

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Первякова Светлана Валерьевна
Должность: Директор колледжа СКФУ в г. Ставрополе
Дата подписания: 18.06.2026 12:25:50
Уникальный программный ключ:
63a18c373da05c089b08fcc89c468af301900f17

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«Северо-Кавказский федеральный университет»

Колледж СКФУ в г. Ставрополе

УТВЕРЖДАЮ

и.о. директора института перспективной
инженерии, канд. тех. наук,
доцент Порохня А.А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ОП.06 Основы алгоритмизации и программирования

Специальность 09.02.11 Разработка и управление программным обеспечением

Форма обучения очная

Ставрополь

1. Паспорт фонда оценочных средств

1.1. Область применения

Фонд оценочных средств (далее – ФОС) предназначен для оценивания знаний, умений, уровня сформированности компетенций студентов, обучающихся по специальности 09.02.11 Разработка и управление программным обеспечением по дисциплине ОП.06 Основы алгоритмизации и программирования.

ФОС составлен на основе ФГОС и рабочей программы.

Промежуточная аттестация по учебной дисциплине предусмотрена в форме *зачета с оценкой* с выставлением отметки по системе «отлично, хорошо, удовлетворительно, неудовлетворительно».

1.2. Планируемые результаты освоения (учебной) дисциплины

ФОС позволяет оценить знания, умения, сформированность общих и профессиональных компетенций в соответствии с требованиями ФГОС и рабочей программой (учебной) дисциплины.

Планируемые результаты освоения (знания и умения) и перечень осваиваемых компетенций (общих и профессиональных) указываются в соответствии с ФГОС, ОП и рабочей программой учебной дисциплины:

умения:

- Разрабатывать алгоритмы для конкретных задач.
- Использовать программы для графического отображения алгоритмов.
- Определять сложность работы алгоритмов.
- Работать в среде программирования.
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- Оформлять код программы в соответствии со стандартом кодирования.
- Выполнять проверку, отладку кода программы.

знания:

- Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.
- Эволюцию языков программирования, их классификацию, понятие системы программирования.
- Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
- Подпрограммы, составление библиотек подпрограмм
- Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

общие компетенции:

ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.

ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях.

ОК 04 Эффективно взаимодействовать и работать в коллективе и команде.

профессиональные компетенции:

ПК 1.5 Защищать информацию в базе данных с использованием технологии.
 ПК 2.4 Выполнять тестирование и отладку программного обеспечения.
 ПК 3.8 Производить оценку информационной системы для выявления возможности ее модернизации.

1.3. Формы контроля и оценивания

Предметом оценки служат умения и знания, предусмотренные ФГОС по (учебной) дисциплине, направленные на формирование общих и профессиональных компетенций.

Таблица 1 Контроль и оценка освоения учебной дисциплины по темам (разделам)

Элемент учебной дисциплины	Виды и методы оценивания			
	Текущий контроль		Промежуточная аттестация	
	Методы оценки	Проверяемые ОК, ПК	Методы оценки	Проверяемые ОК, ПК
Раздел 1. Введение в программирование				
Тема 1.1. Языки программирования	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8	Зачет с оценкой Устный опрос Тестирование	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8
Тема 1.2 Типы данных	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Раздел 2. Основные принципы алгоритмизации и программирования				
Тема 2.1. Операторы языка программирования	Устный опрос Тестирование	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Раздел 3. Процедуры и функции				
Тема 3.1. Подпрограммы	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Тема 3.2. Структуризация в программировании	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Тема 3.3. Модульное программирование	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Раздел 4. Основные конструкции языков программирования				
Тема 4.1 Указатели	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Раздел 5. Объектно-ориентированное программирование				
Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		

Тема 5.2. Интегрированная среда разработчика.	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Тема 5.3. Визуальное событийно- управляемое программирование.	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Тема 5.4 Разработка оконного приложения.	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Тема 5.5 Этапы разработки приложений.	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		
Тема 5.6 Иерархия классов.	Устный опрос	ОК 01 ОК 02 ОК 03 ОК 04 ПК 1.5 ПК 2.4 ПК 3.8		

2. Оценочные средства для проведения текущего контроля успеваемости

2.1. Вопросы для устного опроса

Раздел 1. Введение в программирование

Тема 1.1. Языки программирования

1. Развитие языков программирования.
2. Обзор языков программирования. Области применения языков программирования.
3. Стандарты языков программирования. Среда проектирования
4. Компиляторы и интерпретаторы. Жизненный цикл программы.
5. Основные этапы решения задач на компьютере.
6. Составление блок-схем алгоритмов.
7. Основные элементы блок-схемы.
8. Языки и системы программирования. Языки высокого и низкого уровня.
9. Правила записи выражений и операций. Синтаксис.

