьютере.

В современных программных системах отладка осуществляется с использованием специальных программных средств, называемых отладчиками. Эти средства позволяют **исследовать внутреннее поведение программы**.

Программа-отладчик обычно обеспечивает следующие возможности:

- **пошаговое исполнение программы** с остановкой после каждой команды (оператора);
- **просмотр текущего значения любой переменной или нахождение значения любого выражения**, в том числе, с использованием стандартных функций; при необходимости можно установить новое значение переменной;
- **установку в программе "контрольных точек"**, т.е. точек, в которых программа временно прекращает свое выполнение, так что можно оценить промежуточные результаты, и др.

При отладке программ **важно помнить следующее:**

- в начале процесса отладки надо использовать **простые тестовые данные**;
- возникающие затруднения следует четко **разделять и устранять строго поочередно**;
- **не нужно считать причиной ошибок машину**, так как современные машины и трансляторы обладают чрезвычайно высокой надежностью.

Тест и тестирование программы

Тестирование – это испытание, проверка правильности работы программы в целом либо ее составных частей.

При отладке происходит локализация и устранение синтаксических ошибок и явных ошибок кодирования, в процессе же тестирования проверяется работоспособность программы, не содержащей явных ошибок. Тестирование устанавливает факт наличия ошибок, а отладка выясняет причину неправильной работы программы.

Как бы ни была тщательно отлажена программа, решающим этапом, устанавливающим ее пригодность для работы, является контроль программы по результатам ее выполнения на системе тестов.

Программу условно можно считать правильной, если её запуск для выбранной системы тестовых исходных данных во всех случаях дает правильные результаты. Но, как справедливо указывал известный теоретик программирования Э. Дейкстра, тестирование может показать лишь наличие ошибок, но не их отсутствие. Нередки случаи, когда новые входные данные вызывают "отказ" или получение неверных результатов работы программы, которая считалась полностью отлаженной.

Для реализации метода тестов должны быть изготовлены или заранее известны эталонные результаты. Вычислять эталонные результаты нужно обязательно до, а не после получения машинных результатов. В противном случае име-

ется опасность невольной подгонки вычисляемых значений под желаемые, полученные ранее на машине.

7. Анализ результатов решения задачи и уточнение математической модели с повторным выполнением этапов 2 - 6 (при необходимости).

8. Сопровождение программы: это работы, связанные с обслуживанием программ в процессе их эксплуатации:

- доработка программы для решения конкретных задач;
- составление документации к решённой задаче, к математической модели, к алгоритму, к программе, к набору тестов, к использованию программы.

1.2. Основы алгоритмизации

Алгоритм – это метод (способ) решения задачи, записанный по определённым правилам, в виде конечной последовательности однозначных предписаний, исполнение которых позволяет с помощью конечного числа шагов получить решение задачи, однозначно определяемое исходными данными из некоторого множества значений.

Свойства алгоритма

1. **Дискретность** (прерывность, раздельность). Алгоритм должен представлять процесс решения задачи как последовательное выполнение конечного числа простых (или ранее определенных) законченных действий шагов.
2. **Понятность** для исполнителя — т.е. исполнитель алгоритма должен знать, как его выполнять.
3. **Определенность** (точность, детерминированность). Каждое правило алгоритма должно быть четким и однозначным, содержать действия над известными данными. Каждое действие должно быть понятно исполнителю (для каждого алгоритма предполагается конкретный исполнитель).

Замечание. Часто под свойством детерминированности алгоритма понимается одновременное выполнение свойств точности и понятности.

4. **Результативность** (или конечность). Алгоритм должен приводить к решению задачи, получение определенного результата за конечное число шагов.
5. **Правильность.** Способность алгоритма обеспечить получение именно того результата, который требуется. Неправильность может объясняться неполнотой наших представлений о свойствах объекта или упущением в решении. Для доказательства правильности алгоритма задача часто делится на блоки и правильность доказывается для каждого блока, хотя такая проверка не является полной.
6. **Массовость.** Алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.
7. **Универсальность.** Алгоритм должен быть составлен так, чтобы им мог воспользоваться любой исполнитель для решения аналогичной задачи. (Например, правила сложения и умножения чисел годятся для любых чисел, а не для каких-то конкретных.)

8. **Эффективность.** Выбор алгоритмы, который будет выполнен за минимальное время, с минимальными затратами ресурсов.

Таким образом, исполнитель действует **формально**, т.е. отвлекается от содержания поставленной задачи, а только строго выполняет некоторые правила, инструкции и вместе с тем получать нужный результат.

Критерии качества алгоритма

1. **Связанность.** Определяется количеством промежуточных результатов. Чем выше количество промежуточных результатов, тем ниже связанность.
2. **Объем алгоритма.** Это количество операций или шагов, которые необходимо выполнить, а также сложность этих шагов.
3. **Логическая сложность.** Определяется количеством ветвлений и циклов.

Порядок выполнения алгоритма

1. Действия в алгоритме выполняются в порядке их записи.
2. Нельзя менять местами никакие два действия алгоритма.
3. Нельзя не закончив одного действия переходить к следующему.

Способы описания алгоритмов

1. **Словесно-формульный.** Описание алгоритма с помощью слов и формул на естественном языке.

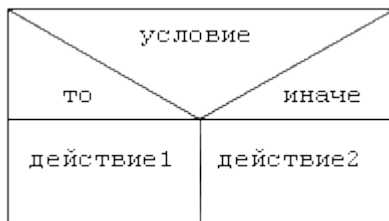
Словесный способ не имеет широкого распространения по следующим причинам:

- такие описания строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний. «Он встретил ее на поле с цветами».

Пример. Составить алгоритм начисления зарплаты согласно следующему правилу: если стаж работы сотрудника менее 5 лет, то зарплата 130 руб., при стаже работы от 5 до 15 лет – 180 руб., при стаже свыше 15 лет зарплата повышается с каждым годом на 10 руб.

Словесно-формульное описание алгоритма решения задачи:

1. Ввести ST , перейти к п. 2.
 2. Если $ST < 5$, то $ZP := 130$, перейти к п. 4, иначе — перейти к п. 3.
 3. Если $ST < 15$, то $ZP := 180$, перейти к п. 4, иначе $ZP := 180 + (ST - 15)10$, перейти к п. 4.
 4. Вывести (отпечатать) значение ZP , перейти к п. 5.
 5. Вычисления прекратить.
2. **Табличный.** Алгоритм представляется в форме таблицы и расчётных формул (физика, химия и т. д.).
 3. **Структурограмма**



4. Синтаксическая диаграмма(формулы Бэкуса-Наура)



5. **Псевдокоды.** Полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.

Псевдокод занимает промежуточное место между естественным и формальными языками. С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд. Однако в псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда.

Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.

Примером псевдокода является школьный алгоритмический язык.

Алгоритмический язык — это средство для записи алгоритмов в аналитическом виде, промежуточном между записью алгоритма на естественном (человеческом) языке и записью на языке ЭВМ (языке программирования).

Запись алгоритма решения задачи примера 1 на алгоритмическом языке:

```

алг зарплата(цел. ST, вещ ZP)
  арг ST
  рез ZP
нач
  если ST < 5
    то ZP := 150
    иначе
      если ST <= 15
        то ZP := 180
        иначе ZP := 180 + (ST - 15) * 10
      все
  все
кон
  
```

6. **Программный.** Описание алгоритма с помощью языков программирования.
7. **Графический.** Алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Такое графическое представление называется схемой алгоритма или **блок-схемой**.

Блок-схема алгоритма представляет собой систему связанных геометрических фигур.

Правила построения блок-схем

1. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Для наглядности операции разного вида изображаются в схеме различными геометрическими фигурами.
2. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий. Порядок выполнения действий указывается стрелками, соединяющими блоки.
3. В схеме блоки стараются размещать сверху вниз, в порядке их выполнения.
4. Все повороты соединительных линий выполняются под углом 90 градусов.

В таблице приведены наиболее часто употребляемые символы.

Название	Обозначение и пример заполнения	Выполняемая функция (пояснение)
Начало/конец (вход/выход)		Начало или конец программы, вход или выход в подпрограмму
Блоки ввода/вывода		Ввод-вывод данных
		Вывод данных на печатающее устройство
Блок вычислений		Арифметический блок определяет вычислительное действие или последовательность действий
Логический блок		Логический блок проверяет истинность или ложность условия и выбирает направления выполнения алгоритма в зависимости от условия. В блоке должны быть указаны вопрос, условие или сравнение, которые он определяет.
Предопределенный процесс		Вычисления по стандартной или пользовательской подпрограмме
Блок модификации		Выполнение действий, изменяющих пункты алгоритма, начало цикла. Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра.

		ра для каждого повторения.
Межстраничный соединитель		Указание связи между частями схемы, расположенной на разных страницах

Общие правила построения схемы алгоритма задачи

1. Выявить исходные данные, результаты, назначить им имена.
2. Выбрать метод (порядок) решения задачи.
3. Разбить метод решения задачи на этапы (с учетом возможностей ЭВМ).
4. Изобразить каждый этап в виде соответствующего блока схемы алгоритма и указать стрелками порядок их выполнения.
5. В полученной схеме при любом варианте вычислений:
 - а) предусмотреть выдачу результатов или сообщений об их отсутствии;
 - б) обеспечить возможность после выполнения любой операции, так или иначе, перейти к блоку *Останов* (к выходу схемы).

Эти правила и есть «**Основные принципы алгоритмизации**». Будем считать, что знание и применение настоящих «принципов» обязательно при составлении алгоритма любой задачи.

Типы алгоритмов

- структурированные;
- неструктурированные (т.е. с нарушением структуры – с операторами безусловного перехода);
- вспомогательные (используемые в составе других алгоритмов).

Виды алгоритмов

- линейный алгоритм;
- алгоритм ветвления;
- циклический алгоритм;
- алгоритм с подпрограммами;
- смешанные (т.е. содержащие и циклы, и ветвление, и функции).
- рекурсивный алгоритм обращается к самому себе, пока не выполнится определенное условие.

Базовые алгоритмические конструкции

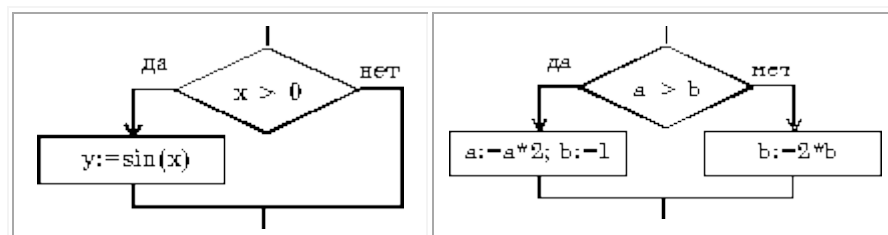


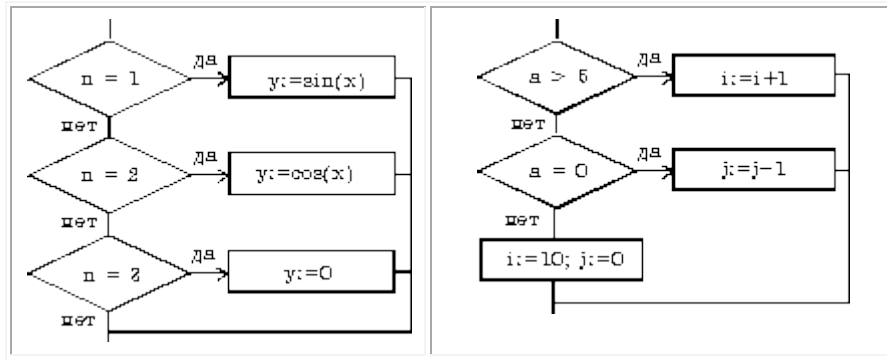
2. **Базовая структура ветвление (алгоритм ветвления)** обеспечивает в зависимости от результата проверки условия выбор одного из альтернативных путей работы алгоритма. Условие – вопрос, имеющий два варианта ответа: **да** или **нет**. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Запись ветвления выполняется в двух формах: полной и неполной.

Структура **ветвление** существует в четырех основных вариантах:

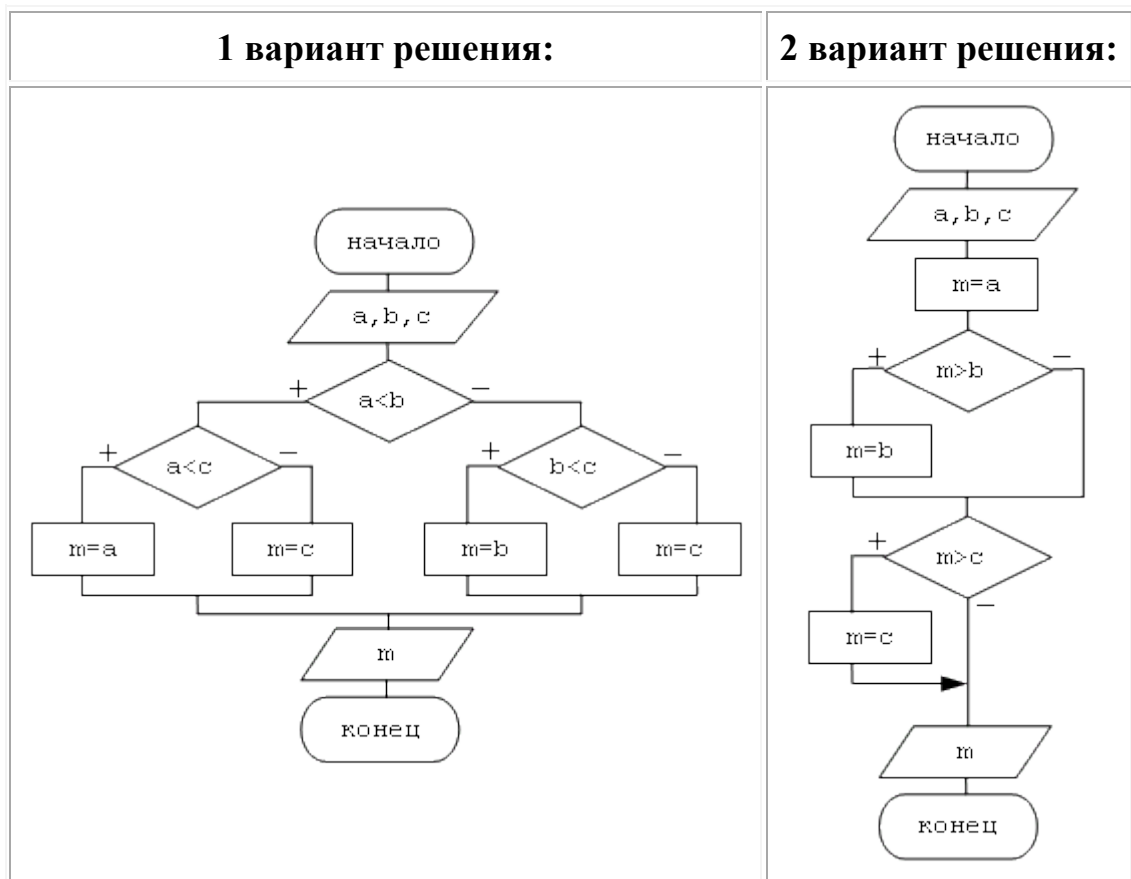
Неполная форма записи		Полная форма записи	
1. если-то		2. если-то-иначе	
3. выбор		4. выбор-иначе	

Примеры команды *если*

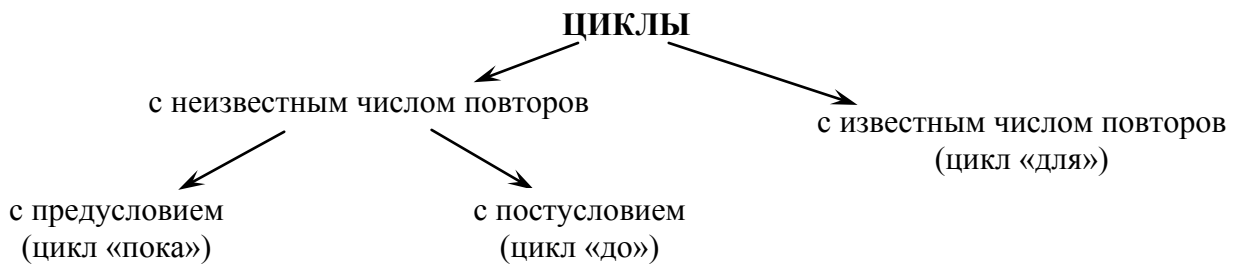




Пример: найти наименьшее из трех чисел.

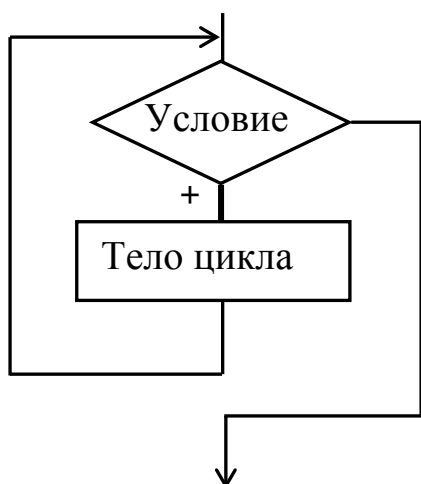


3. Базовая структура **цикл** обеспечивает многократное выполнение некоторой совокупности действий, которая называется **телом цикла**, над новыми данными.



Цикл типа «пока» (Цикл с предусловием)

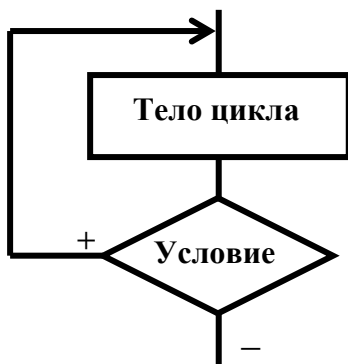
Выполнение цикла «пока» начинается с проверки условия, поэтому такую разновидность циклов называют циклами с **предусловием**.



Переход к выполнению действия осуществляется только в том случае, если условие выполняется, иначе происходит выход из цикла. Тело цикла выполняется до тех пор, пока выполняется условие. Условие цикла необходимо подобрать так, чтобы действия, выполняемые в цикле, привели к нарушению его истинности, иначе произойдет заикливание.

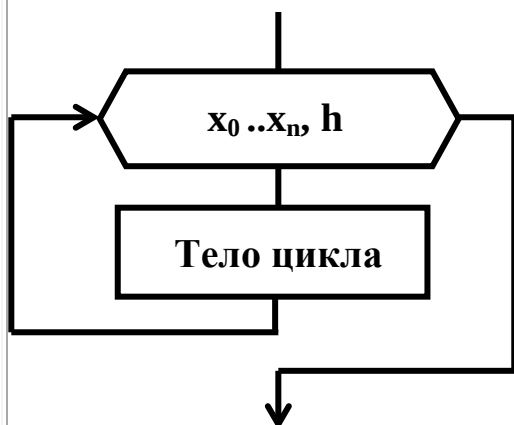
Заикливание - бесконечное повторение выполняемых действий.

Цикл типа «до» (Цикл с постусловием)



Исполнение цикла начинается с выполнения действия. Таким образом, тело цикла будет реализовано хотя бы один раз. После этого происходит проверка условия. Поэтому цикл "до" называют циклом с **постусловием**. Если условие выполняется, то происходит возврат к выполнению действий, иначе осуществляется выход из цикла. Для предотвращения заикливания необходимо предусмотреть действия, приводящие к ложности условия.

Цикл типа «для»



Цикл с параметром, или цикл со счетчиком, или арифметический цикл - это цикл с заранее известным числом повторов. Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.

x_0 – начальное значение параметра; h – шаг; x_n – последнее значение параметра.

Для создания циклов с параметром необходимо использовать правила:

1. Параметр цикла, его начальное и конечное значения и шаг должны быть одного типа.
2. Запрещено изменять в теле цикла начальное, текущее и конечное значения для параметра.
3. Запрещено входить в цикл, минуя блок модификации.
4. После выхода из цикла значение переменной параметра неопределенно и не может использоваться в дальнейших вычислениях.

5. Из цикла можно выйти, не закончив его, тогда переменная параметр сохраняет свое последнее значение.

1.3. Теоретические основы программирования

Программирование – это раздел информатики, изучающий методы и приемы составления программ для компьютеров. Кроме того, программирование – это подготовка задачи к решению ее на компьютере.

Программа – последовательность действий, которые должен выполнить компьютер в строго указанной очередности для достижения конкретного результата.

Языки программирования – это совокупность средств и правил представления алгоритма в виде, приемлемом для компьютера.

Системы программирования – это набор средств ввода, редактирования, трансляции и выполнения программ на ЭВМ.

Транслятор – это устройство или комплекс программ, обеспечивающий перевод программы, написанной на символическом языке, в совокупность машинных команд, либо передающие/преобразующие данные или другую программу.

Например, транслятор воспринимает операторы одного языка и вырабатывает соответствующие операторы другого языка.

Компилятор – это транслятор, обеспечивающий перевод программы, написанной на алгоритмическом языке, в совокупность машинных команд без ее выполнения в компьютере.

Компилятор оценивает исходный текст в соответствии с синтаксической конструкцией языка и переводит на машинный язык.

Например, компилятор берет программу, написанную на языке C, и преобразует ее в программу на языке ассемблера.

Интерпретатор – это транслятор, обеспечивающий перевод каждой конструкции алгоритмического языка в машинные команды и одновременное выполнение этой конструкции в компьютере.

При исполнении программных операторов, интерпретатор должен сначала сканировать каждый оператор с целью прочтения его содержимого, а затем выполнить запрошенную операцию.

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: записать следующие выражения по правилам языка
JAVA

Задание I

1. $\sin 5x + 2 \operatorname{tg} x^2$	5. $\frac{2xy + 4x^2}{5y};$
2. $2x + 3y^2 - \sin 3x$	6. $4\sqrt{\cos 3b - 2\sin b};$
3. $\sqrt{\frac{\sin y^2}{2 \cos 2y}} + 2 \operatorname{tgy} - 25 $	7. $-2 \cos x^2 + 2y^7;$
4. $\sqrt{1 - \frac{2}{x^2}} + x^5 - 3x $	8. $\frac{ 2a + \operatorname{tg} 3a }{\sqrt{1 + b^8} + 11}$

Задание II

Выполнить согласно варианту:

1. $a = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y},$	$b = 1 + \frac{z^2}{3 + \frac{z^2}{5}}.$
2. $s = \left x^{\frac{y}{x}} - 3\sqrt{\frac{y}{x}} \right + \ln x,$	$t = (y - x) \frac{y - z(y - x)}{1 + (y - x)^2}.$
3. $s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!},$	$t = x(\sin x^3 + \cos^2 y).$
4. $y = e^{-bt} \sin(at + b) - \sqrt{bt + a},$	$s = b \sin(at^2 \cos 2t) - 1.$
5. $w = \sqrt{x^2 + b} - \frac{b^2 \sin^3(x + a)}{x},$	$y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}.$
6. $s = x^3 \operatorname{tg}^2(x + b)^2 + \frac{\alpha}{\sqrt{x + b}},$	$d = \frac{bx^2 - a}{e^{ax} - 1}.$
7. $z = \frac{x^2(x + 1)}{b} - \sin^2(x + a),$	$s = \sqrt{\frac{xb}{a}} - \cos^2(x + b)^3.$

8.	$y = \sin^3(x^2 + a)^2 - \sqrt{\frac{x}{b}},$	$z = \frac{x^2}{a} + \cos(x + b)^3.$
9.	$f = \sqrt[3]{m \cdot \operatorname{tg} t + c \cdot \sin t },$	$z = m \cos(bt \cdot \sin t) + c.$
10.	$y = b \operatorname{tg}^2 x - \frac{a}{\sin\left(\frac{x}{a}\right)},$	$d = a \cdot e^{-\sqrt{a}} \cdot \cos\left(\frac{bx}{a}\right).$
11.	$f = \ln(a + x^2) + \sin^2\left(\frac{x}{b}\right),$	$z = e^{-cx} \cdot \frac{x + \sqrt{x + a}}{x - \sqrt{ x - b }}.$
12.	$y = \frac{a^{2x} + b^{-x} \cdot \cos(a + b)x}{x + 1},$	$k = e^{\frac{ax}{\sqrt{2}}} \cdot \cos \sqrt{\frac{bx}{2}}.$
13.	$z = \sqrt{ax \cdot \sin 2x + e^{-2x} \cdot (x + b)},$	$w = e^3 \cdot \sin bx - \frac{x^3}{a}.$
14.	$u = \frac{a^2 x + e^{-x} \cdot \cos bx}{bx - e^{-x} \cdot \sin bx + 1},$	$f = e^{2x} \cdot \ln(a + x) - b^{3x} \cdot \ln(b - x).$
15.	$z = \frac{\sin x}{\sqrt{1 + m^2 \sin^2 x}} - cm \cdot \ln(mx),$	$s = e^{-ax} \cdot \sqrt{x + 1} + e^{-bx} \cdot \sqrt{x + 1.5}.$
16.	$a = e^{-mx} \cdot \frac{\sqrt{m + x^2}}{ m^2 - x^2 },$	$b = \ln(c^2 + \sqrt{c + y}) + e^{-cy}.$
17.	$z = x^3 \cdot \sin^2(x + b) + \frac{x}{(x + b^2)},$	$s = (m^2 + y^2) \cdot e^{my} + \sqrt{3m + \ln(m + 4y)}.$
18.	$l = \frac{a \cos^3 bx}{\sin a + bx \cos b - e^a},$	$w = \frac{\sin(x^3 - a) + b x }{\cos(ax + b)}.$
19.	$y = \cos^2(x + a^2)^2 + \sqrt{\frac{x}{b^2}},$	$z = \frac{a^3}{3x^2} + e^{(x+b)^3}.$
20.	$s = \left x^{\frac{x}{y}} - 5\sqrt{\frac{x}{y}} \right ,$	$t = (x - y) \frac{\sin(x + y) - \operatorname{tg}(y - x) \cdot z}{1 + (x - y)^2}$

21.	$s = \frac{e^{my}}{2 + 5x} + \sqrt{\frac{\ln(m-3)}{ 2a }},$	$v = \sqrt{ bt - a } - e^{\sqrt{bt}} \cdot \sin(bt + a).$
22.	$y = 1 + \frac{b \sin(at^2 \cos 3t)}{12x},$	$f = \sqrt[3]{m \cdot \operatorname{tg} t + c \sin 3t }$
23.	$y = a \cdot \operatorname{tg}^2(x + a) - \frac{b}{\sin^2\left(\frac{a}{x}\right)},$	$d = b \cdot e^{-\sqrt{b}} \cdot \cos\left(\frac{ax}{b} + 1.4\right).$
24.	$z = \sqrt{\frac{x^2}{b}} - \ln(a^2 + x^2),$	$f = e^{-cx} \cdot \frac{x + \sqrt{ x - b }}{x - \sqrt{x + a}}.$
25.	$s = \frac{(m^2 + y^2)}{5a} + \sqrt{ m + \operatorname{tg}(m - 2y)},$	$f = \frac{e^{2x}}{6 \cdot \ln(a + x)} - b^{3x} \cdot \sin(b - x).$

Контрольные вопросы

1. Что такое алгоритм?
2. Перечислите свойства алгоритмов.
3. Способы записи алгоритма.
4. Перечислите базовые алгоритмические конструкции. Какова блок-схема следования?
5. Какова блок-схема полной (неполной) формы команды ветвления?
6. Какова блок-схема цикла с предусловием?
7. Какова блок-схема цикла с постусловием?
8. Какова блок-схема цикла с параметром?
9. Что такое программа?
10. Что такое транслятор?
11. Что такое компилятор?
12. Что такое интерпретатор?
13. Что такое оператор и операнд?
14. В каких случаях при записи математических выражений на языке JAVA используются круглые скобки?
15. Как используются стандартные математические функции в JAVA?

Лабораторная работа № 2

Программирование алгоритмов линейной структуры

ЦЕЛЬ РАБОТЫ: закрепление знаний о типах данных, преобразованиях типов, приобретение навыков составления и отладки программ линейной структуры на языке программирования JAVA

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: в соответствии с вариантом составить и реализовать программы.

Задание I

Написать программу для расчета по двум формулам. Предварительно подготовить тестовые примеры для второй формулы с помощью калькулятора (результаты вычисления по обеим формулам должны совпадать). Список стандартных математических функций JAVA приведен в приложении 2. Отсутствующие в языке функции выразить через имеющиеся.

1.	$z_1 = 2 \sin^2(3p - 2\alpha) \cos^2(5p + 2\alpha);$	$z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}p - 8\alpha\right).$
2.	$z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha;$	$z_2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{p}{4} + 2\alpha\right).$
3.	$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha};$	$z_2 = 2 \sin \alpha.$
4.	$z_1 = \frac{\sin \alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha};$	$z_2 = \operatorname{tg} 3\alpha.$
5.	$z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha;$	$z_2 = \cos^2 \alpha + \cos^4 \alpha.$
6.	$z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha;$	$z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5}{2}\alpha \cdot \cos 4\alpha.$
7.	$z_1 = \cos^2\left(\frac{3}{8}p - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8}p + \frac{\alpha}{4}\right);$	$z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}.$
8.	$z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1;$	$z_2 = \sin(y + x) \cdot \sin(y - x).$

9.	$z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2;$	$z_2 = -4 \sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta).$
10.	$z_1 = \frac{\sin\left(\frac{p}{2} + 3\alpha\right)}{1 - \sin(3\alpha - p)};$	$z_2 = \operatorname{ctg}\left(\frac{5}{4}p + \frac{3}{2}\alpha\right).$
11.	$z_1 = \frac{1 - 2 \sin^2 \alpha}{1 + \sin 2\alpha};$	$z_2 = \frac{1 - \operatorname{tg} \alpha}{1 + \operatorname{tg} \alpha}.$
12.	$z_1 = \frac{\sin 4\alpha}{1 + \cos 4\alpha} \cdot \frac{\cos 2\alpha}{1 + \cos 2\alpha};$	$z_2 = \operatorname{ctg}\left(\frac{3}{2}p - \alpha\right).$
13.	$z_1 = \frac{\sin \alpha + \cos(2\beta - \alpha)}{\cos \alpha - \sin(2\beta - \alpha)};$	$z_2 = \frac{1 + \sin 2\beta}{\cos 2\beta}.$
14.	$z_1 = \frac{\cos \alpha + \sin \alpha}{\cos \alpha - \sin \alpha};$	$z_2 = \operatorname{tg} 2\alpha + \sec 2\alpha \cdot \left(\sec x = \frac{1}{\cos x}\right).$
15.	$z_1 = \frac{\sqrt{2b + 2\sqrt{b^2 - 4}}}{\sqrt{b^2 - 4} + b + 2};$	$z_2 = \frac{1}{\sqrt{b + 2}}.$
16.	$z_1 = \frac{x^2 + 2x - 3 + (x + 1)\sqrt{x^2 - 9}}{x^2 - 2x - 3 + (x - 1)\sqrt{x^2 - 9}};$	$z_2 = \sqrt{\frac{x + 3}{x - 3}}.$
17.	$z_1 = \frac{\sqrt{(3m + 2)^2 - 24m}}{3\sqrt{m} - \frac{2}{\sqrt{m}}};$	$z_2 = \sqrt{m} \cdot (m > 0)$
18.	$z_1 = -2 - \left(\frac{a + 2}{\sqrt{2a}} - \frac{a}{\sqrt{2a} + 2} + \frac{2}{a - \sqrt{2a}}\right) \cdot \frac{\sqrt{a} - \sqrt{2}}{a + 2};$	$z_2 = \frac{1}{\sqrt{a} - \sqrt{2}} \cdot (a > 0)$
19.	$z_1 = \left(\frac{1 + a + a^2}{2a + a^2} + 2 - \frac{1 - a + a^2}{2a - a^2}\right)^{-1} (5 - 2a^2);$	$z_2 = \frac{4 - a^2}{2}.$
20.	$z_1 = \frac{(m - 1)\sqrt{m} - (n - 1)\sqrt{n}}{\sqrt{m^3 n + nm + m^2 - m}};$	$z_2 = \frac{\sqrt{m} - \sqrt{n}}{m}.$

21. $z_1 = (1 + \operatorname{tg} \alpha)(1 + \operatorname{ctg} \alpha) - 2;$	$z_2 = \frac{1}{\sin \alpha \cos \alpha}.$
22. $z_1 = 1 + \operatorname{ctg}^2 \alpha + \frac{1}{\cos^2 \alpha};$	$z_2 = \frac{1}{\sin^2 \alpha \cos^2 \alpha}.$
23. $z_1 = 1 - \frac{1}{1 + \operatorname{ctg}^2 \alpha};$	$z_2 = \frac{1}{1 + \operatorname{tg}^2 \alpha}.$
24. $z_1 = \frac{1 - 2 \cos^2 \alpha}{\sin \alpha \cos \alpha} + \operatorname{ctg} \alpha;$	$z_2 = \operatorname{tg} \alpha.$
25. $z_1 = \sin^4 \alpha - 1 + 2 \sin^2 \alpha \cos^2 \alpha;$	$z_2 = -\cos^4 \alpha.$

Задание II

1. Составить программу вычисления объема конуса по заданному диаметру и образующей.
2. Дана сторона равностороннего треугольника. Составить программу нахождения площади этого треугольника и радиуса описанной окружности.
3. Известны радиусы двух концентрических окружностей. Составить программу нахождения площади кольца, образованного этими окружностями.
4. Составить программу нахождения суммы n членов арифметической прогрессии, для которой известен первый член, разность и число n .
5. Вычислить объем призмы, боковые грани которой - квадраты, а основанием служит равносторонний треугольник, вписанный в круг радиуса r .
6. Вычислить площадь прямоугольника, вписанного в окружность радиуса r , если отношение его сторон равно R .
7. Даны две стороны треугольника и угол между ними. Определить третью сторону и площадь.
8. Вычислить процент материала, ушедшего в отходы, если из куба с ребром a был выточен шар радиуса r ($r < a$).
9. Вычислить площадь кольца, ширина которого равна a , а отношение радиусов окружностей равно b .
10. Вычислить периметр и площадь прямоугольного треугольника, описанного около круга радиуса r , если гипотенуза треугольника равна c .
11. Вычислить диаметр трубы, пропускная способность которой позволяет заменить ею две трубы с диаметрами d_1 и d_2 .
12. Вычислить высоты треугольника со сторонами a , b и c .
13. Вычислить площадь правильного n -угольника, описанного около круга радиусом r .

14. Высота конуса равна h , а радиус основания r . Вычислить объем шара, вписанного в конус.
15. Вычислить массу свинцовой трубы, длина которой равна b м (плотность свинца равна $11,4 \text{ г/см}^3$), толщина стенок a мм, а внутренний диаметр трубы равен d мм.
16. Дан равносторонний треугольник. Вычислить сторону, высоту и площадь этого треугольника, если радиус вписанной окружности равен r .
17. Стальной вал, имеющий b мм длины и d мм в диаметре, обрабатывается на токарном станке, при этом диаметр уменьшается при обточке на S мм. Вычислить, на сколько уменьшается масса тела (плотность стали $7,4 \text{ г/см}^3$).
18. Вычислить объем призмы, боковые грани которой – квадраты. Основанием призмы служит равносторонний треугольник, заданный длиной стороны.
19. В конус с радиусом основания r и высотой h вписан цилиндр, радиус основания которого равен a . Вычислить объем цилиндра.
20. Даны две стороны треугольника d и c и угол между ними α . Найти радиусы окружностей: вписанной в треугольник и описанной около него.
21. Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.
22. Треугольник задан тремя сторонами. Вычислить его высоты.
23. Определить периметр правильного n -угольника, описанного около окружности радиуса R .
24. Смешано V_1 литров воды температуры T_1 и V_2 литрами воды температуры T_2 . Найти объем и температуру образовавшейся смеси.
25. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
26. Даны гипотенуза и катет прямоугольного треугольника. Найти второй катет и радиус вписанной окружности.
27. Найти площадь кольца, внутренний радиус которого равен R_1 , а внешний - R_2 ($R_2 > R_1$).
28. Определить время, через которое встретятся два тела, равноускоренно движущиеся навстречу друг к другу, если известны их начальные скорости, ускорение, начальное расстояние между ними.
29. Вычислить расстояние между двумя точками с координатами (x_1, y_1) и (x_2, y_2) .
30. Даны катеты прямоугольного треугольника. Найти его гипотенузу и площадь.

Задание III

1. Радоновые ванны, применяемые для лечения, содержат $1,8 \cdot 10^6$ атомов радона в воде объемом $1,0 \text{ дм}^3$. На сколько молекул воды приходится один атом радона в лечебной ванне?

2. Сколько атомов ртути содержится в воздухе объемом 1.0 м^3 в помещении, зараженном ртутью, при температуре 20°C , если давление насыщенного пара ртути при этой температуре 133 мПа ?
3. Какова длина ребра куба, содержащего $1.0 \cdot 10^6$ молекул идеального газа при нормальных условиях?
4. При какой температуре молекулы гелия имеют такую же среднюю скорость, как молекулы кислорода при 23°C ?
5. Найдите среднюю кинетическую энергию поступательного движения молекул водорода, если при давлении 0.5 атм. их концентрация равна $1.5 \cdot 10^9 \text{ м}^{-3}$.
6. Газ нагревается в открытом сосуде при нормальном атмосферном давлении от 27°C до 327°C . Какое приращение получает при этом число молекул в единице объема газа?
7. В сосуде объемом 1 дм^3 содержится некоторый газ при температуре 17°C . Найдите приращение давления газа, если вследствие утечки газа из него выйдет 10^{21} молекул.
8. В сосуде объемом 3.0 дм^3 находится гелий массой 4.0 мг , азот массой 70 мг и $5.0 \cdot 10^{21}$ молекул водорода. Каково давление смеси, если температура ее 27°C ?
9. Определить среднюю кинетическую энергию вращательного движения молекул водорода, содержащихся в 1.0 моль при 18°C .
10. При какой температуре молекулы кислорода имеют такую же среднюю скорость, как молекулы водорода при 25°C ?
11. Сколько молекул содержится при нормальных условиях в 1 м^3 воздуха?
12. В сосуде объемом 2.0 дм^3 находится газ под давлением 0.50 МПа . Чему равна средняя кинетическая энергия поступательного движения молекул газа?
13. Кислород массой 12 г находится при температуре 700°C , при этом 40% молекул диссоциировано на атомы. Чему равна средняя кинетическая энергия теплового движения частиц? Колебательные степени свободы молекул кислорода не возбуждаются.
14. Объём помещения 50 м^3 . температура воздуха зимой 0°C , а летом – 40°C . Какова разница в массе воздуха, заполняющего помещение зимой и летом? Принять $\mu = 29 \text{ кг/моль}$.
15. Плотность воздуха при нормальных условиях 1.3 г/л . Какова плотность воздуха при температуре 100°C и давлении $4.0 \cdot 10^5 \text{ Н/м}^2$?
16. Температура воздуха в баллоне объёмом 10 м^3 при давлении 700 мм рт. ст. была 15°C . После нагрева воздуха до 20°C часть воздуха была вытеснена. Найдите массу вытесненного воздуха.
17. Сколько частиц (атомов и молекул) находится в азоте массой 1.0 г , если степень диссоциации азота 7.0% .
18. Какое давление на стенки сосуда производит кислород, если средняя квадратичная скорость его молекул 400 м/с и концентрация молекул $2.7 \cdot 10^{19} \text{ м}^{-3}$?

19. Найдите среднюю кинетическую энергию поступательного движения молекул гелия, если при давлении 0.5 атм. их концентрация равна $1.5 \cdot 10^9 \text{ м}^{-3}$.
20. Определите температуру газа, если средняя кинетическая энергия поступательного движения его молекул равна $1.6 \cdot 10^{-19}$ Дж.
21. Каково давление газа, если в каждом кубическом сантиметре его содержится $1.0 \cdot 10^6$ молекул, а температура газа 87°C ?
22. Газ нагревается в открытом сосуде при нормальном атмосферном давлении от 22°C до 320°C . Какое приращение получает при этом число молекул в единице объема газа?
23. В сосуде объемом 15.0 дм^3 находится газ под давлением 0.50 МПа. Чему равна средняя кинетическая энергия поступательного движения молекул газа?
24. При какой температуре молекулы гелия имеют такую же среднюю скорость, как молекулы водорода при 27°C ?
25. Газ занимает объём 2 л при давлении $5 \cdot 10^5 \text{ Н/м}^2$. Определите суммарную энергию поступательного движения молекул газа.

Задания для самостоятельной работы

1. Даны два действительных числа. Найти среднее арифметическое и среднее геометрическое этих чисел.
2. Даны четыре целых числа. Найти среднее арифметическое этих чисел и среднее геометрическое этих чисел.
3. Треугольник задан длинами своих сторон. Используя формулу Герона, найти площадь этого треугольника.
4. Идет k -я секунда суток. Определить сколько полных часов H , полных минут M прошло к этому моменту.
5. Даны два момента времени одних суток: H_1, H_2 - часы, M_1, M_2 - минуты. Определить интервал между этими моментами в часах H и минутах M .
6. По трем координатам вершин некоторого треугольника найти его высоту.
7. Найти объем и площадь боковой поверхности конуса радиусом r , высотой h и образующей l .
8. Вычислить боковую поверхность пирамиды (пирамида правильная, усеченная), если известны p и p_1 - полупериметры нижнего и верхнего оснований и k - апофема.
9. Найти объем и полную поверхность полого шара, если известны r_1 и r_2 - радиусы внешней и внутренней шаровых поверхностей.
10. Вычислить площадь поверхности и объем усеченного конуса высотой h , радиусами r_1 и r_2 , образующей l .
11. В шар радиуса r вписан конус с углом α при вершине в осевом сечении конуса. Определить объем и полную поверхность конуса.

12. Вычислить объем цилиндра, вписанного в правильную шестиугольную призму, у которой каждое ребро равно a .
13. Найти площадь равнобокой трапеции с основаниями A и B и углом α при большем основании A .
14. Треугольник задан величинами своих углов и радиусом описанной окружности. Найти, стороны треугольника.
15. Найти объем правильной треугольной пирамиды, если стороны ее основания равны a и плоский угол при ее вершине равен α .
16. Вычислить радиус шара, вписанного в правильную четырехугольную пирамиду, если стороны основания пирамиды равны a и двугранный угол при основании равен α .
17. Вычислить площадь круга и равнобедренной трапеции, описанной около него, если периметр трапеции равен p , а угол при нижнем основании равен α .
18. Вычислить объем призмы, боковые грани которой – квадраты. Основанием призмы служит равносторонний треугольник, по который известно, что радиус вписанной в него окружности равен r .

Контрольные вопросы

1. Что такое программа на JAVA?
2. Какая программа является линейной?
3. Какова структура программы на JAVA?
4. Что такое идентификатор?
5. По каким правилам создаются идентификаторы в JAVA?
6. Что такое тип величины?
7. Какие типы величин используются в языке JAVA?
8. Какие типы величин относятся к простым типам?
9. В каком диапазоне могут изменяться переменные типа **int** и **float**?
10. Как описываются переменные в JAVA? (Привести пример.)
11. Что такое операторные скобки? Каково их назначение?
12. Как выполняется команда присваивания?
13. Какая инструкция предназначена для вывода на экран сообщений?
14. Какая инструкция предназначена для ввода данных?
15. Что такое заголовочный файл? Приведите пример заголовочного файла.
16. Каково назначение заголовочных файлов?
17. Как сохранить программу? (Устно)
18. Как осуществить компиляцию текста программы? (Устно)
19. Как осуществить запуск программы? (Устно)
20. Как считать готовую программу с диска? (Устно)

Лабораторная работа № 3

Программирование алгоритмов разветвляющейся структуры

ЦЕЛЬ РАБОТЫ: приобретение навыков программирования вычислительных разветвляющихся процессов с использованием условного, составного операторов, оператора выбора (множественного ветвления) и оператора безусловного перехода.

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: в соответствии с вариантом составить и реализовать программы.