Тема 1.2 Типы данных

1. Основные понятия алгоритмизации. Понятия «алгоритм», «исполнитель алгоритма».
2. Свойства и формы записи алгоритмов.
3. Основные алгоритмические конструкции: линейные, разветвляющиеся и циклические.
4. Логические основы алгоритмизации. Основы алгебры логики.
5. Логические операции: конъюнкция, дизъюнкция, инверсия, импликация, эквиваленция.
6. Составление таблиц истинности для сложных логических функций.
7. Составление словесных алгоритмов.
8. Что такое алгоритм?
9. Свойства алгоритма.
10. Способы записи алгоритма.
11. Виды алгоритмов.
12. Отличительные особенности алгоритмов с предусловием и постусловием.
13. Типы данных. Простые типы данных. Производные типы данных.
14. Структурированные типы данных

15. Понятие модуля и формы.
16. Пользовательские типы данных.
17. Использование форм, событий и методов.
18. Использование управляющих элементов.
19. Составление программ циклической структуры.
20. Работа с логическими операторами и операторами сравнения.
21. Написание кода с использованием операторов и циклов.

Раздел 2. Основные принципы алгоритмизации и программирования

Тема 2.1. Операторы языка программирования

1. Назначение языка программирования Pascal.
2. Что такое среда программирования?
3. Состав среды программирования.
4. Алфавит языка программирования Pascal.
5. Что такое идентификатор? Ограничения при создании идентификатора.
6. Из каких разделов может состоять программа на языке Pascal?
7. Какие из разделов программы на языке Pascal необходимы?
8. Для чего служит раздел описания переменных, раздел операторов?
9. Как записываются операторы вывода на экран в Pascal?
10. Что такое константа. Примеры констант в Pascal
11. Поясните: имя и значение переменной.
12. Операции языка Pascal. Приоритет операций.

Раздел 3. Процедуры и функции

Тема 3.1. Подпрограммы

1. Область видимости и время жизни переменной.
2. Механизм передачи параметров.
3. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов.
4. Что такое процедура? Ее общий вид.
5. Что называется параметром, и каково его назначение?
6. Правила записи параметров процедуры.
7. Формальные и фактические параметры. Каково отличие параметров-значений от параметров-переменных? Что может выступать в качестве параметров-значений и параметров-переменных?
8. Как вызвать процедуру в программе?
9. Опишите последовательность событий при вызове процедуры с параметрами-значениями.
10. Опишите последовательность событий при вызове процедуры с параметрами-переменными.
11. В чем различие между стандартными и определенными пользователем подпрограммами? Приведите примеры.

Тема 3.2. Структуризация в программировании

1. Основы структурного программирования.
2. Методы структурного программирования.
3. Чем отличаются локальные переменные от глобальных переменных?
4. Напишите, что выведет на экран следующая программа. Где в программе используется параметр-переменная, а где параметр-значение?

5. Работа с файлом последовательного доступа.
6. Использование стандартных процедур и функций для работы с файла
7. Чтение и запись файла.
8. Считывание данных из текстового файла.
9. Запись данных в текстовый файл.
10. Открытие и создание файла для чтения и записи.

Тема 3.3. Модульное программирование

1. Модульное программирование.
2. Понятие модуля. Структура модуля.
3. Компиляция и компоновка программы. Стандартные модули.
4. Создание программы с использованием полос прокрутки, таймера и заданием даты.
5. Создание вкладок, индикатора прогресса, ползунка.
6. Программирование модуля.
7. Создание библиотеки подпрограмм.