Задание I

Решить задачу с использованием конструкции ветвления:

1. Расстояние от дома до школы S км. До начала урока осталось t минут. Ученик идет со скоростью v . Определить, придёт ли он раньше звонка, после звонка или во время.
2. Даны действительные числа x , y и z . Найти $\min(x, y, z)$.
3. Вычислить значение функции
$$f = \begin{cases} 2x\sqrt{x}, & \text{если } x > 1, \\ \frac{x^2}{5}, & \text{если } x \leq 0. \end{cases}$$
4. Возвести в квадрат большее из трёх заданных чисел.
5. Определить, есть ли среди трёх заданных чисел чётные.
6. Вывести на экран три заданных числа в порядке убывания.
7. Определить, принадлежит ли точка $A(a_1, a_2)$ кольцу, определяемому окружностями: $x^2 + y^2 = 1$ и $x^2 + y^2 = 16$.
8. Вычислить значение функции
$$y = \begin{cases} \ln^3(x^2 + 1), & \text{если } x \leq -3, \\ \arctg \frac{1}{2x + 1}, & \text{если } x > 3. \end{cases}$$
9. Две окружности заданы координатами центра и радиусами. Сколько точек пересечения имеют эти окружности?
10. В два сосуда, один из которых имеет форму шара радиуса R_1 , а другой – форму куба с ребром A , доверху налита вода. Сравните количество воды в сосудах.
11. Вычислить значение функции:
$$f = \begin{cases} x \sin^2 x, & \text{если } x < 1, \\ e^x, & \text{если } x \geq 2. \end{cases}$$
12. Даны три произвольных действительных числа. Можно ли построить треугольник с такими длинами сторон?

13. Какая из точек $A(a_1, a_2)$ или $B(b_1, b_2)$ находится дальше от начала координат?
14. Две прямые заданы уравнениями: $a_1x + b_1y + c_1 = 0$ и $a_2x + b_2y + c_2 = 0$. Определить, пересекаются ли они.
15. Попадёт ли точка $A(a_1, a_2)$ в окружность заданного радиуса с центром в начале координат?

$$16. \text{ Вычислить значение функции } y = \begin{cases} ax^2 + bx + c, & \text{если } x < 1.2, \\ \frac{a}{x} + \sqrt{x^2 + 1}, & \text{если } x = 1.2, \\ \frac{a + bx}{\sqrt{x^2 + 1}}, & \text{если } x > 1.2, \end{cases}$$

где $a = 2.8$, $b = -0.3$, $c = 4$.

17. В заборе выпилена дыра прямоугольной формы с известными размерами. Определить, пройдёт ли в эту дыру мяч заданного радиуса.
18. Две окружности заданы координатами центра и радиусами. Определить, пересекаются ли они, касаются друг друга или не имеют общих точек.
19. В григорианском календаре каждый год, номер которого делится на 4, является високосным, за исключением тех, которые делятся на 100 и не делятся на 400 нацело. Определить число дней в году по номеру года. Т.о. 1900 г. - не високосный, 2000 г. – високосный.
20. Определить, есть ли среди четырёх заданных чисел кратные пяти.

$$21. \text{ Вычислить значение функции } y = \begin{cases} at^2 \ln t, & \text{если } 1 \leq t \leq 2, \\ 1, & \text{если } t < 1, \\ e^{at} \cos bt, & \text{если } t > 2. \end{cases}, \text{ где } a = -0.5, b = 2.$$

22. Из трех действительных чисел a , b и c выбрать те, модули которых больше 4.
23. Даны три произвольных действительных числа. Можно ли построить треугольник с такими длинами сторон? Если да, то вывести на экран является ли заданный треугольник равносторонним, равнобедренным или разносторонним.
24. Даны действительные числа x и y . Найти $U = \max^2(x^2y, xy^2)$.
25. Решить квадратное уравнение.

Задание II

Решить задачу с использованием конструкции выбора:

1. Дано число c . Распечатать величину этого числа в словесной форме, учитывая его знак. Предусмотреть, что $-7 \leq c \leq 7$.
2. Дано число m ($1 \leq m \leq 12$). Определить, сколько дней в месяце с номером m . Год не високосный.

3. С клавиатуры вводится натуральное число n – количество граней выпуклого многогранника. В зависимости от значения n вывести на экран фразу «Выпуклый многогранник имеет n граней», согласовав окончание слова "грань" с числом n . Предусмотреть, что $0 \leq n \leq 25$. Если выпуклого многогранника с заданным количеством граней не существует, выдать соответствующее сообщение.
4. С клавиатуры вводится цифра m (от 1 до 4). Вывести на экран названия месяцев, соответствующих времени года с номером m (считать зиму временем года № 1).
5. С клавиатуры вводится натуральное число n – количество граней выпуклого многогранника. В зависимости от значения n вывести на экран в словесной форме название правильного выпуклого многогранника или сообщить, что правильного выпуклого многогранника с таким количеством граней не существует.

Предусмотреть, что $4 \leq n \leq 20$.

Примечание. Существует пять выпуклых правильных многогранников: тетраэдр (4 грани); гексаэдр или куб (6 граней); октаэдр (8 граней); додекаэдр (12 граней); икосаэдр (20 граней)

6. Дано натуральное число $n < 25$. Вывести на экран фразу «мне n лет (год или года)» соответственно.

7. Найти значение функции:
$$f = \begin{cases} 2\sin^2 x, & \text{если } -1 \leq x \leq 1, \\ 3x^3, & \text{если } -10 < x < -1, \\ 0 & \text{в остальных случаях.} \end{cases}$$

8. Дано число m ($1 \leq m \leq 12$). Определить, к какому времени года относится месяц с номером m .
9. Вводится целое число $|C| < 5$. Если $C > 0$, то вывести на экран название этого числа в словесной форме. В противном случае сообщить, что число отрицательное.
10. С клавиатуры вводится натуральное число n – количество углов многоугольника. В зависимости от значения n вывести на экран название многоугольника или сообщить, что многоугольника с таким количеством углов не существует.

Предусмотреть, что $0 \leq n \leq 6$.

11. С клавиатуры вводится натуральное число n . В зависимости от значения остатка r при делении числа n на 7 вывести на экран число n в виде $n = 7k + r$, где r представить в словесной форме.
12. Дано число m . Определить полугодие, на которое приходится месяц m и количество дней в этом месяце. Предусмотреть, что $1 \leq m \leq 12$. Год не високосный.
13. Дано число n . Напечатать фразу "Мы успешно сдали n экзаменов", согласовав окончание слова "экзамен" с числом n . Предусмотреть, что $1 \leq n \leq 25$.

14. Определить время года, к которому относится месяц m и определить количество дней в этом месяце. Предусмотреть, что $1 \leq m \leq 12$. Год високосный.
15. Дано число c . Распечатать величину этого числа в словесной форме. Предусмотреть, что $0 \leq c \leq 7$.
16. Даны два числа D – день и M – месяц. Определить K – порядковый номер того дня високосного года, который имеет дату D и M .
17. Даны три числа D , M и G , определяющие день, месяц и год. Проверить образуют ли они правильную дату и вывести соответствующее сообщение. Например, 31.06.90 – неправильная дата. Год не високосный.
18. С клавиатуры вводится цифра от 0 до 9. Вывести на экран название этой цифры в словесной форме.
19. В понедельник фирма работает с 9-00 до 16-00; во вторник, среду, четверг, пятницу – с 8-00 до 19-00; в субботу – с 10-00 до 15-00; воскресенье – выходной. По заданному номеру дня недели определить часы работы.

$$20. \text{ Найти значение функции: } y = \begin{cases} x\sqrt{x-a^2}, & \text{если } x = a+1; \\ \frac{a}{x} - \frac{1}{2}\cos 2x, & \text{если } 0 \leq x \leq a; \text{ где } a = 4. \\ e^x, & \text{если } -a \leq x < 0; \\ 1 & \text{в иных случаях.} \end{cases}$$

21. С клавиатуры вводится натуральное число n . В зависимости от значения остатка r при делении числа n на 5 вывести на экран число n в виде $n = 5k + r$, где r представить в словесной форме.
22. На первом курсе студенты четырёх групп сдают зачёт по информатике. На втором курсе группа № 1 сдаёт экзамен по дисциплине «Информационные технологии в экономике», группа № 2 – экзамен по дисциплине «Информационные технологии в физике», группа № 3 – экзамен по дисциплине «Информационные технологии в географии», группа № 4 – экзамен по дисциплине «Информационные технологии в математических исследованиях». Задать с клавиатуры номер курса и номер группы и вывести на экран сообщение, какой зачёт или экзамен группа будет сдавать.
23. С клавиатуры задано целое число $N < 5$. Если оно положительное, то вывести на экран его название в словесной форме. В противном случае найти квадрат этого числа.
24. Студенты убирают яблоки. Нормы следующие: студент 1 курса должен собрать в день не менее 50 кг яблок, студент второго курса – не менее 60 кг в день, студент 3 курса – не менее 70 кг в день. Зная курс, на котором учится студент, определить, выполнил ли он дневную норму.
25. Дано число m ($1 \leq m \leq 12$). Определить, сколько дней в месяце с номером m . Учесть високосный или не високосный год.

Задания для самостоятельной работы

1. Определить D и M -дату K -го по счету дня високосного года. Месяц вывести в словесной форме. D - день, M - месяц. Предусмотреть, что $1 \leq K \leq 366$.
2. Компания по снабжению электроэнергией взимает плату с клиентов по тарифу:
 - 7 р. за 1 кВт/ч за первые 250 кВт/ч;
 - 17 р. за кВт/ч, если потребление свыше 250, но не превышает 300 кВт/ч;
 - 20 р. за кВт/ч, если потребление свыше 300 кВт/ч.Потребитель израсходовал n кВт/ч. Подсчитать плату.
3. Школьники сдают нормы по прыжкам в длину. Если длина прыжка больше 2,50 м, то оценка – «5», если от 2 м до 2,5 м - оценка «4»; от 1,5 м до 2 м - оценка «3»; если меньше 1,5 м – «2». Выставить школьнику оценку, если известна длина его прыжка. Примечание: 1 м = 100 см.
4. При покупке товара на сумму от 200 до 500 руб предоставляется скидка 3%, при покупке товара на сумму от 500 до 800 – скидка 5%, при покупке товара на сумму от 800 до 1000 руб – скидка 7%, свыше 1000 руб – скидка 10%. Покупатель приобрел 8 рулонов обоев по цене X_1 и две банки краски по цене X_2 . Сколько он заплатил?
5. Студенты убирают урожай помидоров. При сборе до 50 кг в день работа оплачивается из расчёта 30 руб. за 1 кг; при сборе от 50 до 75 кг в день – 50 руб. за 1 кг; при сборе от 75 до 90 кг в день – 65 руб. за 1 кг; при сборе свыше 90 кг в день – 70 руб. за 1 кг плюс 120 руб. премия. Студент собрал X кг за день. Определить его заработок.
6. Ученики начальной школы сдают технику чтения. В 1 классе скорость чтения до 20 слов в мин считается низкой, от 21 до 40 слов в мин – средней, от 41 до 50 – выше среднего, более 51 слова в мин – высокой. Во втором классе скорость чтения до 30 слов в мин считается низкой, от 31 до 60 слов в мин – средней, от 61 до 70 – выше среднего, более 71 слова в мин – высокой. Задать с клавиатуры номер класса и количество прочитанных учеником в минуту слов. Оценить скорость чтения.
7. Студенты убирают яблоки. Студентам 1 курса при сборе до 50 кг в день работа оплачивается из расчёта 30 коп. за 1 кг; при сборе от 50 до 75 кг в день – 50 коп. за 1 кг; при сборе от 75 до 90 кг в день – 65 коп. за 1 кг; при сборе свыше 90 кг в день – 70 коп. за 1 кг плюс 20 руб премия. Студентам 2 курса при сборе до 60 кг в день работа оплачивается из расчёта 35 коп. за 1 кг; при сборе от 61 до 80 кг в день – 55 коп. за 1 кг; при сборе от 81 до 95 кг в день – 65 коп. за 1 кг; при сборе свыше 95 кг в день – 70 коп. за 1 кг плюс 30 руб премия.

Студент собрал X кг яблок за день. Определить его заработок.
8. С клавиатуры задаётся целое число $A > -5$. Для положительного однозначного числа определить, является ли оно чётным; для положительного двузначного определить, кратно ли оно 3; для положительного трёхзначного числа опре-

- делить, кратно ли оно 5; для числа, большего 999, определить, кратно ли оно 10. Если число $A < 0$, то вывести на экран его название в словесной форме.
9. Школьники сдают нормы по прыжкам в длину. Для учеников 5 класса нормы следующие: если длина прыжка больше 2,50 м, то оценка - 5, если от 2 м до 2,5 - оценка 4; от 1,5 м до 2 м - оценка 3; если меньше 1,5 м - 2. Нормы для учеников 6 класса: если длина прыжка больше 3,0 м, то оценка - 5, если от 2,5 м до 3,0 - оценка 4; от 2,0 м до 2,5 м - оценка 3; если меньше 2,0 м – оценка 2. Ученики других классов прыжки не сдают.
- Выставить школьнику оценку, если известны номер его класса и длина прыжка.
10. Школьники сдают нормы по прыжкам в длину. Для учеников 2 класса минимальная длина прыжка должна быть не менее 1,2 м; для учеников 3 класса – не менее 1,5 м; для 4 класса – не менее 1,7 м; для 5 класса не менее 2,0 м.
- Зная номер класса и длину прыжка ученика, определить, сдан ли норматив.
11. С клавиатуры задаётся натуральное число $N < 100$. Если оно однозначное, то вывести на экран его название в словесной форме. Если число двузначное, то сообщить об этом и определить остаток от деления его на 2, 3, 5.
12. Компания по снабжению электроэнергией взимает плату с клиентов по тарифу:
- ✓ 7 р. за 1 кВт/ч за первые 250 кВт/ч;
 - ✓ 17 р. за кВт/ч, если потребление свыше 250, но не превышает 300 кВт/ч;
 - ✓ 20 р. за кВт/ч, если потребление свыше 300 кВт/ч.
- Существуют три категории потребителей. Тариф 1 категории составляет 75% общей стоимости, тариф 2 категории – половину общей стоимости. Остальные оплачивают полностью.
- Потребитель израсходовал n кВт/ч. Подсчитать плату, зная его категорию.
13. С клавиатуры задано натуральное число $N < 10000$. Сообщить, является ли оно однозначным, двузначным, трёхзначным или четырёхзначным. В каждом случае определить, чётное оно или нечётное.
14. Ученики начальной школы сдают технику чтения. Нормы следующие: в 1 классе ребёнок должен читать не менее 20 слов в минуту; во втором классе – не менее 50 слов в мин; в 3 классе – не менее 60 слов в мин, в 4 классе – не менее 70 слов в минуту. Задавая с клавиатуры номер класса и количество прочитанных учеником в минуту слов, определить, сдана ли техника чтения.
15. Даны три целых числа определяющие дату: год, месяц, день. Считая, что год не високосный, определить дату следующего дня.

Контрольные вопросы

1. Что такое составной оператор?
2. Какова полная (неполная) форма команды ветвления (блок-схема)?
3. Каков алгоритм выполнения команды ветвления?

4. Каков алгоритм выполнения команды множественного ветвления (выбора)? Блок-схема.
5. Какие операторы сравнения используются в JAVA?
6. Что называется простым условием? Приведите примеры.
7. Что такое составное условие? Приведите примеры.
8. Какие логические операторы допускаются при составлении сложных условий?
9. Каков результат применения оператора логическое "И" (&&)?
10. Каков результат применения оператора логическое "ИЛИ" (||)?
11. Каков результат применения оператора логическое отрицание (!)?
12. Каков общий вид (формат) инструкции «Ветвление»?
13. Каков алгоритм выполнения условной (тернарной) операции (?:)? Приведите пример.
14. Каков общий вид (формат) инструкции «Выбор»?
15. Может ли оператор ветвления содержать внутри себя другие ветвления?

Лабораторная работа № 4

Программирование алгоритмов циклической структуры

ЦЕЛЬ РАБОТЫ: приобретение навыков программирования вычислительных повторяющихся процессов с использованием циклических операторов.

Теоретическая часть

Циклы

Цикл - вид конструкции, используемый для вычислений, повторяющихся многократно.

Блок, ради которого и организуется цикл, называется **телом цикла**. Остальные операторы служат для управления процессом повторения вычислений: это начальные установки, проверка условия продолжения цикла и модификация параметра цикла (рис.1). Один проход цикла называется **итерацией**.

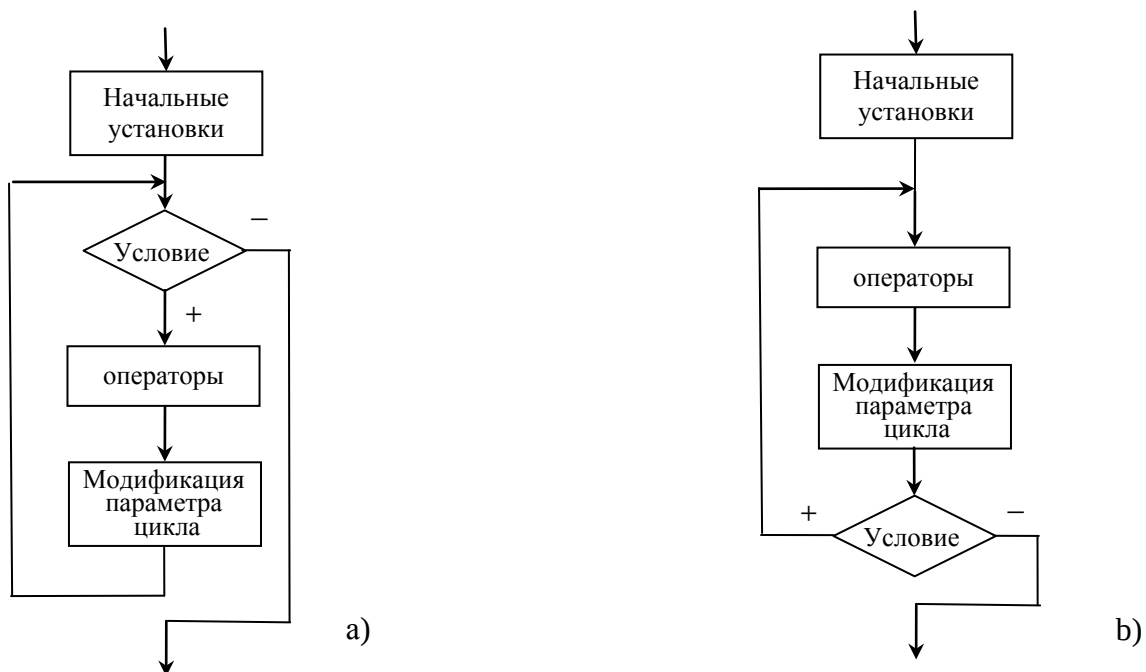


Рисунок 4.1 – Структурные схемы выполнения цикла

Начальные установки служат для того, чтобы до входа в цикл задать значения переменных, которые в нем используются.

Проверка условия продолжения цикла выполняется на каждой итерации либо до тела цикла (тогда говорят о цикле с предусловием, рис. 4.1а), либо после тела цикла (цикл с постусловием, рис. 4.1б). Разница между ними состоит в том, что тело цикла с постусловием всегда выполняется хотя бы один раз, после чего проверяется надо ли его выполнять еще раз. Проверка необходимости выполнения цикла с предусловием делается до тела цикла, поэтому возможно, что он не выполнится ни разу.

Параметром цикла называется переменная, которая используется при проверке условия цикла и принудительно изменяется на каждой итерации, причем, как правило на одну и ту же величину. Если параметр цикла целочисленный, он называется счетчиком цикла.

Для реализации циклических вычислений предусмотрены три оператора: цикл с предусловием, цикл с постусловием, цикл с параметром. Первые два оператора используются в том случае, когда число повторений цикла неизвестно. Если же число повторений цикла может быть определено перед его началом, то используется оператор цикла с параметром.

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: в соответствии с вариантом составить и реализовать программы.

Задание I

Используя оператор цикла с параметром, решить предложенную задачу:

1. Вычислить сумму и произведение первых n натуральных чисел. Предусмотреть ввод n с клавиатуры.
2. Вычислить $S = \sum_{n=1}^k (-1)^n n^2$, $k \geq 10$.
3. Вычислить сумму первых n членов ряда с общим членом $a_k = \frac{k+1}{\sqrt{k^2+2}}$, $k = \overline{1, n}$.
4. Определить средний рост девочек, и мальчиков одного класса. В классе учатся n учеников. ($n \geq 15$).
5. Вводя в цикле по 4 оценки каждого студента группы, подсчитать число студентов, не имеющих оценок 2 и 3. В группе учатся n студентов. Предусмотреть ввод n с клавиатуры.
6. Вводя в цикле по 4 оценки, полученные студентами в сессию, определить средний балл группы по всем экзаменам. В группе учатся n студентов. Предусмотреть ввод n с клавиатуры.
7. Задано n троек, чисел a , b , c . Вводя их по очереди с клавиатуры в порядке возрастания и интерпретируя их как длины сторон треугольника, определить, сколько троек может быть использовано для построения треугольника.
8. В ЭВМ по очереди поступают результаты соревнований по плаванию на дистанции 200 м, в которых участвует n спортсменов ($n > 10$). Выдать на экран дисплея лучший результат после ввода результата очередного спортсмена.
9. В ЭВМ вводятся по очереди координаты n точек ($n \geq 10$). Определить, сколько из них попадет в круг радиуса r_c центром в точке (a, b) .
10. Ученикам первого класса назначается стакан молока (200 мл), если их вес составляет меньше 30 кг. Определить, сколько литров молока потребуется, ежедневно для одного класса, состоящего из n учеников. После взвешивания вес каждого ученика вводится в ЭВМ.
11. В ЭВМ вводятся номер по списку и рост учеников 10 класса. Вывести на экран номера по списку тех учеников, рост которых больше 170 см.

12. В соревнованиях по бегу на дистанции 100 м принимают участие n спортсменов. Вводя очереди номера по списку и результаты участников в ЭВМ, определить, сколько из них выполнили норму, и напечатать их номера (норма равна 13,2 сек.)
13. В ЭВМ вводятся по очереди координаты n точек. Определить, сколько из них попадет в кольцо с внутренним радиусом r_1 , внешним r_2 и центром в начале координат.
14. В ЭВМ по очереди вводятся координаты n точек. Определить, сколько из них принадлежит фигуре, ограниченной осью абсцисс и аркой синусоиды, построенной для аргумента со значениями от 0 до π .
15. Задана окружность радиуса r с центром в начале координат. Вводя последовательно в ЭВМ координаты n точек, являющихся центрами других окружностей того же радиуса, определить, сколько из этих окружностей пересекает данную.
16. Найти все трехзначные натуральные числа, сумма цифр которых равна их произведению.
17. Найти сумму целых положительных чисел, больших 20, меньших 100 и кратных 3.
18. Вычислить сумму $S = 1 + \frac{1}{2^3} + \frac{1}{3^3} + \dots + \frac{1}{n^3}$, где $n \geq 10$.
19. Вводя в цикле по 4 оценки, полученные студентами в сессию, определить число неуспевающих студентов группы. В группе учится n студентов. Предусмотреть ввод n с клавиатуры.
20. Цилиндр объема 1 куб.ед. имеет высоту h . Определить радиус основания для значений h от 1 до 6 с шагом 0.5 ед.
21. Вывести на печать последовательность с общим членом $a_k = \sin(k^2 + b)$, где $k = 0, 1, \dots, K$. Предусмотреть ввод K и b с клавиатуры. Вычислить количество элементов, удовлетворяющих условию: $0 \leq a_k \leq 0.5$.
22. В ЭВМ вводятся поочередно n точек. Определить, сколько из них принадлежит фигуре, ограниченной осью ординат и правой полуокружностью $x^2 + y^2 = r^2$.
23. В ЭВМ по очереди вводятся координаты n точек. Определить, сколько из них принадлежит фигуре, ограниченной линией $|x| + |y| = 1$.
24. Составить программу вычисления значений функции $p(x) = \frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5)$ при изменении x от 0 до 1 с шагом 0.1.
25. Вычислить значение выражения $\frac{(x-2)(x-4)(x-6)\dots(x-64)}{(x-1)(x-3)(x-5)\dots(x-63)}$ для произвольного действительного числа x из области определения данного выражения.

Задание II

Двумя способами (используя, оператор цикла с предусловием и оператор цикла с постусловием) для данного значения x найти сумму ряда S с точностью до члена ряда, по абсолютной величине меньшего $\varepsilon_{ps} = 0.0001$. Сравнить эту сумму со значением контрольной функции y .

$$1. \quad S = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots; \quad x = 1.5; \quad y = e^x.$$

$$2. \quad S = \sum_{n=0}^{\infty} \frac{x^n \ln^n 3}{n!} = 1 + \frac{x \ln 3}{1!} + \frac{x^2 \ln^2 3}{2!} + \dots; \quad x = 1; \quad y = 3^x.$$

$$3. \quad S = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \dots; \quad x = 0.7; \quad y = \frac{1}{1-x}.$$

$$4. \quad S = \sum_{n=0}^{\infty} (-1)^n \frac{x^n}{2 \cdot 2^n} = \frac{1}{2} - \frac{x}{4} + \frac{x^2}{8} - \dots; \quad x = 1.2; \quad y = \frac{1}{2+x}.$$

$$5. \quad S = \sum_{n=1}^{\infty} \left(\frac{1}{2^n} + \frac{1}{3^n} \right) x^{n-1} = \frac{5}{6} + \frac{13}{36}x + \frac{35}{216}x^2 + \dots; \quad x = -0.8; \quad y = \frac{5-2x}{6-5x+x^2}.$$

$$6. \quad S = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots; \quad x = 0.3; \quad y = \cos x.$$

$$7. \quad S = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^n}{n} = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots; \quad x = 0.4; \quad y = \ln(1+x).$$

$$8. \quad S = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{\sin(nx)}{n} = \sin x - \frac{\sin 2x}{2} + \dots; \quad x = -\frac{\pi}{2}; \quad y = \frac{x}{2}.$$

$$9. \quad S = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \dots; \quad x = 2; \quad y = \frac{e^x - e^{-x}}{2}.$$

$$10. \quad S = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots; \quad x = 0.6; \quad y = \arctg x.$$

$$11. \quad S = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \dots; \quad x = 1.4; \quad y = \sin x.$$

$$12. \quad S = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots; \quad x = 0.5; \quad y = \frac{(e^x + e^{-x})}{2}.$$

$$13. \quad S = \sum_{n=1}^{\infty} \frac{\sin(2n-1)x}{2n-1} = \sin x + \frac{\sin 3x}{3} + \dots; \quad x = \frac{\pi}{2}; \quad y = \frac{\pi}{4}.$$

14. $S = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{\cos nx}{n} = \cos x - \frac{\cos 2x}{2} + \dots; x = \frac{\pi}{8}; y = \ln \left(2 \cos \frac{x}{2} \right).$
15. $S = \sum_{n=1}^{\infty} \frac{\cos nx}{n} = \cos x + \frac{\cos 2x}{2} + \dots; x = \pi; y = -\ln \left(2 \sin \frac{x}{2} \right).$
16. $S = 2 \sum_{n=1}^{\infty} \frac{x^{2n-1}}{2n-1} = 2 \left(x + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right); x = -0.4; y = \ln \frac{1+x}{1-x}.$
17. $S = \sum_{n=1}^{\infty} \frac{1}{(2n-1)x^{2n-1}} = \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots; x = 3; y = \frac{1}{2} \ln \frac{x+1}{x-1}.$
18. $S = \sum_{n=1}^{\infty} \frac{1}{(2n-1)} \left(\frac{x-1}{x+1} \right)^{2n-1} = \frac{x-1}{x+1} + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \dots; x = 0.6; y = \frac{1}{2} \ln x.$
19. $S = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{n!} = 1 - \frac{x^2}{1!} + \frac{x^4}{2!} - \frac{x^6}{3!} + \dots; x = -0.7; y = e^{-x^2}.$
20. $S = \sum_{n=0}^{\infty} (n+1)x^n = 1 + 2x + 3x^2 + 4x^3 + \dots; x = -0.7; y = \frac{1}{(1-x)^2}.$
21. $S = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots; x = 0.35; y = \ln \sqrt{\frac{1+x}{1-x}}.$
22. $S = \sum_{n=1}^{\infty} \frac{(-1)^n}{n} (x+1)^{2n} = -(x+1)^2 + \frac{1}{2} (x+1)^4 + \dots; x = -1.2; y = \ln \frac{1}{2 + 2x + x^2}.$
23. $S = 1 - \sum_{n=1}^{\infty} (-1)^{n+1} \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-3)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n} x^n = 1 + \frac{1}{2} x - \frac{1}{2 \cdot 4} x^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6} x^3 - \dots;$
 $x = -0.8; y = \sqrt{1+x}.$
24. $S = 1 + \sum_{n=1}^{\infty} \frac{1 \cdot 3 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot \dots \cdot 2n} x^{2n} = 1 + \frac{1}{2} x^2 + \frac{1 \cdot 3}{2 \cdot 4} x^4 + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} x^6 + \dots; x = 0.64;$
 $y = \frac{1}{\sqrt{1-x^2}}.$
25. $S = \sum_{n=1}^{\infty} \frac{\sin nx}{n} = \sin x + \frac{\sin 2x}{2} + \dots; x = \frac{\pi}{3}; y = \frac{\pi - x}{2}.$

Задание III

1. Какое количество теплоты потребуется для испарения 100 г воды при 100, 110, ..., 200°C?
2. Определите внутреннюю энергию 1 г гелия при температуре 10, 15, ..., 50°C.

3. В сосуде объёмом 1 см^3 находится 10^{16} молекул газа. Определить давление газа в сосуде при температуре $0, 1, \dots, 10^\circ\text{C}$.
4. Какова средняя квадратичная скорость молекул кислорода при $10, 12, \dots, 20^\circ\text{C}$?
5. Сколько теплоты выделится при конденсации $100, 200, \dots, 900 \text{ г}$ водяного пара?
6. В котле ёмкостью 6.0 м^3 содержится 25 кг перегретого пара. Каково давление пара, если его температура равна $200, 220, \dots, 300^\circ\text{C}$?
7. Какова средняя квадратичная скорость молекул водорода при температуре $10, 15, \dots, 50^\circ\text{C}$?
8. Определите внутреннюю энергию 1 г гелия при температуре $20, 22, \dots, 40^\circ\text{C}$.
9. В сосуде объёмом 480 см^3 при давлении $2.5 \cdot 10^4 \text{ Н/м}^2$ находится газ. Подсчитать количество молекул газа для значений температуры $10, 11, \dots, 20^\circ\text{C}$.
10. Сколько теплоты потребуется для испарения 200 г воды при температуре $100, 110, \dots, 200^\circ\text{C}$?
11. Какова средняя квадратичная скорость молекул углекислого газа при температуре $0, 5, \dots, 30^\circ\text{C}$?
12. Определить среднюю квадратичную скорость молекул водорода при $0, 5, \dots, 50^\circ\text{C}$.
13. В сосуде объёмом 450 см^3 при давлении $2.5 \cdot 10^4 \text{ Н/м}^2$ находится газ. Подсчитать количество молекул газа для значений температуры $10, 12, \dots, 20^\circ\text{C}$.
14. Определите внутреннюю энергию 1 г водорода при температуре $20, 21, \dots, 30^\circ\text{C}$.
15. Определить среднюю кинетическую энергию вращательного движения молекул водорода, содержащихся в 1.0 моль при $10, 11, \dots, 20^\circ\text{C}$.
16. Плотность воздуха при нормальных условиях 1.3 г/л . Какова плотность воздуха при давлении $4.0 \cdot 10^5 \text{ Н/м}^2$ и температуре $50, 60, \dots, 100^\circ\text{C}$?
17. Определите внутреннюю энергию 1 г углекислого газа при температуре $10, 15, \dots, 50^\circ\text{C}$.
18. В колбе содержится газ при нормальных условиях. Сколько молекул газа содержится в колбе, если её ёмкость равна $0.1, 0.2, \dots, 0.5 \text{ л}$?
19. Какова средняя квадратичная скорость молекул водорода при $0, 2, \dots, 20^\circ\text{C}$?
20. Какова средняя квадратичная скорость молекул кислорода при температуре $0, 2, \dots, 10^\circ\text{C}$?
21. В сосуде объёмом 400 см^3 при давлении $2.5 \cdot 10^4 \text{ Н/м}^2$ находится газ. Подсчитать количество молекул газа для значений температуры $0, 1, \dots, 10^\circ\text{C}$.
22. Сколько теплоты потребуется для испарения 150 г воды при температуре $100, 105, \dots, 150^\circ\text{C}$?
23. Определите внутреннюю энергию 1 г гелия при температуре $20, 21, \dots, 30^\circ\text{C}$.

24. Какова средняя квадратичная скорость молекул углекислого газа при температуре 0, 5, ..., 30°C ?
25. Определите внутреннюю энергию 1 г кислорода при температуре 10, 15, ..., 50°C.

Список литературы, рекомендуемый к использованию по дан-ной те- ме:

[1] стр. 32-36;
 [5]Операторы цикла - <https://informatics.msk.ru/course/view.php?id=8>;
 [6]Цикл while - <https://stepik.org/lesson/13023/step/1?unit=4371>
 Учить:п. 4.1-4.4
 Письменно ответить на контрольные вопросы.

Контрольные вопросы

1. Какие базовые алгоритмические конструкции можно реализовать средствами языка JAVA?
2. Что называется циклом?
3. Что такое тело цикла?
4. Что такое параметр цикла?
5. Что такое итерация?
6. Что такое заикливание?
7. Какие типы циклов существуют?
8. Какие инструкции используются для реализации циклов на JAVA?
9. Какая инструкция используется для реализации цикла с параметром?
10. Можно ли использовать в теле цикла переменную, являющуюся параметром цикла?
11. Какая инструкция используется для реализации цикла с предусловием?
12. От чего зависит количество повторений тела цикла с предусловием?
13. Какая инструкция используется для реализации цикла с постусловием?
14. Какие из конструкций повторения могут привести к заикливанию?
15. Как работает цикл со счетчиком?
16. В каком случае в теле цикла используются операторные скобки?
17. Каковы отличия между циклами с пред- и с постусловием?

Пример выполнения Практической работы

Задание I. Составить программу табулирования функции

$$y = e^x \frac{2.42x^2 - 6.47x + 1.03}{x - 5.8} \text{ на отрезке } [-4, 4] \text{ с шагом } 0.2.$$

Решение

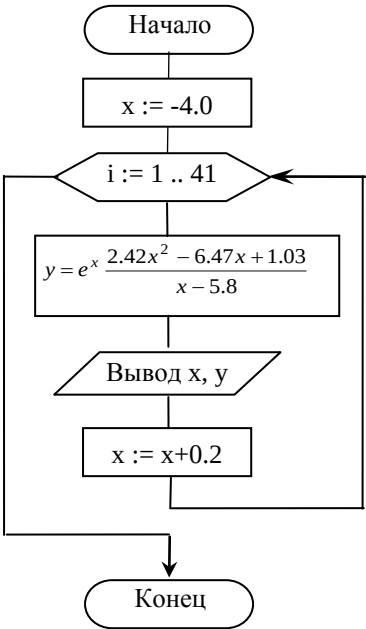
1. Математическая модель

Прежде всего, определим число повторений тела цикла: $S = \frac{4 - (-4)}{0.2} + 1 = 41.$

Аргументы: переменная x вещественного типа.

Результаты: значение заданной функции y (вещественного типа).

Промежуточные величины: счетчик i целого типа.

2. Алгоритм	3. Программа
	<pre>#include<iostream> #include<conio.h> #include<math.h> usingnamespace std; int main() { double x = -4, y; for(int i=1; i<=41; i++) { y = exp(x)*(2.42*pow(x, 2.0) - 6.47*x+1.03)/(x-5.8); cout<<" x = "<<x<<" y = "<<y<<endl; x+=0.2; } _getch(); return 0; }</pre>

4. Результат работы программы

...

$x = -3.6$	$y = -0.161865$
$x = -3.4$	$y = -0.185016$

...

$x = -2.2$	$y = -0.373639$
------------	-----------------

...

Задание II. Для $x = 0.47$ найти сумму ряда

$$y = x + \left(\frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} + \dots + \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n} \cdot \frac{x^{2n+1}}{2n+1} + \dots \right)$$

с точностью до члена ряда, меньшего (по абсолютной величине) $\text{eps} = 0.0001$, используя оператор цикла с предусловием. Сравнить полученную сумму со значением функции для контроля $y = \arcsin x$.

Решение

1. Математическая модель. Обозначим через $p1$ последовательные произведения нечетных чисел. Через $p2$ – последовательные произведения четных чисел.

В C++ (`math.h`), есть функция `arcsin x`, поэтому для вычисления контрольной функции воспользуемся соотношением: $y = \text{asin}(x)$.

Аргументы: x – переменная вещественного типа, $-1 < x < 1$.

Результаты: `summa` – сумма членов ряда вещественного типа;

у—значение контрольной функции(вещественного типа).

Промежуточные величины:

eps — константа, с которой сравнивается абсолютная величина текущего члена ряда: члены ряда представляют собой убывающую последовательность, поэтому согласно условию вычисление суммы членов ряда будет продолжаться до тех пор пока абсолютное значение текущего члена ряда не окажется меньше eps;

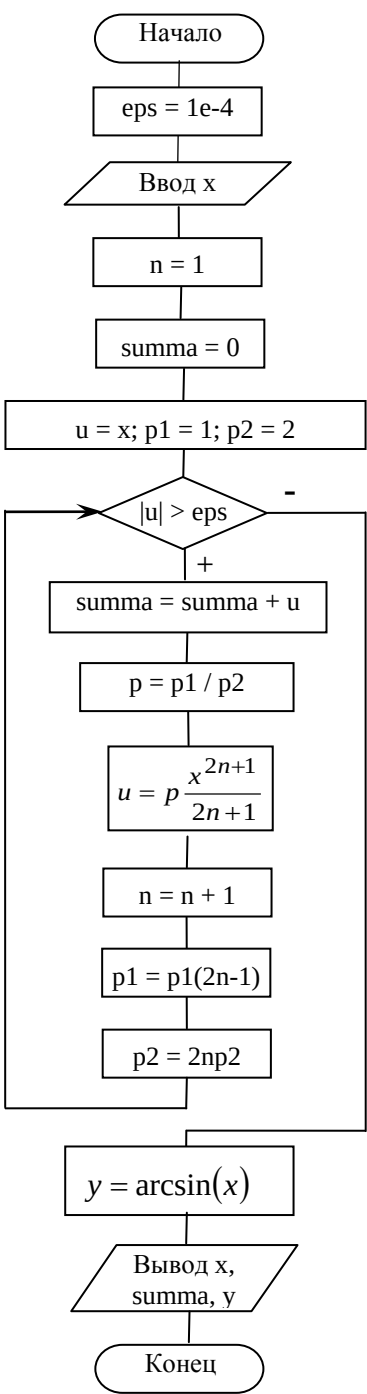
p1— последовательные произведения нечетных чисел, вещественного типа;

p2— последовательные произведения четных чисел, вещественного типа;

p — частное, получаемое при делении p1 на p2, вещественного типа;

n — номер члена ряда, целого типа; u — текущий член ряда, вычисляемый по фор-

муле $u = p \frac{x^{2n+1}}{2n+1}$, вещ. типа;

2. Алгоритм	3. Программа
 <pre> graph TD Start([Начало]) --> Init[eps = 1e-4] Init --> Input[/Ввод x/] Input --> N1[n = 1] N1 --> Summa[summa = 0] Summa --> InitVars[u = x; p1 = 1; p2 = 2] InitVars --> Decision{<math> u > \text{eps}</math>} Decision -- "+" --> SummaAdd[summa = summa + u] SummaAdd --> PDiv[p = p1 / p2] PDiv --> UCalc["<math>u = p \frac{x^{2n+1}}{2n+1}</math>"] UCalc --> NInc[n = n + 1] NInc --> P1Calc[p1 = p1(2n-1)] P1Calc --> P2Calc[p2 = 2np2] P2Calc --> Decision Decision -- "-" --> YCalc[y = arcsin(x)] YCalc --> Output[/Вывод x, summa, y/] Output --> End([Конец]) </pre>	

4. Результат работы программы

Введите значение x : 0.47

Сумма данного ряда при $x = 0.4700$ сточностью до 0.0001, $S = 0.4893$

Значение контрольной функции, $y = 0.4893$

Вывод: С точностью до 0.0001 значение суммы ряда и значение контрольной функции совпадают, следовательно, вычисление суммы членов ряда запрограммировано верно.

Лабораторная работа № 5

Использование функций для решения прикладных задач

ЦЕЛЬ РАБОТЫ: закрепление теоретических знаний и приобретение навыков программирования функций.

Теоретическая часть

Подпрограммы

5.1. Понятие подпрограммы

При решении задач часто встречаются случаи, когда для разных наборов исходных данных приходится выполнять одни и те же команды. В некоторых случаях для организации повторяющихся вычислений можно воспользоваться командой повторения. Иногда имеет смысл выделить повторяющиеся последовательности команд в самостоятельный блок, вынеся его за пределы основной программы и в нужном месте организовать вызов этого блока, указав соответствующий набор параметров. Такой блок, реализующий некоторый вполне законченный этап обработки информации и допускающий обращение к нему из различных частей основной программы, называется **подпрограммой**. Подпрограмма оформляется в виде замкнутого участка программы, имеющего чётко обозначенный вход и выход. Применение подпрограмм экономит оперативную память, т.к. для хранения переменных, используемых только в подпрограмме, память отводится только на время выполнения подпрограммы.

Большинство языков программирования содержат следующие разновидности подпрограмм:

- встроенные подпрограммы;
- библиотечные подпрограммы;
- подпрограммы, создаваемые программистом.

Библиотечные (стандартные) функции

JAVA предоставляет стандартные функции, имеющие определённые имена и выполняющие стандартные действия. Эти функции, определённые в языке и объединены в заголовочные файлы или библиотеки. Чтобы использовать библиотечную функцию, необходимо подключить к программе соответствующий заголовочный файл. Чтобы вызвать такую функцию, достаточно в теле основной или пользовательской функции указать её имя и список фактических параметров.

Функции, создаваемые программистом

JAVA позволяет программисту определить свою собственную функцию и в дальнейшем использовать её точно так же, как и библиотечные функции.

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: освоить теоретический материал, выполнить общее для всех задание I и в соответствии с вариантом составить программу (задание II), реализовав пользовательскую функцию.

Задание

В соответствии с вариантом составить и реализовать программу:

1. Три прямоугольных треугольника заданы катетами. Определить, площадь какого из них больше.
2. Три отрезка заданы координатами вершин. Определить, есть ли среди них отрезки равной длины.
3. В порт в среднем приходит три корабля в день. Какова вероятность того, что в порт придёт 2 корабля? 4 корабля? Вычисление вероятности производится по формуле: $P = 3^k e^{-3} / k!$.
4. Найти среднее арифметическое действительных корней трёх квадратных уравнений.
5. Даны три различных бруска, имеющих форму кубов с заданным произвольным ребром. Из каждого выпилена правильная четырехугольная пирамида (основание – грань куба). Вычислить объем материала, ушедшего в отходы.
6. Даны действительные числа a и b . Вычислите,

$$g(1.2, a) + g(b, a) - g(2a - 1, ab), \text{ где } g(x, y) = \frac{(x^2 + y^2)}{x^2 + 2xy + 3y^2 + 4}$$

7. Заданы длины рёбер произвольной треугольной пирамиды. Вычислить площадь полной поверхности этой пирамиды.
8. Вычислить объем «снеговика», состоящего из трех шаров с произвольными радиусами.
9. Каждое из трех колец задано внутренним и внешним радиусами. Чему равна площадь наибольшего кольца?
10. Вычислить значение выражения $a! + b! + c! + d!$.
11. Вычислить площадь «елки», состоящей из четырех треугольников с известными сторонами.
12. Найти минимальное значение среди корней трех квадратных уравнений (каждое уравнений задано тройкой коэффициентов).
13. Даны действительные числа a и b . Вычислить

$$\max(u + v^2, 3.14),$$

где $u = \max(a, b)$, $v = \max(ab, a + b)$, а $\max$ – подпрограмма для вычисления максимального значения для двух данных чисел.