Раздел 4. Основные конструкции языков программирования

Тема 4.1. Указатели

1. Указатели. Описание указателей.
2. Основные понятия и применение динамически распределяемой памяти.
3. Создание и удаление динамических переменных.
4. Структуры данных на основе указателей. Задача о стеке.
5. Что называется подпрограммой?
6. Какие виды подпрограмм используются в языке программирования Паскаль ABC?
7. Чем отличается процедура от функции?
8. Как описывается процедура?
9. Как описывается функция?
10. Опишите оператор вызова процедуры?
11. Опишите оператор вызова функции?
12. Общие сведения о подпрограммах.
13. Определение и вызов подпрограмм.

Раздел 5. Объектно-ориентированное программирование

Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)

1. История развития ООП.
2. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
3. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
4. Что такое ООП?
5. Что такое объект?
6. Что такое абстракция?
7. Как задать свойства объекта? Пример.
8. Как выделить группу компонентов? Зачем это нужно?
9. Какие операции можно производить с группой компонентов?
10. Что такое проект? Как создать проект? Как сохранить проект?
11. Что такое приложение?
12. Что такое форма? Как задать свойства формы?
13. Что такое событие? Обработчик события?
14. Как создать обработчик события?
15. Компонентная модель COM

Тема 5.2. Интегрированная среда разработчика.

1. Перечислите основные свойства форм
2. Как выполняется программная настройка свойств?
3. Как задается фон и цвета формы?
4. Опишите алгоритм добавления форм и взаимодействия между формами
5. Как выполняется создание прямоугольных форм, закрытие формы?
6. Требования к аппаратным и программным средствам интегрированной среды разработчика.
7. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты.
8. Форма и размещение на ней управляющих элементов.
9. Панель компонентов и их свойства.
10. Окно кода проекта.
11. Состав и характеристика проекта.
12. Выполнение проекта.
13. Настройка среды и параметров проекта.

Тема 5.3. Визуальное событийно-управляемое программирование.

1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
2. Дополнительные элементы управления.
3. Свойства компонентов.
4. Виды свойств.
5. Синтаксис определения свойств.
6. Назначения свойств и их влияние на результат.
7. Управление объектом через свойства.
8. Создание проекта с использованием кнопочных компонентов и компонентов для работы с текстом.

Тема 5.4 Разработка оконного приложения

1. Разработка функционального интерфейса приложения.
2. Создание интерфейса приложения.
3. Разработка функциональной схемы работы приложения.
4. Разработка игрового приложения.
5. Разработка оконного приложения.
6. Разработка оконного приложения с несколькими формами.

Тема 5.5 Этапы разработки приложений

1. Разработка приложения. Основные этапы
2. Проектирование объектно-ориентированного приложения.
3. Создание интерфейса пользователя.
4. Тестирование, отладка приложения.
5. Разработка функционального интерфейса приложения.
6. Создание интерфейса приложения.
7. Разработка игрового приложения.

Тема 5.6 Иерархия классов

1. Объявление класса, создание экземпляров класса.
2. Создание наследованного класса.
3. Перегрузка операций.
4. Объявление класса, создание экземпляров класса.

Критерии оценивания:

Оценка «отлично» выставляется студенту за глубокое и полное овладение содержанием учебного материала, в котором студент легко ориентируется; владение понятийным аппаратом, умение использовать биографические материалы, давать доказательную и убедительную оценку освещаемому вопросу, активно и целесообразно использовать различные виды справочной литературы, высказывать в письменной форме и обосновывать свои суждения. Отличная отметка предполагает грамотное, логичное изложение материала, качественное внешнее оформление.

Оценка «хорошо» выставляется студенту, если студент полно освоил учебный материал, владеет понятийным аппаратом, ориентируется в изученном материале, осознанно применяет знания при освещении излагаемого материала, грамотно излагает материал, но содержание и форма ответа имеют некоторые неточности.

Оценка «удовлетворительно» выставляется студенту, если студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности в определение понятий, в применении знаний, не умеет доказательно обосновать свои суждения.

Оценка «неудовлетворительно» выставляется студенту, если студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, искажает их смысл, беспорядочно излагает материал.

3. Оценочные средства для промежуточной аттестации

3.1. Вопросы к зачету с оценкой

1. Развитие языков программирования.
2. Обзор языков программирования.
3. Области применения языков программирования.
4. Стандарты языков программирования.
5. Среда проектирования.
6. Компиляторы и интерпретаторы.
7. Жизненный цикл программы.
8. Основные этапы решения задач на компьютере.
9. Типы данных. Простые типы данных. Производные типы данных.
Структурированные типы данных.
10. Составление словесных алгоритмов.
11. Операции и выражения. Правила формирования и вычисления выражений.
12. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.
13. Условный оператор.
14. Оператор выбора.
15. Цикл с постусловием.
16. Цикл с предусловием.
17. Цикл с параметром. Вложенные циклы.
18. Массивы. Двумерные массивы.
19. Строки. Стандартные процедуры и функции для работы со строками.
20. Структурированный тип данных – множество. Операции над множествами.
Комбинированный тип данных – запись.

21. Файлы последовательного доступа. Файлы прямого доступа
22. Общие сведения о подпрограммах.
23. Определение и вызов подпрограмм.
24. Область видимости и время жизни переменной.
25. Механизм передачи параметров.
26. Организация функций.
27. Рекурсия. Программирование рекурсивных алгоритмов.
28. Основы структурного программирования. Методы структурного программирования.
29. Модульное программирование. Понятие модуля. Структура модуля.
30. Компиляция и компоновка программы. Стандартные модули.
31. Указатели. Описание указателей.
32. Основные понятия и применение динамически распределяемой памяти.
33. Создание и удаление динамических переменных.
34. Структуры данных на основе указателей. Задача о стеке.
35. История развития ООП.
36. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
37. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
38. Требования к аппаратным и программным средствам интегрированной среды разработчика.
39. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты.
40. Форма и размещение на ней управляющих элементов.
41. Панель компонентов и их свойства.
42. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
43. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
44. Дополнительные элементы управления. Свойства компонентов.
45. Виды свойств. Синтаксис определения свойств.
46. Назначения свойств и их влияние на результат.
47. Управление объектом через свойства.
48. Разработка функционального интерфейса приложения.
49. Создание интерфейса приложения.
50. Разработка игрового приложения.

3.3. Перечень практических заданий для проведения зачета с оценкой

Задачи с решениями.

Задача № 1

- 1) **Постановка задачи:** Составить программу нахождения площади прямоугольника со сторонами X и Y.
- 2) **Интерфейс задачи:**

3) Листинг программы:

```
Dim x As Integer, y As Integer, z As Integer
Private Sub Command1_Click()
Text3.Text = Text1.Text + Text2.Text
x = Text1.Text
y = Text2.Text
z = x * y: Text3.Text = z
End Sub
Private Sub Command2_Click()
Form1.Hide: Form2.Show
End Sub
```

Задача № 2

- 1) **Постановка задачи:** Составить программу перевода строки в нижний регистр.
- 2) **Интерфейс задачи:**

3) Листинг программы:

```
Dim x As String, y As String
Private Sub Command1_Click()
x = Text1.Text
y = LCase(x): Text2.Text = y
End Sub
Private Sub Command2_Click()
```

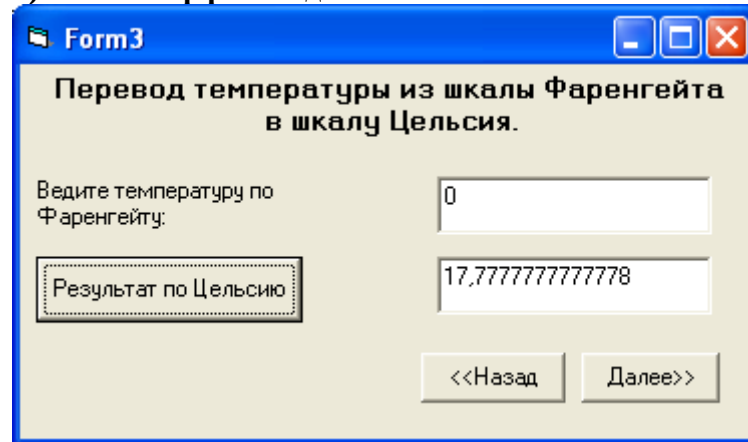
```
Form2.Hide: Form3.Show
End Sub
```

```
Private Sub Command3_Click()
Form2.Hide: Form1.Show
End Sub
```

Задача № 3

1) **Постановка задачи:** Составить программу перевода температуры из шкалы Фаренгейта в шкалу Цельсия (0 F соответствует -17,8, а 0 C соответствует +32 F).