14. Даны радиусы трех шаров. Вычислить среднее арифметическое объемов этих шаров.
15. Дана арифметическая прогрессия a_n , a_1 – первый ее элемент, d – разность. Найти второй, третий, четвертый и седьмой элементы и их сумму.
16. Даны действительные числа s и t . Вычислить $f(t, -2s, 1.17) + f(2.2, t, s - t)$,

$$f(x, y, z) = \frac{2x - y - \sin z}{5 + |z|}.$$

17. Найти наибольшее из пяти заданных чисел, используя подпрограмму нахождения наибольшего из двух чисел.

18. Вычислить значение функции:

$$y = \varphi(3) + \varphi(7.6) \cdot (\varphi(11.2) - \varphi(5)), \text{ если } \varphi(x) = \sqrt{2x^2 - 4x + e^x}$$

19. Вычислить значение функций

$$f(x) = 2 \sin x - 4.2 \cos x + e^x \text{ и } g(x) = \frac{2x^2}{10\sqrt{5x}}$$

для значений $x = 0,1; 0,35; 0,53$.

20. Вычислить значение функции $x = 2f(15) - 4f(-1)g(11) + g(32)$, если

$$f(x) = 10x + e^{2x}; g(x) = \sqrt{2x + x^3}.$$

21. Из куба со стороной a выпилили куб со стороной b ($a > b$). Какой процент материала ушёл в отходы?

22. Каждая из трёх окружностей задана координатами центра и координатами точки на окружности. Какова наибольшая длина окружности?

23. Найти наименьшее из четырёх заданных чисел, используя подпрограмму нахождения наименьшего из двух чисел.

24. Даны действительные числа a и b . Вычислить

$$\min(2u - v^2, 4.5),$$

где $u = \min(a, b)$, $v = \min(a/b, a - b)$, а $\min$ – подпрограмма для вычисления минимального значения для двух данных чисел.

25. Вычислить большие вещественные корни квадратных уравнений $ax^2 + bx + c = 0$ и $tx^2 + rx + q = 0$, если они существуют. В противном случае вывести на экран монитора соответствующее сообщение.

Контрольные вопросы

1. Что такое подпрограмма?
2. Как подпрограмму можно реализовать в JAVA?
3. Какие параметры называются формальными?
4. Какие параметры называются фактическими?
5. Что такое переменная?
6. Что такое область действия идентификатора?
7. Что такое локальная переменная? Что такое глобальная переменная?
8. Какова область действия локальных идентификаторов?
9. Какова область действия глобальных идентификаторов?
10. Что такое функция?
11. Чем отличается объявление функции от ее определения (описания)?
12. Перечислите составные части описания функции.
13. Что такое прототип функции?

14. Как описывается функция пользователя в программе?
15. Что такое возвращаемое значение функции?
16. Как осуществляется вызов функций из основной программы?

Пример выполнения Практической работы

Задание. Задавая с клавиатуры координаты вершин треугольника, определить, является ли он равносторонним.

Решение

1. Математическая модель

Треугольник является равносторонним, если длины его сторон равны. Находим длины сторон треугольника, используя подпрограмму нахождения расстояния между двумя точками.

Аргументы: $a_1, b_1, c_1, a_2, b_2, c_2$ целого типа – координаты вершин треугольника.

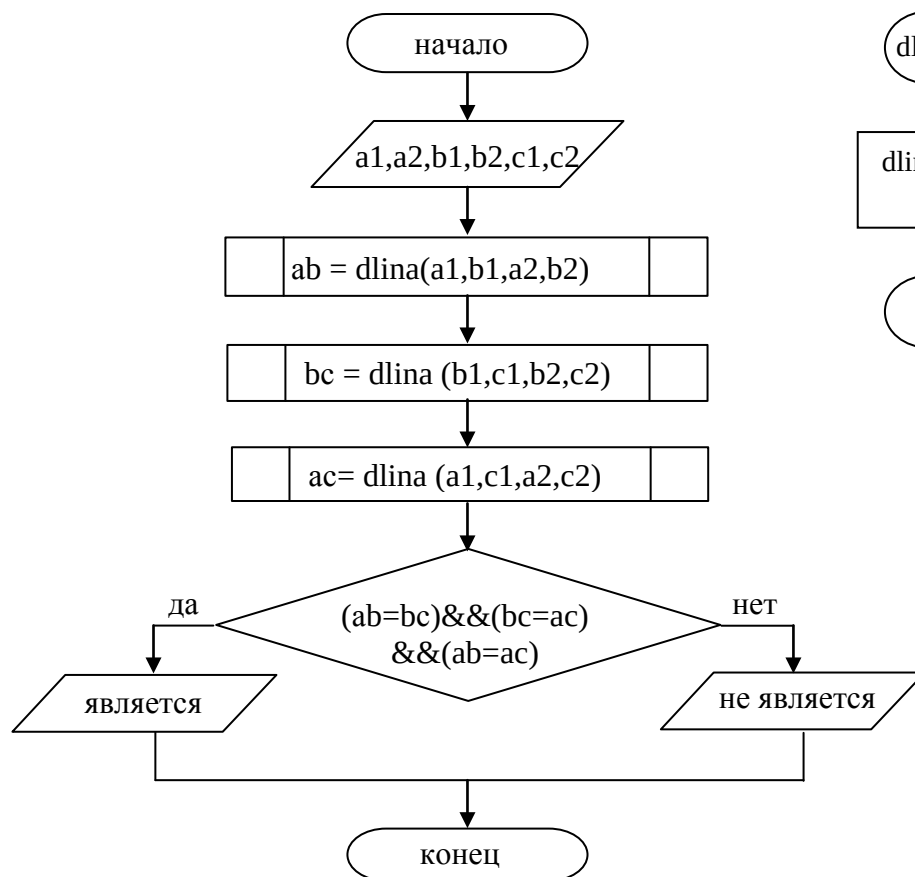
Результаты: текстовое сообщение.

Промежуточные величины: ab, bc, ac вещественного типа – длины сторон треугольника.

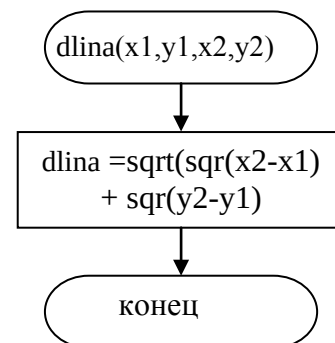
Расстояние между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ находим по формуле:

$$AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

2. **Алгоритм**– основная программа:



– подпрограмма:



3. **Программа**

4. **Результат работы программы:**

Лабораторная работа №6

Программирование рекурсивных алгоритмов

ЦЕЛЬ РАБОТЫ:закрепление навыков программирования с использованием рекурсивных функций.

Теоретическая часть

Итерация и рекурсия

6.1. Понятие итеративного процесса

Итеративный процесс можно определить как процесс вычислений, основанный на повторении последовательности операций, при котором на каждом шаге повторения используется результат предыдущего шага. Примерами итеративных алгоритмов могут служить алгоритмы вычисления факториала натурального числа, последовательности чисел Фибоначчи, вычисление суммы членов последовательности с заданным числом слагаемых и т.д. Итеративные процессы программируются обычно с помощью циклических конструкций.

Пример. Вычисление факториала числа n .

```
intfactor (intn)
{
    intf=1;
    for(inti = 2; i<=n; i++)
        f *= i;
    returnf;
}
```

6.2. Понятие рекурсии

В то же время повторение последовательности действий можно осуществить и другим методом, основанным на принципе возможности явного включения в конструкции языка программирования конструкции того же вида. В частности, в языке программирования JAVA всякая функция может вызывать сама себя. Такой вызов называется **рекурсией**. Функция, вызывающая саму себя, называется **рекурсивной**.

Вообще алгоритм решения задачи называется *рекурсивным*, если он выражается в терминах решения более простой версии той же задачи и, кроме того, в нем предусмотрено решение самого простого варианта задачи.

При рекурсивном вызове функций в JAVA используется стековая организация хранения локальных параметров.

Стек является простейшей динамической структурой организации данных. Добавление элементов в стек и выборка из него выполняются из одного конца, называемого вершиной стека. Говорят, что стек реализует принцип обслуживания LIFO (lastin–firstout, последним пришел - первым ушел (обслужен)).

Стеки широко применяются в системном программном обеспечении, компиляторах, в различных рекурсивных алгоритмах.

Различия между итерацией и рекурсией:

- 1) итерации необходим цикл и вспомогательные величины, в то время как рекурсия обходится без вспомогательных величин и обычно проще для понимания, короче и нагляднее итерации;
- 2) итерация требует меньше места в оперативной памяти и меньших затрат машинного времени, чем рекурсия, которой необходимы затраты на управление стеком;
- 3) при реализации рекурсии велика вероятность переполнения стека.

Поэтому при выборе способа описания функции следует вначале решить, чему отдать предпочтение – эффективности программы или ее компактности. Однако **следует избегать рекурсий там, где есть очевидное итерационное решение**. Вместе с тем существует много хороших примеров применения рекурсии (построение графических образов разнообразных узоров, работа с двоичными деревьями).

При рекурсивном описании функции последняя всегда должна содержать, по крайней мере, одну альтернативную нерекурсивную ветвь алгоритма с условием окончания вычислений, заканчивающуюся оператором возврата результата.

Различают две формы рекурсивных подпрограмм:

- **прямая рекурсия;**
- **косвенная рекурсия.**

В первом случае функция содержит вызов этой же функции.

Косвенная рекурсия возникает в том случае, когда две или более функций вызывают друг друга. В случае двух функций: некоторая функция обращается к другой функции, а та в свою очередь вызывает первоначальную функцию. Для предотвращения закливания одна или обе функции должны проверять условие завершения.

Пример: Составить рекурсивную функцию, вычисляющую факториал числа n по формуле (учесть, что $0! = 1$):

$$n! = \begin{cases} 1, & n = 1; \\ n \cdot (n-1)!, & n > 1. \end{cases}$$

```
long fact (long n)
{
    if(n==0 || n == 1) return 1;
    return (n*fact(n-1));
}
```

То же самое можно записать короче:

```
long fact (long n)
{
    return (n > 1) ? n*fact(n-1) : 1;
}
```

Пример: Распечатать последовательность, состоящую из n букв А и n букв В.


```

void abn (int n)
{
    cout<<"A";
    if(n > 1)abn(n-1);
    cout<<"B";
    return
}

```

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: в соответствии с вариантом, используя прямую рекурсию, составить и реализовать программу.

Задание

1. Описать рекурсивную функцию $NOD(A, B)$ целого типа, находящую наибольший общий делитель (НОД) двух целых положительных чисел A и B , используя алгоритм Евклида: $НОД(A, B) = НОД(B, A \bmod B)$, если $B \neq 0$; $НОД(A, 0) = A$. С помощью этой функции найти $НОД(A, B)$, $НОД(A, C)$, $НОД(A, D)$, если даны числа A, B, C, D .
2. Для $n = 12$ найти числа Фибоначчи. Числа Фибоначчи: $F(0) = 1, F(1) = 1$
 $F(n) = F(n - 2) + F(n - 1)$
3. Даны целые числа m и n , где $0 \leq m \leq n$, вычислить, используя рекурсию, число сочетаний $C(n, m)$ по формуле: $C_n^0 = C_n^n = 1, C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$ при $0 \leq m \leq n$. Воспользовавшись формулой $C_n^m = \frac{n!}{m!(n-m)!}$ можно проверить правильность результата.
4. Опишите рекурсивную функцию, которая по заданным вещественному x и целому n вычисляет величину x^n согласно формуле:

$$x^n = \begin{cases} 1, & n = 0, \\ \frac{1}{x^{|n|}}, & n < 0, \\ x \cdot (x^{n-1}), & n > 0. \end{cases}$$

5. Задана последовательность положительных чисел, признаком конца которых служит отрицательное число. Используя рекурсию, подсчитать количество чисел и их сумму.
6. Напишите рекурсивную функцию $K(n)$, которая возвращает количество цифр в заданном натуральном числе n , используя формулу:

$$K(n) = \begin{cases} 1, & \text{если } n < 10, \\ K(n \div 10) + 1, & \text{если } n \geq 10. \end{cases}$$

7. Определим функцию $S(n)$, вычисляющую сумму цифр заданного натурального числа:

$$S(n) = \begin{cases} n, & \text{если } n < 10, \\ S(n \operatorname{div} 10) + n \bmod 10, & \text{если } n \geq 10. \end{cases}$$

Описать рекурсивную функцию DigitSum(K) целого типа, которая находит сумму цифр целого числа K, не используя оператор цикла. С помощью этой функции найти суммы цифр для пяти данных целых чисел.

8. Дан вектор X из n вещественных чисел. Найти минимальный элемент вектора, используя вспомогательную рекурсивную функцию, находящую минимум среди элементов вектора X, начиная с n-го.
9. Напишите рекурсивную функцию для нахождения биномиальных коэффициентов (для заданного $M \geq i \geq j > 0$ вычислите все C_i^j):

$$C_n^m = \begin{cases} 1, & \text{если } n = m \text{ или } m = 0; \\ C_{n-1}^{m-1} + C_{n-1}^m. \end{cases}$$

10. Напишите программу вычисления функции Аккермана для всех неотрицательных целых аргументов m и n:

$$A(m, n) = \begin{cases} A(0, n) = n + 1, \\ A(m, 0) = A(m - 1, 1), m > 0, \\ A(m, n) = A(m - 1, A(m, n - 1)), m, n > 0. \end{cases}$$

11. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int e^{ax} \cos^n x dx = \begin{cases} \frac{e^{ax} \cos^{n-1} x (a \cos x + n \sin x)}{a^2 + n^2} + \frac{n(n-1)}{a^2 + n^2} \int e^{ax} \cos^{n-2} x dx, n \geq 2, \\ \frac{-e^{ax} (\sin x + a \cos x)}{a^2 + n^2}, n = 1, \\ \frac{e^{an}}{a}, n = 0. \end{cases}$$

12. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \cos^n x dx = \begin{cases} \frac{\cos^{n-1} x \sin x}{n} + \frac{n-1}{n} \int \cos^{n-2} x dx, n > 2, \\ \frac{x}{2} + \frac{1}{4} \sin 2x, n = 2, \\ \sin x, n = 1. \end{cases}$$

13. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \sin^n x dx = \begin{cases} -\frac{\sin^{n-1} x \cos x}{n} + \frac{n-1}{n} \int \sin^{n-2} x dx, n > 2, \\ \frac{x}{2} - \frac{1}{4} \sin 2x, n = 2, \\ -\cos x, n = 1. \end{cases}$$

14. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \frac{dx}{\sin^n x} = \begin{cases} -\frac{1}{n-1} \cdot \frac{\cos x}{\sin^{n-1} x} + \frac{n-2}{n-1} \int \frac{dx}{\sin^{n-2} x}, n \geq 2, \\ \ln \operatorname{tg} \frac{x}{2}, n = 1, \\ x, n = 0; \end{cases}$$

15. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \frac{dx}{\cos^n x} = \begin{cases} \frac{1}{n-1} \cdot \frac{\sin x}{\cos^{n-1} x} + \frac{n-2}{n-1} \int \frac{dx}{\cos^{n-2} x}, n \geq 2, \\ \ln \operatorname{tg} \left(\frac{\pi}{4} + \frac{x}{2} \right), n = 1, \\ x, n = 0. \end{cases}$$

16. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int x^n e^{ax} dx = \begin{cases} \frac{x^n e^{ax}}{a} - \frac{n}{a} \int x^{n-1} e^{ax} dx, n > 1, \\ \frac{e^{ax}}{a^2} (ax - 1), n = 1. \end{cases}$$

17. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int x^n a^{mx} dx = \begin{cases} \frac{x^n a^{mx}}{n \ln a} - \frac{n}{m \ln a} \int x^{n-1} a^{mx} dx, n > 1, \\ \frac{x a^{mx}}{m \ln a} - \frac{a^{mx}}{m (\ln a)^2}, n = 1. \end{cases}$$

18. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \ln^n x dx = \begin{cases} x \ln^n x - n \int \ln^{n-1} x dx, & n > 1, \\ x \ln x - x, & n = 1. \end{cases}$$

19. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int x^m \ln^n x dx = \begin{cases} \frac{x^{m+1}}{m+1} \ln^n x - \frac{n}{m+1} \int x^m \ln^{n-1} x dx, & n > 1, \\ x^{m+1} \left[\frac{\ln x}{m+1} - \frac{1}{(m+1)^2} \right], & n = 1. \end{cases}$$

20. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \frac{dx}{(a^2 + x^2)^n} = \begin{cases} \frac{1}{2(n-1)a^2} \left[\frac{x}{(a^2 + x^2)^{n-1}} + (2n-3) \int \frac{dx}{(a^2 + x^2)^{n-1}} \right], & n > 1, \\ \frac{1}{a} \operatorname{arctg} \frac{x}{a}, & n = 1. \end{cases}$$

21. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \operatorname{tg}^n x dx = \begin{cases} \frac{\operatorname{tg}^{n-1} x}{n-1} - \int \operatorname{tg}^{n-2} x dx, & n \geq 2, \\ -\ln |\cos x|, & n = 1, \\ x, & n = 0. \end{cases}$$

22. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \operatorname{ctg}^n x dx = \begin{cases} -\frac{\operatorname{ctg}^{n-1} x}{n-1} - \int \operatorname{ctg}^{n-2} x dx, & n \geq 2, \\ \ln |\sin x|, & n = 1, \\ x, & n = 0. \end{cases}$$

23. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int x^m \sin ax = \begin{cases} -\frac{x^m}{a} \cos ax + \frac{m}{a} \int x^{m-1} \cos ax dx, & m \geq 1, \\ -\frac{\cos ax}{a}, & m = 0. \end{cases}$$

24. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int x^m \cos ax = \begin{cases} -\frac{x^m}{a} \sin ax + \frac{m}{a} \int x^{m-1} \sin ax dx, m \geq 1, \\ -\frac{\sin ax}{a}, m = 0. \end{cases}$$

25. Напишите рекурсивную функцию, перемножающую два целых числа, одно из которых неотрицательно, без использования операции умножения.

Контрольные вопросы

1. Какой процесс называется итеративным?
2. Каким образом реализуются итеративные процессы?
3. Какой алгоритм называется рекурсивным?
4. Какая функция называется прямо рекурсивной? Косвенно рекурсивной?
5. Что такое стек? В какой последовательности происходит заполнение стека и выбор элементов из стека?
6. Что должно обязательно присутствовать в теле рекурсивно описанной функции?
7. Перечислите различия между итерацией и рекурсией.
8. Что произойдет, если рекурсивный алгоритм будет вызывать сам себя «бесконечное» число раз?
9. Как предотвратить бесконечное выполнение рекурсивного алгоритма?
10. Верно ли, что решение задачи, реализуемое рекурсивным алгоритмом, можно выразить, используя итерацию?

Пример выполнения Практической работы

Задание. Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int x^n e^{cx} dx = \begin{cases} \frac{x^n e^{cx}}{c} - \frac{n}{c} \int x^{n-1} e^{cx} dx, n > 1, \\ \frac{e^{cx}}{c^2} (cx - 1), n = 1; \end{cases}$$

Решение

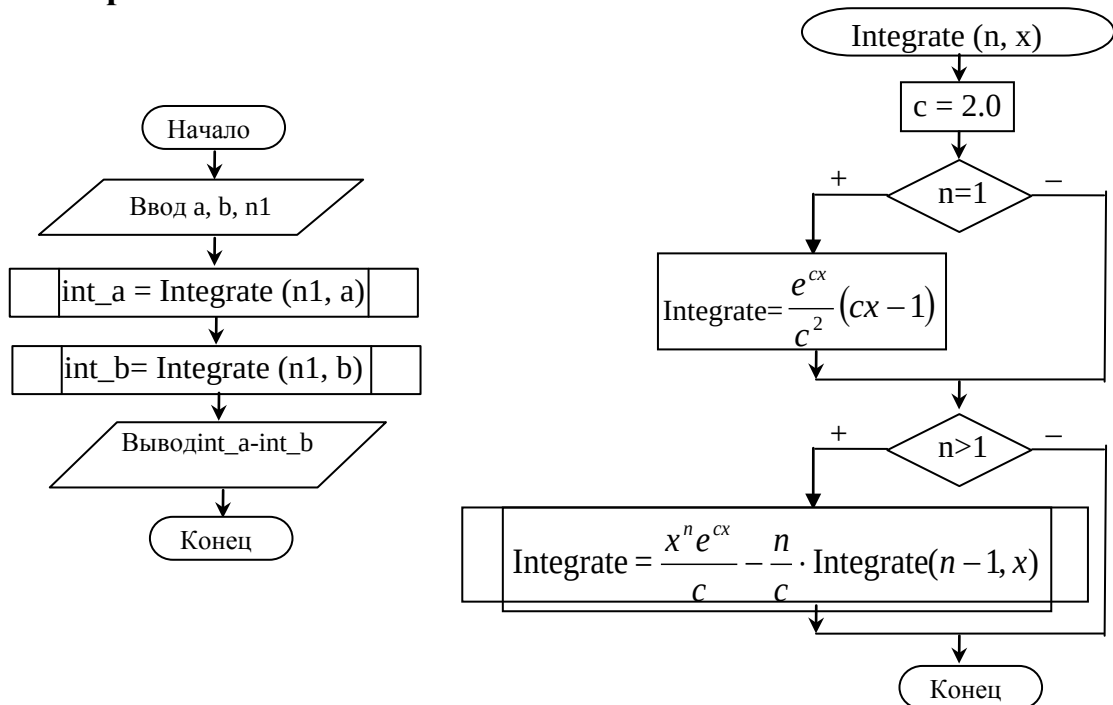
1. Математическая модель

Аргументы: границы интегрирования a , b целого типа;
степень n целого типа.

Результаты: разность $\text{int_b} - \text{int_a}$ вещественного типа.

Промежуточные величины: значение интеграла для нижней границы интегрирования int_a; значение интеграла для верхней границы интегрирования int_b (вещественного типа).

2. Алгоритм



3. Программа

4. Результат работы программы

Введите a и b, где b больше a: 2 4

Введите степень n: 3

Значение определенного интеграла при a = 2 и b = 4 равно 67328,1534

Лабораторная работа №7

Использование числовых одномерных массивов

ЦЕЛЬ РАБОТЫ: освоение способов описания массива, приобретение навыков организации ввода-вывода и обработки массива.