2) **Интерфейс задачи:**



3) **Листинг программы:**

```
Dim x As Variant, y As Variant
Private Sub Command1_Click()
x = Text1.Text
y = (5 * (32 - x) / 9): Text2.Text = y

End Sub
Private Sub Command2_Click()
Form3.Hide: Form4.Show
End Sub
Private Sub Command3_Click()
Form3.Hide: Form2.Show
End Sub
```

Задача № 4

1) **Постановка задачи:** Составить программу определения, в норме ли вес обследуемого пациента (нормой считается вес, равный (рост(см)-100)±5кг).

2) **Интерфейс задачи:**

Form4

В норме ли вес?

Введите рост(см): 120

Введите вес(кг): 74

Результат

<<Назад Далее>>

Project1

Вес не в норме

ОК

3) **Листинг программы:**

```
Dim x As Integer, y As Integer
Private Sub Command1_Click()
x = Text1.Text
y = Text2.Text
If (y < (x - 100) - 5) Or (y > (x - 100) + 5) Then MsgBox "Вес не в норме" Else MsgBox "Вес в норме"
End Sub
```

```
Private Sub Command2_Click()
Form4.Hide: Form5.Show
End Sub
```

```
Private Sub Command3_Click()
Form4.Hide: Form3.Show
End Sub
```

Задача № 5

1) **Постановка задачи:** Составить программу, определяющую сколько раз встречается заданное число (вводится с клавиатуры) в диапазоне от 10 до 352.

2) **Интерфейс задачи:**

Form5

Сколько раз встречается заданное число в диапазоне от 10 до 352?

Введите число: 75

Результат 3

<<Назад Далее>>

3) **Листинг программы:**

```

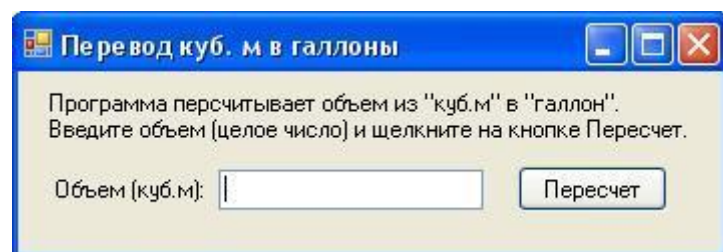
Dim x As String, s As Integer
Private Sub Command1_Click()
x = Text1.Text
k = Len(x)
Select Case k
Case 1
s = 0
For i = 10 To 352
n = Len(i)
For j = 1 To n
For y = 1 To k
If Mid(x, y, 1) = Mid(i, j, k) Then s = s + 1
Next y
Next j
Next i
Case 2
s = 0
For i = 10 To 352
For j = 1 To k
If x = Mid(i, j, 2) Then s = s + 1
Next j
Next i
Case 3
s = 0
For i = 10 To 352
For j = 1 To k
If x = Mid(i, j, 3) Then s = s + 1
Next j
Next i
End Select
Text2.Text = s
End Sub
Private Sub Command2_Click()
Form5.Hide: Form6.Show
End Sub
Private Sub Command3_Click()
Form5.Hide: Form4.Show
End Sub

```

Задача № 6

1) Постановка задачи: Программа пересчитывает из кубического метра в галлон.

2) Интерфейс задачи:



Перевод куб. м в галлоны

Программа пересчитывает объем из "куб.м" в "галлон".
Введите объем (целое число) и щелкните на кнопке Пересчет.

Объем (куб.м):

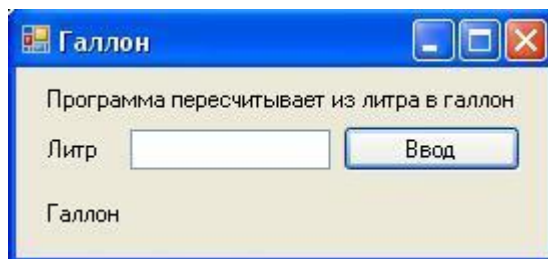
3) Листинг программы:

```
Dim S As Integer
Dim T As Integer
If Not IsNumeric(Vvod.Text) Then
MsgBox("Неверный формат записи!", MsgBoxStyle.OkOnly, Title:="Ошибка")
Else
    S = Vvod.Text
    T = S * 264
Vvod.Text = T
End If
```

Задача № 7

1) **Постановка задачи:** Программа пересчитывает из литра в галлон.