Теоретическая часть

Одномерные массивы

7.1. Понятие структурированного типа данных

Синонимами названия «базовый тип» являются названия «стандартные», «предопределенные». Реализация этих типов заложена в стандарте языка, и программист не вправе изменить предопределенное. Производным типом (определяемым программистом) является такой тип, который должен быть описан пользователем перед употреблением. К производным типам можно отнести массивы, перечисления, функции, структуры, ссылки, объединения и классы. Полезно помнить, что тип данного определяет его размещение в памяти и набор операций.

Структурированный тип представляет собой набор элементов – данных, объединенных программистом в единый блок в соответствии с особенностями решаемой задачи. Данные, входящие в блок, могут быть однотипными или разнотипными, весь блок данных имеет общее имя. Каждый элемент отличается от других либо порядковым номером (индексом) – в массиве, либо своим именем и, возможно, типом – в структуре.

Любой из структурированных типов данных характеризуется множественностью образующих этот тип элементов. Переменная или константа структурированного типа всегда имеет несколько компонент. Каждая из этих компонент, в свою очередь, может принадлежать структурированному типу, что позволяет говорить о возможной вложенности типов.

В JAVA можно выделить следующие структурированные типы: массивы, структуры, объединения, классы, файлы.

7.2. Понятие и описание типа массив

При использовании простых переменных каждой области памяти для хранения данных соответствует своё имя. Если с группой величин одинакового типа требуется выполнять однообразные действия, им дают одно имя, а различают по порядковому номеру. Это позволяет компактно записывать множество операций с помощью циклов. Конечная именованная последовательность однотипных величин называется **массивом**. Описание массива в программе отличается от описания простой переменной наличием после имени квадратных скобок, в которых задается количество элементов массива (размер).

Общий вид описания массива:

тип_данных имя_массива [количество_элементов 1] ... [количество_элементов n] ;

Здесь:

1) `тип_данных` – это тип элементов массива. Это может быть любой стандартный или ранее определённый пользователем тип данных. Например, можно определить массив координат точек на плоскости, тогда один элемент массива содержит две координаты одной точки. Можно определить массив сведений о студенте, тогда один элемент массива содержит разнообразные сведения о студенте, например, фамилия, имя, возраст, адрес и прочие. В этих случаях для определения типа одного элемента массива используется также производный тип, например, массив или структура;

2) `имя_массива` – правильный идентификатор;

3) `количество_элементов` – это константа, заданная явно или именованная (константное выражение). **Использовать переменные нельзя!;**

4) `n` – размерность массива.

При `n = 1` описание соответствует одномерному массиву (вектору),

`n = 2` – двумерному массиву (матрице),

`n > 2` – многомерный массив.

На размерность массивов ограничений не налагается.

Размерность массива всегда равна количеству индексов (измерений).

Под **размером** обычно понимают количество элементов массива.

Пример фиксированного массива на JAVA

```
float a[10]; // описание одномерного массива из 10 вещественных чисел
int Array[10]; // одномерный массив целых чисел с размером 10
// нумерация элементов от 0 до 9
double Array[12][15]; // Двумерный массив вещественных чисел двойной
// точности размера 12 на 15. Нумерация
// по строкам от 0 до 11, по столбцам от 0 до 14
```

Замечание. При описании массивов квадратные скобки являются элементом синтаксиса, а не указанием на необязательность конструкции.

Элементы массива нумеруются с нуля. Инициализирующие значения для массивов записываются в фигурных скобках. Значения элементам присваиваются по порядку. Если элементов в массиве больше, элементы, для которых значения не указаны, обнуляются:

```
int b[5] = {3, 2, 1}; // b[0] = 3, b[1] = 2, b[2] = 1, b[3] = 0, b[4] = 0
```

Замечание. К каждому элементу массива имеется прямой доступ. Для этого необходимо указать имя массива и номер элемента (индекс) в квадратных скобках.

Размер массива вместе с типом его элементов определяет объем памяти, необходимый для размещения массива, которое выполняется на этапе компиляции, поэтому размер может быть задан только целой положительной константой или константным выражением. Если при описании массива не указан размер, должен присутствовать инициализатор, в этом случае компилятор выделит память по количеству инициализирующих значений.

7.3. Одномерные массивы

Одномерный массив – массив, с одним параметром (измерением), характеризующим количество элементов одномерного массива. **Размерность $n = 1$.**

На рисунке 7.1 показана структура целочисленного одномерного массива a . Размер этого массива – 10 элементов (ячеек).

-5	2	0	-6	12	56	8	23	4	-3
$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$	$a[8]$	$a[9]$

Рисунок 7.1 – Одномерный массив в JAVA

Максимальный индекс одномерного массива a равен 9, но размер массива 10 ячеек, так как нумерация ячеек массива всегда начинается с 0. Индекс ячейки – это целое неотрицательное число, по которому можно обращаться к каждой ячейке массива и выполнять какие-либо действия над ней (ячейкой).

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: освоить теоретический материал, выполнить общее для всех задание I и в соответствии с вариантом составить программу (задание II), при необходимости реализовав пользовательскую функцию.

Задание

В соответствии с вариантом составить и реализовать программу:

1. Даны два массива разных размеров. Определить, какие элементы первого массива и сколько раз встречаются во втором массиве.
2. Массив содержит $2n$ чисел. Из суммы первых n его элементов вычесть сумму последних n элементов.
3. Транспонировать массив, т.е. по $a_1, a_2, \dots, a_n$ сформировать $a_n, a_{n-1}, \dots, a_1$. В полученном массиве найти индекс минимального элемента.
4. Из заданного целочисленного массива удалить все повторяющиеся элементы, оставив только их первые вхождения, т.е. из заданного массива получить новый массив, состоящий из различных целых чисел.
5. Заменить отрицательные числа в массиве их квадратами, оставив остальные без изменения.
6. В заданном массиве найти среднее арифметическое положительных чисел, среднее арифметическое отрицательных чисел и число нулей.
7. В массиве из $2n$ чисел найти сумму квадратов элементов с четными индексами и сумму кубов элементов с нечетными индексами.
8. Из чисел $a_1, a_2, \dots, a_n$ выбрать те, которые больше по модулю заданного числа c , и образовать из них новый массив, сохранив порядок следования элементов.

9. Из массива целых чисел составить три других, в первый из которых записать числа, кратные 5, во второй - числа, кратные 7, а в третий - остальные числа.
10. Задан массив из 100 целых случайных чисел, принадлежащих промежутку $[0, 100]$. Найти сумму тех элементов массива, которые больше 15, но меньше 45, а также вычислить количество этих элементов.
11. В линейном массиве заменить все элементы на число m (m – индекс максимального элемента).
12. Дан массив, состоящий как из положительных, так и отрицательных чисел. Нужно сначала записать положительные числа, а затем отрицательные в том же порядке, как они были расположены в исходном массиве. Если есть нули, записать их в последнюю очередь.
13. Найти сумму элементов данного массива. Разделить каждый элемент исходного массива на полученное значение.
14. Вычислить сумму и разность массивов одного размера.
15. Найти среднее арифметическое значение элементов заданного массива. Преобразовать исходный массив, вычитая из каждого элемента среднее значение.
16. Даны два массива одинакового размера. Рассматривая их как арифметические векторы, найти длины этих векторов и их скалярное произведение.
17. Заданы два массива разных размеров. Объединить их в один массив, включив второй массив между k -ым и $(k + 1)$ -ым элементами первого (вводится с клавиатуры).
18. Вычесть из положительных элементов данного массива элемент с номером k_1 а к отрицательным элементам прибавить элемент с номером k_2 . Нулевые элементы заменить 1. Номера k_1 и k_2 вводятся с клавиатуры.
19. К четным элементам целочисленного массива прибавить данное число a , а из элементов с четными номерами вычесть данное число b .
20. Дан первый член геометрической прогрессии и ее знаменатель. Сформировать одномерный массив, элементами которого служат первые n членов этой прогрессии.
21. Сформировать массив из первых 30 членов последовательности Фибоначчи.
22. Вставить одно и то же число, введенное с клавиатуры, перед каждым отрицательным элементом заданного целочисленного массива.
23. Дан массив четного размера. Поменять местами его половины следующим образом: первый элемент - с последним, второй - с предпоследним элементом и т.д.
24. Даны два целочисленных массива одинакового размера. Получить третий массив того же размера, каждый элемент которого равен большему из соответствующих элементов данных массивов.

25. Задан массив из n целых случайных чисел, принадлежащих промежутку $[-25, 25]$. Найти произведение тех элементов массива, которые больше 1, но меньше 15, а также вычислить количество четных элементов массива.

Контрольные вопросы

1. Дайте определение производного типа данных, структурированного типа.
2. Дайте определение массива.
3. Каким может быть тип элементов массива?
4. Имя, размер и размерность массива.
5. Какова структура одномерного массива?
6. Правила описания одномерного массива.
7. Как осуществляется доступ к элементам одномерного массива?
8. Как осуществляется ввод массива?
9. Какие способы ввода массива вы знаете?
10. Вывод линейного массива.

Пример выполнения задания II Практической работы

Задание. Массив D содержит 24 значения атмосферного давления за каждый час в течение суток. Определить, какое значение атмосферного давления было наибольшим и в какое время оно было зафиксировано.

Решение

1. Математическая модель

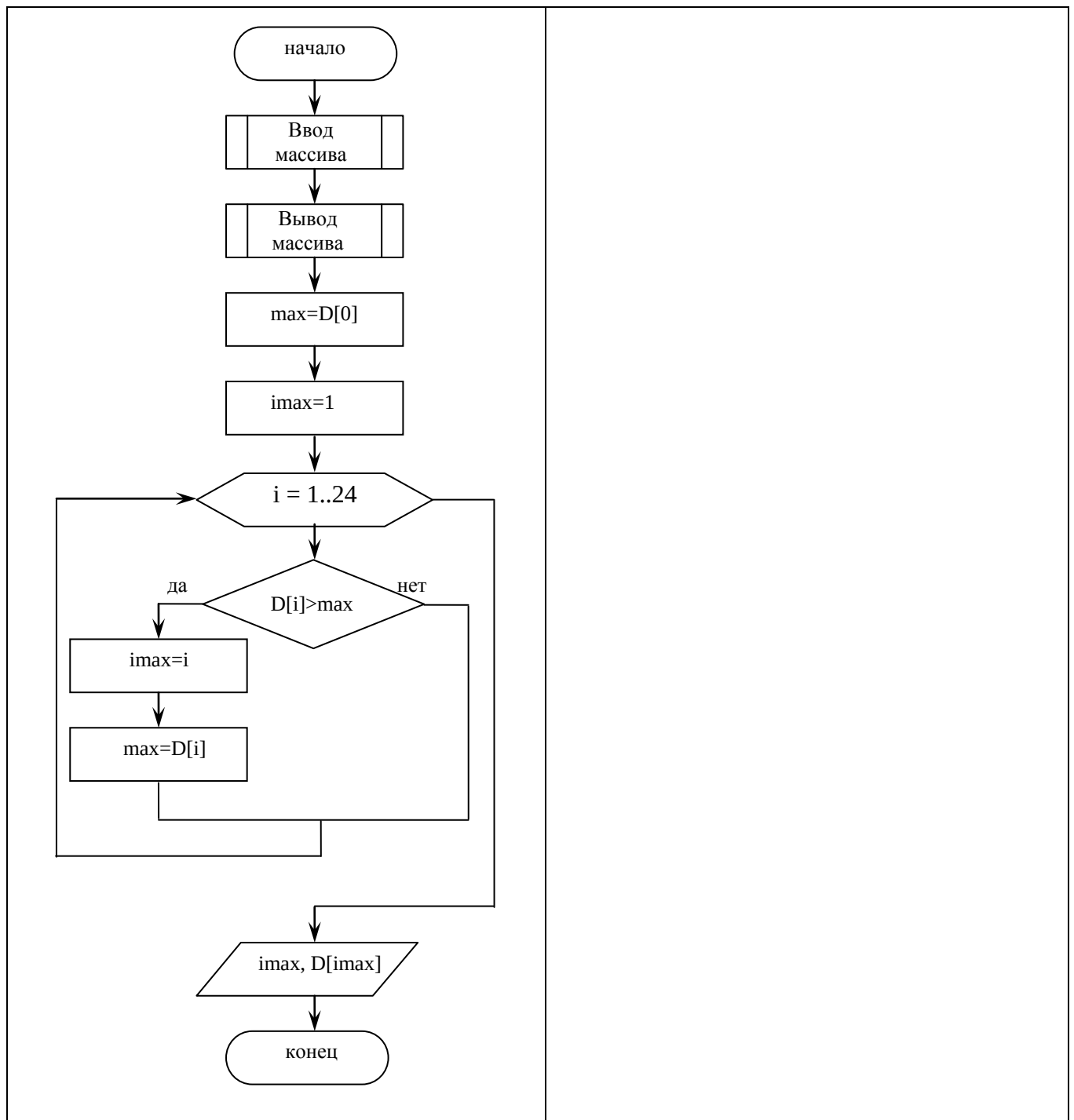
Значением атмосферного давления являются элементы массива $D[24]$, значением времени – индексы элементов. Решение задачи сводится к поиску максимального элемента в массиве и определению его индекса.

Аргументы: $D[24]$ – массив целого типа.

Результаты: $i_{\max}$ целого типа – индекс максимального элемента;
 $D[i_{\max}]$ целого типа – значение максимального элемента.

Промежуточная величина: $\max$ целого типа – максимальное значение элемента массива.

2. Алгоритм	3. Программа
-------------	--------------



4. Результат работы программы:

Массив значений давления:

748 743 756 748 741 760 748 743 757 748 756 756 751 760 754
756 752 758 758 757 750 747 759 750

Максимальное значение давления 760 в 6 ч.

Лабораторная работа №8

Двумерные массивы

ЦЕЛЬ РАБОТЫ:приобретение навыков объявления, организации ввода-вывода и обработки двумерного массива.

Теоретическая часть

Двумерные массивы

8.1 Двумерные массивы

Итак, **массив** представляет собой структуру данных, позволяющую хранить под одним именем совокупность данных любого, но только одного какого-то типа. Массив характеризуется своим именем, типом хранимых элементов, размером (числом хранимых элементов), нумерацией элементов и размерностью.

Объявление переменной как одномерного массива (массива с размерностью 1) имеет вид:

тип_данныхидентификатор [константное_выражение];

Например, оператор

`intA[10];`//объявляет массив с именем A, содержащий 10 целых чисел

Доступ к элементам этого массива осуществляется выражением $A[i]$, где i - индекс, являющийся в данном примере целым числом в диапазоне 0 - 9. Например, $A[0]$ - значение первого элемента, $A[1]$ - второго, $A[9]$ - последнего.

Напомним, что объявление переменной массива можно совмещать с заданием элементам массива начальных значений. Например, оператор:

`intA[10] = {0};`//присваивает нулевые значения всем элементам массива
`intA[10] = {1,2,3,4,5,6,7,8,9,10};`// $A[0] = 1$, $A[2] = 2$, ..., $A[9] = 10$

Если начальных значений меньше, чем элементов в массиве, оставшиеся элементы автоматически получают нулевые начальные значения.

Если массив при его объявлении не инициализирован, то его элементы имеют случайные значения. Элементы такого массива нельзя использовать в выражениях, пока им не будут присвоены какие-нибудь значения.

Можно объявлять многомерные массивы, т.е. массивы, элементами которых являются массивы. Например, двумерный массив можно объявить таким образом:

`intA2[10][3];`

Этот оператор описывает двумерный массив, который можно представить как таблицу, состоящую из 10 строк и 3 столбцов.

Таким образом, **двумерный массив** – это массив, размерность которого равна 2. Визуально, двумерный массив – это обычная таблица, со строками и столбцами или матрица $A = \{a_{ij}\}_{m \times n}$. Фактически двумерный массив – это одномерный массив одномерных массивов. Структура двумерного массива, с именем A, размером m на n показана ниже (см. рис.8.1).

$A[0][0]$	$A[0][1]$	...	$A[0][n-1]$	$\left(\begin{array}{cccc} a_{00} & a_{01} & \dots & a_{0n-1} \end{array} \right)$
-----------	-----------	-----	-------------	---

A[1][0]	A[1][1]	...	A[1][n-1]
A[2][0]	A[2][1]	...	A[2][n-1]
...	...	...	...
A[m-1][0]	A[m-1][1]	...	A[m-1][n-1]

Рисунок 8.1– Двумерный массив в виде таблицы (слева)или матрицы (справа)

На рисунке 8.1 m – количество строк двумерного массива;

n – количество столбцов двумерного массива;

$m * n$ – количество элементов массива.

Общий вид описания двумерного массива:

тип_данных имя_массива [количество_строк][количество_столбцов];

В объявлении двумерного массива, также как и в объявлении одномерного массива, первым делом, нужно указать:

- тип данных;
- имя массива.

Доступ к значениям элементов многомерного массива обеспечивается через индексы, каждый из которых заключается в квадратные скобки. Например, A2[3][2] - значение элемента, лежащего на пересечении четвертой строки и третьего столбца (помните, что индексы начинаются с 0).

Если многомерный массив инициализируется при его объявлении, список значений по каждой размерности заключается в фигурные скобки. Приведенный ниже оператор объявляет трехмерный массив A3 размерностью 4 на 3 на 2.

```
int A3[4][3][2] = {{{0,1},{2,3},{4,5}},
                  {{6,7},{8,9},{10,11}},
                  {{12,13},{14,15},{16,17}},
                  {{18,19},{20,21},{22,23}}};
```

Этот оператор создает массив A3, четыре строки которого являются матрицами вида:

0	1	6	7	12	13	18	19
2	3	8	9	14	15	20	21
4	5	10	11	16	17	22	23

Например, элемент A3[0][1][0] равен 2, элемент A3[3][0][1] равен 19 и т.д.

Если в списке инициализации в какой-то из размерностей не хватает данных, то все дальнейшие не перечисленные элементы считаются равными нулям.

Все размерности массива должны быть константами или константными выражениями, поскольку инструкции по выделению памяти формируются компилятором до выполнения программы.

В памяти многомерный массив располагается по строкам.

Строки массива ничем не отделены одна от другой, то есть, например, двумерный массив является прямоугольной матрицей только в нашем воображении. При просмотре массива от начала в первую очередь изменяется правый индекс (номер столбца).

К элементу двумерного массива обращаются, указывая номер строки и номер столбца, на пересечении которых он расположен, например:

`a[1][4]` `a[i][j]`

Компилятор воспринимает как номер строки первый индекс, как бы он ни был обозначен в программе.

1. **Инициализация** двумерных массивов аналогична инициализации одномерных массивов:

а) инициализация:

```
const int n = 3; // число строк
const int m = 4; // число столбцов
// выделяем память под матрицу с одновременной
// инициализацией элементов
double a[n][m] =
{
    {3, -3.1, 4.6, 5.5},
    {2, 1.2, -10, -2.5},
    {7.6, 3.1, 1.6, 0.5}
};
```

При инициализации двумерного массива в ходе работы программы, необходимо организовать вложенные циклы:

б) ввод данных с клавиатуры:

```
for(i = 0; i < n; i++)
    for(j = 0; j < m; j++)
        cin >> a[i][j];
```

В соответствии с приведенным здесь порядком следования циклов элементы массива должны вводиться по строкам.

в) формирование массива с помощью генератора псевдослучайных чисел;

г) непосредственное присваивание (вычисление по формуле):

2. **Вывод** матрицы надо реализовать в удобном для чтения виде, т.е. чтобы на одной строке экрана располагалась одна строка матрицы. С этой целью в тело внешнего цикла, помимо внутреннего, включается еще оператор `cout << endl;`, который переводит курсор к началу следующей строки экрана после вывода текущей строки матрицы.

```
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
        cout << a[i][j] << "\t ";
    cout << endl;
}
```

Пример. Дана прямоугольная матрица действительных чисел размером $n \times m$. Пронормировать эту матрицу, т.е. поделить значение всех элементов матрицы на максимальный по модулю элемент.

Возможный текст программы:

```

#include<iostream>
#include<cmath>

using namespace std;

int main()
{
    const int n = 3; // число строк
    const int m = 4; // число столбцов
    double a[n][m]; // выделяем память под матрицу
    int i, j;
    double max;

    // Ввод матрицы с клавиатуры
    cout<< "Matriz A("<< n << "*" << m << "):" << endl;
    for(i = 0; i < n; i++)
        for(j = 0; j < m; j++)
            cin>>a[i][j];

    // Поиск в матрице максимального по модулю значения
    max = fabs(a[0][0]);
    for(i = 0; i < n; i++)
        for(j = 0; j < m; j++)
            if(fabs(a[i][j]) > max)
                max = fabs(a[i][j]);
    cout<< "max=" << max<< endl;

    // Нормируем матрицу
    for(i = 0; i < n; i++)
        for(j = 0; j < m; j++)
            a[i][j] /= max;

    // Вывод матрицы в виде таблицы на экран монитора
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < m; j++)
            cout<< a[i][j] << "\t ";
        cout<< endl;
    }
    return 0;
}

```

Как видно из текста программы, для работы с матрицей почти всегда требуются двойные циклы. Особое внимание следует обратить на вывод двумерного массива в виде таблицы.

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: в соответствии с вариантом составить и реализовать программу.

Задание I

1. Составить программу формирования матрицы, число строк и столбцов которой определяется как случайное натуральное число в промежутке $[3, 8]$, а элементы - случайные целые числа от -50 до 50. Заменить неотрицательные числа полученной матрицы единицей, а отрицательные - нулем.
2. В целочисленной прямоугольной матрице числа, кратные заданному числу m , вводимому с клавиатуры, заменить частным от деления на m , а остальные числа заменить их остатками от деления на m .
3. В квадратной матрице порядка n найти сумму всех тех элементов, сумма индексов которых равна n .
4. Определить, является ли заданная квадратная матрица симметричной относительно главной диагонали. Найти след данной матрицы. Элементы матрицы вводить с клавиатуры.
5. Найти число элементов в каждой из нечетных строк матрицы, равных заданному числу, вводимому с клавиатуры.
6. Преобразовать данную прямоугольную матрицу так, чтобы все элементы в ее строках были записаны в обратном порядке.
7. Вычислить количество и сумму положительных элементов в нижнем левом треугольнике данной квадратной матрицы, включая диагональные элементы.
8. Даны вещественные числа $a_1, a_2, \dots, a_n$ и вещественная квадратная матрица порядка n ($n > 5$). Сформировать вещественную матрицу размером $n \times (n + 1)$, вставив в исходную матрицу между пятым и шестым столбцами новый столбец с элементами $a_1, a_2, \dots, a_n$.
9. Дана вещественная квадратная матрица порядка n . Преобразовать эту матрицу по правилу: строку с номером k сделать столбцом с номером k , а столбец с номером k сделать строкой с тем же номером.
10. В данной вещественной матрице размером $n \times m$ ($n > 2, m > 2$) поменять местами строки с номерами $n - 2$ и $n - 1$.
11. Положительные элементы первой строки прямоугольной матрицы умножить на первый элемент этой же строки, а отрицательные - на последний ее элемент; то же самое проделать с остальными строками.
12. Заменить все элементы строки с номером s и столбца с номером k прямоугольной матрицы на противоположные по знаку (элемент, стоящий на пересечении выбранной строки и выбранного столбца, не изменять).

13. Из квадратной матрицы получить другую матрицу путем исключения диагональных элементов и вычитания из каждого элемента следа исходной матрицы.
14. Дан целочисленный одномерный массив: $a_1, a_2, \dots, a_n$. Сформировать квадратную матрицу порядка n , у которой элементы, расположенные на главной диагонали, равны соответственно $a_1, a_2, \dots, a_n$; элементы, расположенные выше главной диагонали, равны единице, а элементы, расположенные ниже главной диагонали, равны минус единице.
15. Даны две прямоугольные целочисленные матрицы A и B одинакового размера $m \times n$. Создать матрицу того же размера, в которой элементы равны 1, если соответствующие элементы матриц A и B имеют одинаковый знак; -1, если соответствующие элементы имеют разные знаки, и нулю, если хотя бы один из соответствующих элементов равен нулю.
16. Составить программу вывода на экран арифметического квадрата: в нем первый столбец и первая строка заполнены единицами, а каждый из остальных элементов равен сумме своих соседей сверху и слева.
17. Из прямоугольной матрицы размером $m \times n$ удалить строку с номером s и столбец с номером k ($0 \leq s < m, 0 \leq k < n$).
18. Сформировать одномерный массив, каждый элемент которого равен сумме отрицательных элементов соответствующей строки данной целочисленной прямоугольной матрицы.
19. Найти количество элементов в каждом столбце прямоугольной вещественной матрицы, меньших среднего арифметического элементов рассматриваемого столбца.
20. Вывести на экран максимальные элементы каждой строки двумерного массива с указанием их индексов.
21. Задать два двумерных массива, содержащих положительные и отрицательные элементы. Определить, в каком из них больше положительных элементов.
22. Задан массив целых случайных чисел, принадлежащих промежутку $[0, 200]$. Определить, каких элементов в нём больше: превышающих k_1 или не превышающих k_2 .
23. Задан массив целых случайных чисел, принадлежащих промежутку $[-50, 50]$. Найти сумму тех элементов массива, которые больше a , но меньше b , а также вычислить количество этих элементов.
24. Поменять местами максимальный и минимальный элемент заданного двумерного массива. Результат вывести на экран в виде матрицы.
25. Дана целочисленная матрица. Сформировать матрицу, получающуюся из данной перестановкой первого столбца с последним, второго - с предпоследним.

26. Из данной матрицы получить другую следующим образом: на место первой строки поместить вторую, на место второй - третью и т.д., на место последней поместить первую.

Задание II

Квадратная матрица A третьего порядка сформирована из вещественных чисел, принадлежащих диапазону $[-10, 10]$. Найти с точностью до $\varepsilon = 10^{-3}$:

- 1) матрицу B по заданной формуле вычисления ее элементов b_{ij} через индексы i и j ($i = 1, 2, 3; j = 1, 2, 3$);
- 2) сумму матриц A и B ;
- 3) матрицу, транспонированную матрице $A + B$;
- 4) матрицу, обратную матрице $A + B$;
- 5) произведение матриц $A + B$ и $(A + B)^T$.

Вывести, используя вспомогательную функцию, на экран матрицы A , B , $A + B$, $(A + B)^{-1}$, $(A + B)^T$, $(A + B)(A + B)^{-1}$.

1.

Замечание. Обратную матрицу рекомендуется вычислять с помощью присоединенной матрицы, а определитель матрицы – по правилу Саррюса.

Контрольные вопросы

1. Дайте определение массива.
2. Какие типы данных не допустимы для компонентов массива?
3. Может ли левая граница индексов массива быть отрицательной?
4. Какой массив называется двумерным?
5. Правила описания двумерного массива.
6. Как осуществляется доступ к элементам двумерного массива?
7. Как осуществляется ввод двумерного массива?
8. Какие способы ввода двумерного массива вы знаете?
9. Размещение массива в памяти ЭВМ.
10. Вывод двумерного массива.
11. Может ли инструкция `cout<<x`; ввести массив `x` целиком?

Пример выполнения Практической работы

Задание. Программа, которая для целочисленной матрицы 3×4 определяет среднее арифметическое ее элементов и количество положительных элементов в каждой строке.

Решение

1. Математическая модель

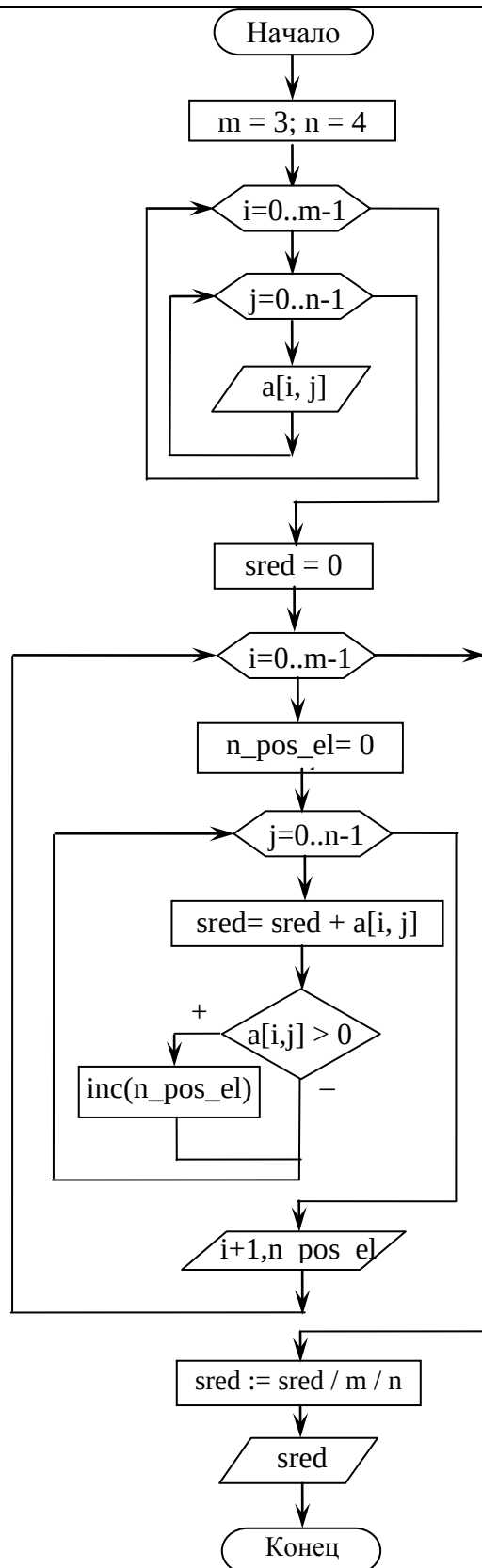
Для нахождения среднего арифметического элементов массива требуется найти их общую сумму, после чего разделить ее на количество элементов. Порядок перебора элементов массива (по строкам или по столбцам) роли не играет. Нахождение количества положительных элементов каждой строки требует просмотра матрицы по строкам.

Аргументы: целочисленный массив `a`; целочисленные `m` и `n` – количество строк и столбцов.

Результаты: среднее арифметическое элементов `sred` (вещественного типа); количество положительных элементов в каждой строке `pos_el` (целого типа).

Промежуточные величины: счетчики `i`, `j` (целого типа).

2. Алгоритм



3. Программа

4. Результат работы программы:

Введите элементы массива

1 2 3 4 -2 0 -3 -1 1 1 1 0

В 1-й строке 4 положительных элементов

В 2-й строке 0 положительных элементов

В 3-й строке 3 положительных элементов

Среднее арифметическое: 0.58

Задание для самостоятельного изучения

Задачи сортировки и поиска

ЦЕЛЬ: закрепление теоретических знаний и приобретение практических навыков по составлению алгоритмов и программ различных методов сортировки.

Теоретическая часть

Алгоритмы решения задач внутренней сортировки и алгоритмы поиска информации

9.1. Сложность алгоритмов

Теория сложности обеспечивает методологию анализа **вычислительной сложности** различных криптографических методов и алгоритмов.

Алгоритм – это формально описанная вычислительная процедура, получающая исходные данные, называемые также входом алгоритма или его аргументом, и выдающая результат вычислений на выход [4].

Чаще всего, говоря о сложности, имеют в виду *временную сложность алгоритма*: T . Временем работы алгоритма называется число элементарных шагов, которые он выполняет. Под элементарными шагами понимают такие базисные операции, как сложение, умножение, замены, безусловные передачи управления и вызовы подпрограмм. T обычно представляется в виде функции от n , где n – это размер входных данных. *Функция времени вычислений* – $T_A(n)$.

Обычно вычислительная сложность алгоритма выражается с помощью нотации "О большого", т.е. описывается порядком величины вычислительной сложности. Это просто член разложения функции сложности, быстрее всего растущий с ростом n , все члены низшего порядка игнорируются. Например, если временная сложность данного алгоритма равна $4n^2 + 7n + 2$, то вычислительная сложность порядка n^2 , записываемая как $O(n^2)$ [4].

Эта нотация позволяет увидеть, как объем входных данных влияет на требования к времени и объему памяти. Например, если $T = O(n)$, то удвоение входных данных удвоит и время выполнения алгоритма. Если $T = O(2^n)$, то добавление одного бита к входным данным удвоит время выполнения алгоритма.

Обычно алгоритмы классифицируются в соответствии с их временной или пространственной сложностью. Алгоритм называют **постоянным**, если его сложность не зависит от n : $O(1)$. Алгоритм является **линейным**, если его временная сложность $O(n)$. Алгоритмы могут быть **квадратичными**, **кубическими** и т.д. Все эти алгоритмы – **полиномиальные**, их сложность $O(n^m)$, где m – константа. Алгоритмы с полиномиальной временной сложностью называются алгоритмами с **полиномиальным временем**.

Алгоритмы, сложность которых равна $O(t^{f(n)})$, где t – константа, большая, чем 1, $f(n)$ – некоторая полиномиальная функция от n , называются **экспоненциальными**. Подмножество экспоненциальных алгоритмов, сложность которых равна $O(c^{f(n)})$, где c – константа, $f(n)$ возрастает быстрее, чем постоянная, но медлен-

нее, чем линейная функция, называется **суперполиномиальным**. Алгоритмы, сложность которых равна $O(n!)$, называются алгоритмами с факториальной сложностью. Таким образом, справедливо следующее соотношение:

$$O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n) < O(n!).$$

В настоящее время алгоритм считается практически полезным только в том случае, если его временная функция растет полиномиально относительно размеров входных данных, причем ограниченной степени. Практичными являются алгоритмы сложности $O(n)$, $O(n^2)$. Не практичны экспоненциальные и факториальные алгоритмы. Их сложность превосходит любую полиномиальную оценку.

9.2. Постановка задачи поиска

В практической деятельности приходится решать такие прикладные задачи, в которых необходимо локально или автономно произвести поиск нужных объектов. Рассмотрим задачу поиска элемента в массиве.

Пусть задан массив, состоящий из одного или нескольких элементов любого типа, отличного от файлового. Предположим, что в некотором массиве хранится множество из $n(n \times m)$ элементов и необходимо определить положение некоторого элемента в этом массиве. Каждый элемент массива имеет индекс (индексы), причем индексы различны и однозначно идентифицируют элементы массива.

Задача поиска в этом случае состоит в отыскании индекса (индексов) элемента по заданному свойству элемента.

Существуют два возможных результата поиска:

- поиск оказался удачным, т.е. позволил определить положение элемента в массиве;
- поиск оказался неудачным, т.е. достигнут конец массива, но элемент с заданным свойством отсутствует.

9.3. Последовательный (линейный) поиск

Простейшим методом поиска является последовательный просмотр таблицы, одномерного массива или строки до нахождения элемента с требуемыми свойствами или установления факта, что такого элемента нет. Для массива, данные в котором не упорядочены сортировкой, единственный путь поиска заданного элемента состоит в сравнении каждого элемента массива с заданным. При совпадении некоторого элемента массива с заданным, его позиция в массиве фиксируется. Этот алгоритм называется **последовательным**, или **линейным**, поиском. Ядром такого алгоритма является цикл по индексу массива. Так, если в массиве `inta[n]` нужно найти введенное с клавиатуры целое число `x`, о котором никакой дополнительной информации нет, то соответствующий цикл можно записать следующим образом:

```
for(i = 0; i < n; i++) if(a[i] == x) k = i;
```

Этот способ решения обладает двумя существенными недостатками:

- если значение `x` встречается в массиве несколько раз, то будет найдено последнее из них;

- после того, как нужное значение уже найдено, массив просматривается до конца, то есть всегда выполняется n сравнений.

Для устранения этих недостатков необходимо прервать просмотр массива сразу после обнаружения заданного числа. Так как в этом случае число повторений не известно, необходимо использовать цикл с предусловием:

```
while((i < n) && (a[i] != x))i++;
```

9.4. Бинарный поиск

Наиболее эффективным и распространенным методом непоследовательного поиска в одномерных *упорядоченных* массивах является **бинарный** поиск, который называют также методом *половинного деления*, методом *дихотомии*, методом *логарифмического* поиска или методом *деления пополам*. Основная идея бинарного поиска состоит в том, что сначала искомый элемент x сравнивается со средним элементом массива $a[n]$. Результат сравнения либо приводит к решению задачи, либо позволяет определить, в какой части массива – левой или правой – продолжать поиск. После каждой такой итерации область поиска сокращается вдвое и не более чем через $\lceil \log_2 n \rceil$ сравнений мы закончим поиск, то есть либо попадем на элемент x , либо покажем, что x не принадлежит массиву $a[n]$. Например, при $n = 1000$ бинарный поиск в 100 раз быстрее последовательного, а при $n = 1000000$ – в 50000 раз. Разумеется, если массив неупорядоченный, то применить к нему бинарный поиск нельзя. Таким образом, факт упорядоченности элементов в массиве играет ключевую роль при бинарном поиске.

9.5. Постановка задачи сортировки данных

От порядка расположения данных в памяти ЭВМ, во многом зависит скорость и простота выполнения алгоритмов, предназначенных для обработки этих данных. Поэтому в программировании часто возникает задача перегруппировки данных в невозрастающем или неубывающем, порядке. Такая задача называется упорядочением или сортировкой элементов данной структуры, в простейшем случае – элементов одномерного массива.

Задача сортировки связана со многими важными приложениями, кроме того, служат хорошей иллюстрацией анализа сложности алгоритмов и тем самым позволяют разумно выбирать лучшие среди, казалось бы, равноценных методов. Каждый из алгоритмов сортировки имеет свои достоинства и недостатки, и выбирать алгоритмы нужно, исходя из конкретной постановки задачи. Выбор алгоритма сортировки существенно зависит от структуры обрабатываемых данных, поэтому все методы сортировки подразделяют на два класса: внутренний (сортировка массивов) и внешний (сортировка последовательных файлов). Различие: при внутренней сортировке все данные хранятся в оперативной памяти ЭВМ, а при внешней – во внешней памяти. При внутренней сортировке имеются более гибкие возможности для построения алгоритмов, так как все данные выстроены в виде массива и как бы лежат перед пользователем, он видит каждый элемент и имеет к нему прямой доступ. При внешней сортировке доступ к данным ограничен, поскольку они из-за своего размера в оперативной памяти не помещаются.

Уточним математическую постановку задачи сортировки данных [7].

Пусть надо упорядочить элементы $m_1, m_2, \dots, m_n$, которые назовем записями. Каждой записи m_i поставим в соответствие свой ключ k_i , который и будет управлять процессом сортировки. Помимо ключа запись может содержать и дополнительную информацию, которая не влияет на сортировку, но всегда остается в этой записи. Задача заключается в нахождении такой перестановки $m_{p_1}, m_{p_2}, \dots, m_{p_n}$, записей $m_1, m_2, \dots, m_n$, после которой ключи будут располагаться в неубывающем (невозрастающем) порядке:

$$k_{p_1} \leq k_{p_2} \leq \dots \leq k_{p_n} \quad (k_{p_1} \geq k_{p_2} \geq \dots \geq k_{p_n})$$

Сортировка называется **устойчивой**, если она удовлетворяет дополнительному условию: записи с одинаковыми ключами остаются в прежнем порядке, т.е. $p_i < p_j$, если $k_{p_i} = k_{p_j}$ и $i < j$.

Так как каждый ключ идентифицирует соответствующую запись, то эти записи можно не определять при сортировке, рассматривая лишь их ключи (в простейшем случае, когда запись состоит лишь из сортируемых элементов, понятия записи и ключ совпадают).

При внутренней сортировке выбранный метод должен экономно использовать время работы процессора и память. Хорошие алгоритмы затрачивают на сортировку записей время порядка $n \log n$, а мерой эффективности может служить число необходимых сравнений ключей и число перестановок записей. Эти числа являются функциями от n – числа сортируемых записей.

При сортировке массивов будем предполагать, что перестановки, переводящие элементы массива в нужный порядок, должны выполняться *на том же месте*, т.е. без использования промежуточного массива. Методы, в которых элементы из массива A передаются в результирующий массив B , представляют значительно меньший интерес.

9.6. Прямые и быстрые методы внутренней сортировки

Все методы внутренней сортировки делятся на прямые и быстрые (или улучшенные) [7].

Для прямых методов сортировки требуется порядка n^2 сравнений ключей, они более просты и удобны для объяснения характерных черт основных принципов большинства сортировок. Программы этих методов легко понимать и они короче программ быстрых алгоритмов.

Улучшенные методы сортировки требуют небольшого числа операций, но эти операции обычно сами более сложны, и поэтому для достаточно малых n прямые методы оказываются быстрее, хотя при больших значениях n их использовать не следует.

Методы внутренней сортировки можно разбить в соответствии с определяющими их принципами на три класса:

1. Сортировка вставками: элементы просматриваются по одному, и каждый новый элемент вставляется на подходящую позицию среди ранее упорядоченных элементов.

2. Сортировка с помощью выбора: сначала выделяется наименьший (или наибольший) элемент и каким-либо образом отделяется от остальных, затем выбирается наименьший (наибольший) из оставшихся элементов и т.д.

3. Сортировка с помощью обмена: последовательно просматриваются пары соседних элементов; если элементы в паре образуют инверсию (т.е. неупорядочены), то они меняются местами.

Все прямые методы сортировки фактически передвигают каждый элемент массива на каждом элементарном шаге на одну позицию. Поэтому требуют $O(n^2)$ таких шагов. В основу любых улучшенных методов положен принцип перемещения элементов на каждом шаге сортировки на большие расстояния. Такой подход позволяет существенно уменьшить время сортировки, но усложнить алгоритмы выполнения.

Улучшенный метод обмена – Шейкерная сортировка, метод Хоара (самый быстрый из известных на сегодняшний день методов внутренней сортировки), усовершенствование сортировки прямого включения – метод Шелла, прямого выбора – сортировка с помощью бинарного дерева.

9.7. Сортировка вставками

Рассмотрим процесс сортировки методом вставки (прямого включения) массива $a[1], a[2], \dots, a[n]$. Вначале в исходном массиве выделяется уже отсортированная, например по неубыванию, часть $a[1] \leq a[2] \leq \dots \leq a[i-1]$ и не отсортированная $a[i], a[i+1], \dots, a[n]$. Затем с шагом 1, начиная с $i = 2$ чередуя сравнения и перемещения элемента $a[i]$, в отсортированной части массива определяется такой индекс (ключ) j ($1 \leq j \leq i-1$), чтобы выполнялись неравенства $a[j-1] \leq a[i] \leq a[j]$. Если такой индекс найден, то элемент $a[i]$ вставляется на соответствующее место. Для этого выполняется сдвиг на одну позицию вправо элементов отсортированной части: $a[j], \dots, a[i-1]$. После этого осуществляется переход к следующему значению i . В противном случае просто происходит переход к следующему значению i . С каждым шагом отсортированная часть массива увеличивается на один элемент. Очевидно, что для выполнения полной сортировки потребуется $n-1$ шаг, где n - число элементов исходного массива.

Покажем процесс сортировки вставками на примере целочисленного массива $a[1..12]$, состоящего из чисел 22, -14, 13, 28, 6, 41, -5, 18, 0, 35, 6, -15 (значком * обозначены вставленные элементы, выделены элементы отсортированной части)

Исходный массив:	22	-14	13	28	6	41	-5	18	0	35	6	-15
1-й шаг	-14*	22	13	28	6	41	-5	18	0	35	6	-15
2-й шаг:	-14	13*	22	28	6	41	-5	18	0	35	6	-15
3-й шаг:	-14	13	22	28*	6	41	-5	18	0	35	6	-15
4-й шаг:	-14	6*	13	22	28	41	-5	18	0	35	6	-15
5-й шаг:	-14	6	13	22	28	41*	-5	18	0	35	6	-15
6-й шаг:	-14	-5*	6	13	22	28	41	18	0	35	6	-15
7-й шаг:	-14	-5	6	13	18*	22	28	41	0	35	6	-15
8-й шаг:	-14	-5	0*	6	13	18	22	28	41	35	6	-15
9-й шаг:	-14	-5	0	6	13	18	22	28	35*	41	6	-15
10-й шаг:	-14	-5	0	6	6*	13	18	22	28	35	41	-15

11-й шаг: -15* -14 -5 0 6 6 13 18 22 28 35 41

Как видно из примера, в процессе сортировки на первом шаге первый элемент исходного массива считается отсортированным, а следующий за ним элемент сравнивается с ним. Если этот элемент больше, он остается на своем месте, в противном случае первый элемент сдвигается на вторую позицию, а второй помещается на первую (как в нашем примере). Далее все выбираемые элементы из неотсортированной части массива последовательно сравниваются с элементами отсортированной части, начиная с первого, до тех пор, пока не встретится больший или равный ему. Этот элемент и все последующие элементы отсортированной части перемещаются на одну позицию вправо (или вниз, если массив расположен в столбик), освобождая место для нового элемента. Этот процесс закончится при выполнении одного из двух условий:

- найден первый слева элемент $a[j] \geq a[i]$, что говорит о необходимости вставки $a[i]$ между $a[j]$ и $a[j-1]$, в частности, если $j=1$, то $a[i]$ помещается в первую позицию, что влечет за собой необходимость увеличения диапазона индекса массива на единицу;
- достигнут правый конец отсортированной части массива (правый барьер), следовательно, элемент $a[i]$ нужно оставить на прежнем месте.