2) **Интерфейс задачи:**



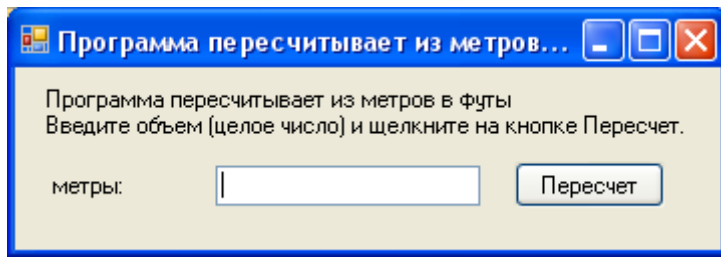
3) Листинг программы:

```
Dim L, G As Double, dial As DialogResult
If Not IsNumeric(TextBox1.Text) Then
    MessageBox.Show("неправильный формат")
    TextBox1.Focus()
Else
    L = TextBox1.Text
    G = L / 3.78541178
    Label4.Text = G
End If
dial = MessageBox.Show("хотите посчитать еще раз?", "Выйти", MessageBoxButtons.YesNo,
    MessageBoxIcon.Asterisk)
If dial = DialogResult.Yes Then
    TextBox1.Text = ""
    Label4.Text = ""
Else
    Application.Exit()
End If
```

Задача № 8

1) **Постановка задачи:** Программа пересчитывает из метров в футы.

2) **Интерфейс задачи:**



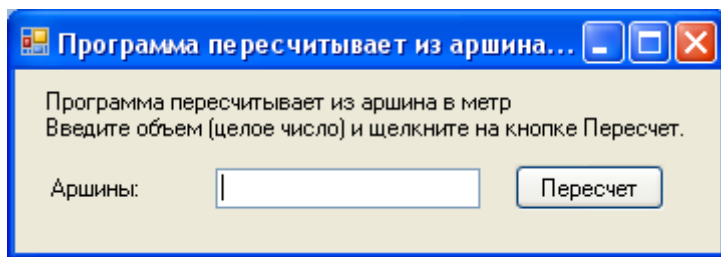
3)Листинг программы:

```
Dim S As Double
Dim T As Double
If Not IsNumeric(Vvod.Text) Then
MsgBox("Неверныйформатзаписи!", MsgBoxStyle.OkOnly, Title:="Ошибка")
Else
    S = Vvod.Text
    T = S * 0,305
Vvod.Text = T
End If
```

Задача № 9

1) **Постановка задачи:** Программа пересчитывает из аршина в метр.

2)**Интерфейс задачи:**



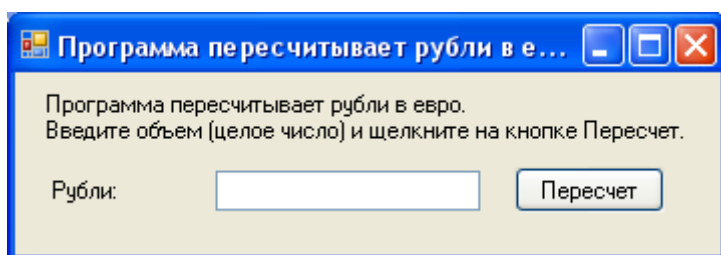
3)Листинг программы:

```
Dim S As Double
Dim T As Double
If Not IsNumeric(Vvod.Text) Then
MsgBox("Неверныйформатзаписи!", MsgBoxStyle.OkOnly, Title:="Ошибка")
Else
    S = Vvod.Text
    T = S * 1.28
Vvod.Text = T
End If
```

Задача № 10

1) **Постановка задачи:** Программа пересчитывает рубли в евро.