Заметим, что сравнение и перемещение элемента $a[i]$ в отсортированной части массива можно проводить и справа налево, начиная с $(i-1)$ -го элемента.

9.8. Сортировка с помощью прямого выбора

Сортировка массива размером n в неубывающем (невозрастающем) порядке методом прямого выбора выполняется по следующему алгоритму.

1. Выбирается элемент данного массива с минимальным(максимальным) значением (или ключом).
2. Выбранный элемент и первый элемент меняются местами.
3. Этот процесс повторяется с оставшимися $n - 1$ элементами, затем с $n - 2$ элементами и т.д. до тех пор, пока не останется один, самый большой/меньший элемент.

Всего потребуется $n - 1$ раз выполнить указанную последовательность действий. В процессе сортировки будет увеличиваться отсортированная часть массива, а неотсортированная соответственно уменьшаться.

В отличие от сортировки вставками, когда на каждом шаге рассматривается только один очередной элемент исходного массива и **все** элементы отсортированной части, среди которых ищется место для вставки, при прямом выборе для поиска одного элемента с наименьшим (наибольшим) ключом просматриваются все элементы исходного массива и найденный элемент помещается как очередной в отсортированную часть массива.

Напомним, что во всех рассматриваемых методах сортировки мы не используем промежуточных массивов.

Схему сортировки методом прямого выбора проиллюстрируем на примере массива, рассмотренного в предыдущем пункте. Элементы, которые меняются местами на очередном шаге, выделены полужирным шрифтом, а минимальные элементы неотсортированной части, кроме того, отмечены звездочкой. Отсортированная часть подчеркнута.

Исходный массив:	22	-14	13	28	6	41	-5	18	0	35	6	-15*
1-й шаг	-15	-14*	13	28	6	41	-5	18	0	35	6	22
2-й шаг:	-15	-14	13	28	6	41	-5*	18	0	35	6	22
3-й шаг:	-15	-14	-5	28	6	41	13	18	0*	35	6	22
4-й шаг:	-15	-14	-5	0	6*	41	13	18	28	35	6	22
5-й шаг:	-15	-14	-5	0	6	41	13	18	28	35	6*	22
6-й шаг:	-15	-14	-5	0	6	6	13*	18	28	35	41	22
7-й шаг:	-15	-14	-5	0	6	6	13	18*	28	35	41	22
8-й шаг:	-15	-14	-5	0	6	6	13	18	28	35	41	22*
9-й шаг:	-15	-14	-5	0	6	6	13	18	22	35	41	28*
10-й шаг:	-15	-14	-5	0	6	6	13	18	22	28	41	35
11-й шаг:	-15	-14	-5	0	6	6	13	18	22	28	35	41

9.9. Сортировка с помощью прямого обмена

Алгоритм данного метода сортировки заключается в последовательных просмотрах массива от начала к концу и обмену местами *соседних* элементов, если они образуют *инверсию*. Опишем алгоритм сортировки по **неубыванию** подробнее.

Пусть задан массив a размером n . Начинаем просмотр массива с первой пары элементов $a[1]$ и $a[2]$. Если первый элемент этой пары больше второго, то меняем их местами, иначе оставляем их без изменения и сравниваем вторую пару элементов $a[2]$ и $a[3]$. Если $a[2] > a[3]$, то меняем их местами и т.д. На первом шаге будут просмотрены все пары элементов массива: $a[i]$ и $a[i+1]$ для i от 1 до $n-1$. В результате такого просмотра и необходимых обменов **максимальный** элемент переместится в конец массива и будет являться отсортированной частью. На втором шаге аналогичная процедура проводится с первого до $(n-1)$ -го элемента. Тем самым второй по величине элемент массива переместится на предпоследнее место. Отсортированная часть будет содержать два элемента. Эти действия продолжают-ся до тех пор, пока количество элементов в неотсортированной части массива не уменьшится до двух. На последнем шаге выполняется упорядочение оставшихся двух элементов. После $(n-1)$ шагов массив окажется отсортированным по **неубыванию**. Для иллюстрации отсортируем массив по неубыванию методом простого обмена. Отсортированная часть выделена полужирным шрифтом.

Исходный массив:	22	-14	13	28	6	41	-5	18	0	35	6	-15
1-й шаг	-14	13	22	6	28	-5	18	0	35	6	-15	41
2-й шаг:	-14	13	6	22	-5	18	0	28	6	-15	35	41
3-й шаг:	-14	6	13	-5	18	0	22	6	-15	28	35	41
4-й шаг:	-14	6	-5	13	0	18	6	-15	22	28	35	41
5-й шаг:	-14	-5	6	0	13	6	-15	18	22	28	35	41
6-й шаг:	-14	-5	0	6	6	-15	13	18	22	28	35	41
7-й шаг:	-14	-5	0	6	-15	6	13	18	22	28	35	41
8-й шаг:	-14	-5	0	-15	6	6	13	18	22	28	35	41
9-й шаг:	-14	-5	-15	0	6	6	13	18	22	28	35	41
10-й шаг:	-14	-15	-5	0	6	6	13	18	22	28	35	41
11-й шаг:	-15	-14	-5	0	6	6	13	18	22	28	35	41

Сортировку методом прямого обмена называют еще методом "пузырька". Это название происходит от образной интерпретации, при которой элементы данного массива расположены вертикально и в **процессе** выполнения сортировки на каждом шаге более легкие элементы, как пузырьки в ванне с водой, поднимаются до уровня, соответствующего их весу.

Практическая часть

ВЫПОЛНЕНИЕ РАБОТЫ: в соответствии с заданиями пунктов 1-4, учитывая вариант (п. 3), составить и реализовать программу.

Задание

1. Сформировать при помощи генератора псевдослучайных чисел линейный целочисленный массив $A[20]$, таким образом, что бы элементы массива принадлежали отрезку $[-50, 50]$.
2. Выполнить в нём линейный поиск заданного элемента.
3. Отсортировать массив определённым методом в соответствии с вариантом (см. таблицу).

Метод сортировки	Вариант
Вставкой	1, 4, 7, 10, 13, 16, 19, 22
Обменом	2, 5, 8, 11, 14, 17, 20, 23
Выбором	3, 6, 9, 12, 15, 18, 21, 25

4. Выполнить бинарный поиск элемента.

Формирование массива, вывод его на экран, линейный поиск, сортировку и бинарный поиск оформить в виде соответствующих функций.

Контрольные вопросы

1. Понятие сложности алгоритмов.
2. На какие классы делятся алгоритмы в соответствии с их временной или пространственной сложностью?
3. Постановка задачи сортировки данных.
4. Прямые и быстрые методы внутренней сортировки.
5. Алгоритм сортировки массива методом вставки.
6. Алгоритм сортировки массива методом прямого выбора.
7. Алгоритм сортировки массива методом прямого обмена.
8. Понятие инверсии.
9. Постановка задачи поиска элемента в массиве.
10. Алгоритм последовательного (линейного) поиска.
11. Алгоритм бинарного поиска.

СПИСОК ЛИТЕРАТУРЫ

Перечень основной литературы:

1. Блох Дж. Java. Эффективное программирование / Дж. Блох; перевод В. Стрельцов; под редакцией Р. Усманова. - 3-е изд. - Саратов: Профобразование, 2024. - 310 с. - ISBN 978-5-4488-0127-3. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/145890.html>
2. Вязовик Н. А. Программирование на Java: учебное пособие / Н. А. Вязовик. - 4-е изд. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. - 601 с. - ISBN 978-5-4497-0852-6. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/146383.html>

Перечень дополнительной литературы:

1. Ховард М. Как написать безопасный код на C++, Java, Perl, PHP, ASP.NET / М. Ховард, Д. Лебланк, Д. Виега. - Москва: ДМК Пресс, 2018. - 288 с. - ISBN 978-5-97060-617-9. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/124992.html>
2. Мухаметзянов Р.Р. Основы программирования на Java: учебное пособие / Мухаметзянов Р.Р. - Набережные Челны: Набережночелнинский государственный педагогический университет, 2017. - 114 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/66812.html>
3. Васюткина И. А. Технология разработки объектно-ориентированных программ на JAVA: учебно-методическое пособие / И. А. Васюткина. – Новосибирск: Новосибирский государственный технический университет, 2012. – 152 с. – ISBN 978-5-7782-1973-1. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/45047.html>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- Интернет-портал образовательных ресурсов по ИТ - <http://www.intuit.ru>
- Компьютерная энциклопедия - <http://www.computer-encyclopedia.ru>
- Официальный сайт Java - <http://java.sun.com/>
- Сайт по программированию и алгоритмике - <http://algolist.manual.ru/>
- Форум по ИТ - <http://www.citforum.ru>

Приложение 1

Порядок выполнения лабораторных работ

1. Ознакомиться с теоретическим материалом, соответствующим теме лабораторной работы и подготовить ответы на предложенные контрольные вопросы.
2. Из списка задач выбрать соответствующие своему варианту. Для каждой задачи разработать математическую модель, алгоритм и программу решения.
3. Реализовать программу на ЭВМ.
4. Для защиты лабораторной работы знать ответы на контрольные вопросы и предоставить письменный отчёт, содержащий:
 - а) название лабораторной работы;
 - б) номер варианта;
 - в) решение каждой задачи, включающее в себя:
 - условие задачи;
 - математическую модель;
 - алгоритм;
 - листинг программы с отметкой преподавателя о её выполнении;
 - исходный набор данных и соответствующий ему результат;
 - анализ результатов работы программы.

Приложение 2

Базовые функции

В классе Math, на что как бы намекает это название, собраны самые часто употребляемые программистами математические функции. Вот самые интересные из них:

Метод	Описание
double sin(double d)	Возвращает синус угла d, заданного в радианах
double cos(double d)	Возвращает косинус угла d, заданного в радианах
double tan(double d)	Возвращает тангенс угла d, заданного в радианах
double asin(double d)	Возвращает арксинус
double acos(double d)	Возвращает арккосинус
double atan(double d)	Возвращает арктангенс
double sinh(double d)	Возвращает гиперболический синус
double cosh(double d)	Возвращает гиперболический косинус
double tanh(double d)	Возвращает гиперболический тангенс

Функции Math.sin(), Math.cos() и Math.tan() принимают угол, заданный в радианах. Чтобы преобразовать углы из градусов в радианы и обратно, в классе Math есть две специальные функции:

Метод	Описание
double toRadians(double angdeg)	Преобразует угол из градусов в радианы
double toDegrees(double angrad)	Преобразует угол из радиан в градусы

Константа	Описание
<code>doubleMath.PI</code>	Число «Пи» равное 3.141592653589793
<code>doubleMath.E</code>	Число «Е» равное 2.718281828459045

Приложение 3

Правила поведения в компьютерном классе

1. К работе в компьютерном классе допускаются лица, ознакомленные с инструкцией по технике безопасности и охране труда, с правилами поведения и размещения информационных ресурсов.
2. Работа студентов в компьютерном классе разрешается только в присутствии преподавателя (инженера, лаборанта).
3. Во время групповых занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
4. **Перед началом работы необходимо:**
 - убедиться в отсутствии видимых повреждений на рабочем месте;
 - разместить на столе тетради, учебные пособия так, чтобы они не мешали работе на компьютере;
 - принять правильную рабочую позу;
 - если сеанс работы предыдущего пользователя не был завершен, завершить его;
 - ввести регистрационную информацию (при необходимости).
5. **При работе в компьютерном классе категорически запрещается:**
 - находиться в классе в верхней одежде;
 - размещать одежду и сумки на рабочих местах;
 - находиться в классе с едой и напитками;
 - класть книги, тетради и т.п. на клавиатуру;
 - располагаться сбоку или сзади от включенного монитора;
 - присоединять или отсоединять кабели, трогать разъемы, провода и розетки;
 - передвигать компьютеры;
 - открывать системный блок;
 - пытаться самостоятельно устранять неисправности в работе аппаратуры;
 - перекрывать вентиляционные отверстия на системном блоке и мониторе;
 - ударять по клавиатуре, нажимать бесцельно на клавиши;
 - удалять или перемещать чужие файлы;
 - устанавливать и запускать компьютерные игры;
 - использовать Интернет-ресурсы неучебного назначения.
6. **Находясь в компьютерном классе, необходимо:**
 - соблюдать тишину и порядок, выключать мобильные телефоны от громкой связи;
 - выполнять все требования преподавателя, инженера и лаборанта;
 - работать только под своим именем и паролем;
 - соблюдать режим работы (продолжительность непрерывной работы за компьютером не более двух часов с обязательным 10-минутным перерывом и гимнастикой для глаз; продолжительность интенсивной работы с клавиатурой не более 30 минут с последующей гимнастикой для рук; общая продолжительность работы не более 4 часов в день);
 - при появлении рези в глазах, резком ухудшении видимости, невозможности сфокусировать взгляд или навести его на резкость, появлении боли

в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем преподавателю и обратиться к врачу;

- после окончания работы завершить все активные программы и корректно завершить сеанс;
- оставить рабочее место чистым.

7. Работая за компьютером, необходимо соблюдать правильную позу:

- расстояние от экрана до глаз 70-80 см (расстояние вытянутой руки);
- вертикально прямая спина;
- плечи опущены и расслаблены;
- ноги на полу и не скрещены;
- локти, запястья и кисти рук на одном уровне;
- локтевые, тазобедренные, коленные, голеностопные суставы под прямым углом.

8. При появлении программных ошибок или сбоях оборудования студент обязан немедленно обратиться к преподавателю (инженеру, лаборанту).

9. В случае порчи или выхода из строя оборудования компьютерного класса по вине пользователя ремонт или замена оборудования производится за счет пользователя.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по организации и проведению самостоятельной работы
по дисциплине
«Моделирование на основе проблемно-ориентированных языков»
для студентов направления подготовки
02.03.01 Математика и компьютерные науки
направленность (профиль)
«Анализ данных и искусственный интеллект»

Общие положения

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
- подготовка к семинарам, практическим и лабораторным работам, их оформление;
- составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине **«Моделирование на основе проблемно-ориентированных языков»** направлена на формирование следующих **компетенций**:

ПК-4

Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования

ПК-5

Способен принимать участие в управлении проектами создания информационных систем и программных комплексов на стадиях их жизненного цикла.

1. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование соответствующего набора компетенций будущего бакалавра по направлению подготовки «Математика и компьютерные науки».

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

2. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателем	Всего
6 семестр					
ПК-4, ПК-5	Самостоятельное изучение литературы	Собеседование	18	2	20
ПК-4, ПК-5	Решение практических задач	Конспект	18	2	20
Итого за 6 семестр			36	4	40
Итого			36	4	40

3. Порядок выполнения самостоятельной работы студентом

3.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста:**

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;
2. Выделите главное, составьте план;
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

3.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

3.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

3.4. Методические рекомендации по написанию научных текстов (докладов, рефератов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т. п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Реферат (доклад) - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Реферат не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в реферате должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки реферата студентом.

Выполнение реферата начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;

- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания реферата. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста реферата предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки реферат сдается на кафедру для его оценивания руководителем.

Требования к написанию реферата

Написание 1 реферата является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема реферата может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Реферат должен быть написан научным языком.

Объем реферата должен составлять 20-25 стр.

Структура реферата:

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.

- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.

- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса

- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

- Приложение (при необходимости).

Требования к оформлению:

- текст с одной стороны листа;
- шрифт Times New Roman;
- кегль шрифта 14;

- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- реферат должен быть представлен в сброшюрованном виде.

Порядок защиты реферата:

Защита реферата проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту реферата отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите реферата приветствуется использование мультимедиа-презентации.

Оценка реферата

Реферат оценивается по следующим критериям:

- соблюдение требований к его оформлению;
- необходимость и достаточность для раскрытия темы приведенной в тексте реферата информации;
- умение студента свободно излагать основные идеи, отраженные в реферате;
- способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

**Темы рефератов по дисциплине
«Моделирование на основе проблемно-ориентированных языков»**

Общие темы, охватывающие концепции и применение DSL:

- **Эволюция языков программирования и место DSL в ней:** Обзор истории развития языков, переход от общих языков к специализированным, причины и преимущества использования DSL.
- **Классификация DSL:** Изучение различных типов DSL (внутренние/внешние, декларативные/императивные, графические/текстовые и т.д.) и критериев их классификации.
- **Преимущества и недостатки использования DSL:** Детальный анализ плюсов (повышение производительности, снижение сложности, улучшение коммуникации) и минусов (затраты на разработку, необходимость обучения, ограниченность применения) DSL.
- **Методологии разработки DSL:** Изучение и сравнение различных подходов к созданию DSL, включая Domain-Driven Design, Language-Oriented Programming, метапрограммирование.
- **Инструменты для разработки DSL:** Обзор инструментов и фреймворков, облегчающих создание и поддержку DSL (например, ANTLR, Xtext, MPS, Spoofox).
- **Роль DSL в модельно-ориентированной разработке (MDD/MDA):** Рассмотрение DSL как ключевого элемента в MDD/MDA, позволяющего описывать модели предметной области и автоматически генерировать код.
- **Применение DSL в Agile-разработке:** Исследование возможностей использования DSL для ускорения разработки, улучшения коммуникации и повышения гибкости в Agile-проектах.
- **Будущее DSL:** Обсуждение тенденций развития DSL, их роли в новых парадигмах программирования (например, Low-Code/No-Code платформы) и областях применения (например, IoT, Big Data).

Темы, фокусирующиеся на конкретных проблемных областях и DSL:

- **DSL для моделирования бизнес-процессов (BPMN):** Анализ BPMN как DSL для описания и автоматизации бизнес-процессов, инструменты для его использования, преимущества и недостатки.
- **DSL для конфигурирования программного обеспечения:** Изучение языков для описания конфигураций, их роль в управлении сложностью, примеры (например, YAML, JSON, HOCON).
- **DSL для моделирования пользовательских интерфейсов:** Рассмотрение языков для декларативного описания UI, их влияние на скорость разработки и удобство поддержки, примеры (например, QML, XAML).
- **DSL для разработки игр:** Анализ языков, упрощающих создание игрового контента, логики и взаимодействия, примеры (например, Lua в Unity, Unreal Engine Blueprints).
- **DSL для моделирования финансовых инструментов:** Изучение языков, позволяющих описывать сложные финансовые продукты и проводить анализ рисков.
- **DSL для генетического программирования:** Исследование использования DSL для представления программ, которые эволюционируют в процессе генетического алгоритма.
- **DSL для машинного обучения:** Рассмотрение языков и фреймворков, упрощающих создание и развертывание моделей машинного обучения.
- **DSL для автоматизации DevOps:** Изучение языков, позволяющих описывать инфраструктуру и процессы развертывания, примеры (например, Terraform, Ansible).

Темы, связанные с разработкой и реализацией DSL:

- **Синтаксический анализ и генерация кода для DSL:** Изучение техник и инструментов для разбора DSL и преобразования его в исполняемый код или другие представления.

- **Семантический анализ DSL:** Рассмотрение методов проверки корректности DSL-программ и выявления ошибок на этапе компиляции или интерпретации.
- **Разработка редакторов и IDE для DSL:** Изучение подходов к созданию специализированных инструментов для работы с DSL, обеспечивающих подсветку синтаксиса, автодополнение, отладку и другие полезные функции.
- **Интеграция DSL с существующими языками программирования:** Рассмотрение методов использования DSL внутри программ, написанных на общих языках, и обеспечения взаимодействия между ними.
- **Оптимизация DSL-программ:** Изучение техник повышения производительности DSL-программ, включая статический анализ, оптимизацию кода и распараллеливание вычислений.

Учебно-методическое обеспечение дисциплины

Перечень основной литературы:

1. Блох Дж. Java. Эффективное программирование / Дж. Блох; перевод В. Стрельцов; под редакцией Р. Усманова. - 3-е изд. - Саратов: Профобразование, 2024. - 310 с. - ISBN 978-5-4488-0127-3. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/145890.html>
2. Вязовик Н. А. Программирование на Java: учебное пособие / Н. А. Вязовик. - 4-е изд. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. - 601 с. - ISBN 978-5-4497-0852-6. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/146383.html>

Перечень дополнительной литературы:

1. Ховард М. Как написать безопасный код на C++, Java, Perl, PHP, ASP.NET / М. Ховард, Д. Лебланк, Д. Виега. - Москва: ДМК Пресс, 2018. - 288 с. - ISBN 978-5-97060-617-9. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/124992.html>
2. Мухаметзянов Р.Р. Основы программирования на Java: учебное пособие / Мухаметзянов Р.Р. - Набережные Челны: Набережночелнинский государственный педагогический университет, 2017. - 114 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/66812.html>
3. Васюткина И. А. Технология разработки объектно-ориентированных программ на JAVA: учебно-методическое пособие / И. А. Васюткина. – Новосибирск: Новосибирский государственный технический университет, 2012. – 152 с. – ISBN 978-5-7782-1973-1. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/45047.html>

Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

Непретимова Е.В. Моделирование на основе проблемно-ориентированных языков: Учебное пособие. Электронный ресурс: Учебное пособие / Непретимова Е.В. - Ставрополь: СКФУ, 2025. - 108 с. - экземпляров неограниченно

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- Интернет-портал образовательных ресурсов по ИТ - <http://www.intuit.ru>
 - Компьютерная энциклопедия - <http://www.computer-encyclopedia.ru>
 - Официальный сайт Java - <http://java.sun.com/>
 - Сайт по программированию и алгоритмике - <http://algotlist.manual.ru/>
- Форум по ИТ - <http://www.citforum.ru>