2)**Интерфейс задачи:**



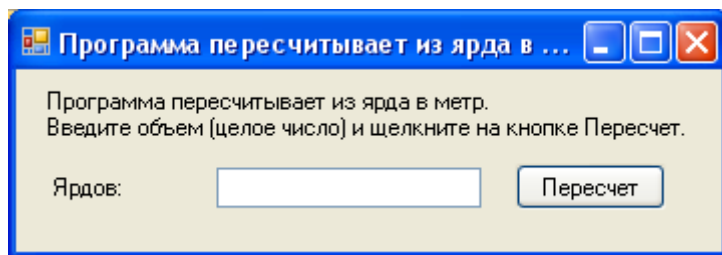
3).Листинг программы:

```
Dim S As Double
Dim T As Double
If Not IsNumeric(Vvod.Text) Then
MsgBox("Неверный формат записи!", MsgBoxStyle.OkOnly, Title:="Ошибка")
Else
    S = Vvod.Text
    T = S * 50
Vvod.Text = T
End If
```

Задача № 11

1) Постановка задачи: Программа пересчитывает из ярда в метр.

2)Интерфейс задачи:



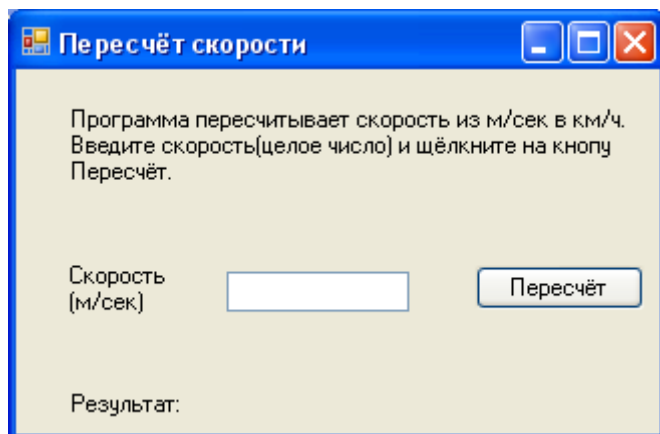
3).Листинг программы:

```
Dim S As Double
Dim T As Double
If Not IsNumeric(Vvod.Text) Then
MsgBox("Неверный формат записи!", MsgBoxStyle.OkOnly, Title:="Ошибка")
Else
    S = Vvod.Text
    T = S * 1.905
Vvod.Text = T
End If
```

Задача № 12

1) Постановка задачи: Программа пересчитывает скорость ветра из м/с в км/ч.

2)Интерфейс задачи:



3)Листинг программы:

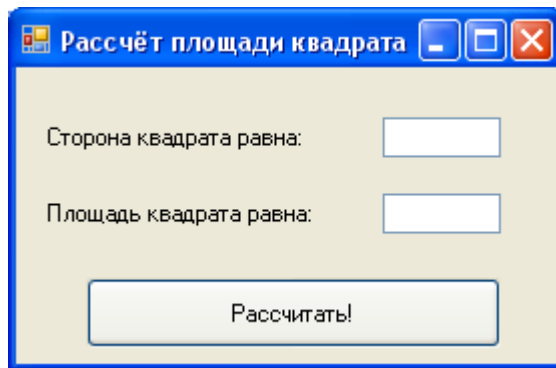
```
Dim Skorost As Integer
Dim rezult As Integer

If Not IsNumeric(txtVvod.Text) Then
    MessageBox.Show("Ошибка!", "Проверка ошибки", MessageBoxButtons.OK,
    MessageBoxIcon.Stop)
Else
    Skorost = txtVvod.Text
    rezult = Skorost * 360
    lblResult.Text = rezult
EndIf
```

Задача № 13

1) **Постановка задачи:** Программа рассчитывает площадь квадрата.

2) **Интерфейс задачи:**



3)Листинг программы:

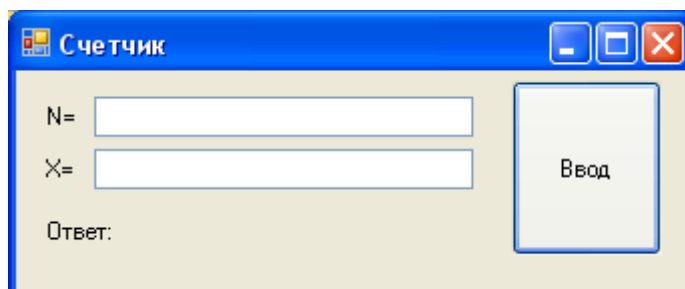
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Button1.Click
        Dim Side As Integer
        Side = TextBox1.Text
        Label1.Text = Square(Side)
        'By value, ByVal, by reference ByRef
    End Sub
    Function Square(ByVal a As Integer) As Integer
        Square = a ^ 2
    End Function
```

Задача № 14

1) **Постановка задачи:** Программа рассчитывает значение по заданной формуле.

$$S = \sum_{i=1}^N \quad S = \sum_{i=1}^N$$

2) **Интерфейс задачи:**



3)Листинг программы:

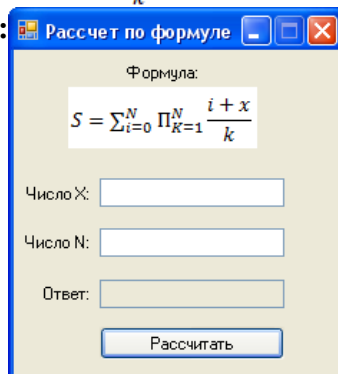
```
Dim N As Long, x, S1, S As Single
Dim k As Integer
Dim i As Integer
Dim dial As DialogResult
Public Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    If Not IsNumeric(TextBox1.Text) Or Not IsNumeric(TextBox2.Text) Then
        MessageBox.Show("неправильный формат")
        TextBox1.Focus()
        TextBox2.Focus()
    Else
        S1 = 1
        N = TextBox1.Text
        x = TextBox2.Text
        For i = 0 To N
            S1 = S1 * MATH.SIN(x)
        Next i
        Label4.Text = S
    End If
EndSub
```

Задача № 15

1) Постановка задачи: Программа рассчитывает значение по заданной формуле.

$$S = \sum_{i=0}^N \prod_{k=1}^N \frac{i+x}{k} S = \sum_{i=0}^N \prod_{k=1}^N \frac{i+x}{k}$$

2)Интерфейс задачи:



3)Листинг программы:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButtonRasschitat.Click
    Dim i As Integer
    Dim k As Integer
    Dim N As Integer
    Dim S As Double = 1
    Dim S1 As Double
    Dim x As Integer
    If Not IsNumeric(TextBoxChisloN.Text) Or Not IsNumeric(TextBoxChisloX.Text) Then
        MsgBox("Неверный формат записи!")
    Else
        N = TextBoxChisloN.Text
        x = TextBoxChisloX.Text
        For i = 0 To N
```

```

For k = 1 To N
    S = S * ((i + x) / k)
Next k
S1 = S1 + S
Next i

```

```

TextBoxOtvet.Text = S1
TextBoxOtvet.Text = (Format(S1, "#.##"))
EndIf
EndSub

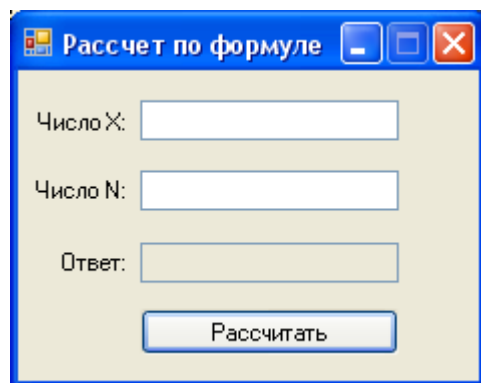
```

Задача № 16

1) **Постановка задачи:** Программа рассчитывает значение по заданной формуле.

$$S = \sum_{i=1}^N \frac{1}{x^i} S = \sum_{i=1}^N \frac{1}{x^i}$$

2) **Интерфейс задачи:**



3) **Листинг программы:**

```

Dim N As Long, x, S1, S As Single
Dim k As Integer
Dim i As Integer
Dim dial As DialogResult
Public Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    If Not IsNumeric(TextBox1.Text) Or Not IsNumeric(TextBox2.Text) Then
        MessageBox.Show("неправильный формат")
        TextBox1.Focus()
        TextBox2.Focus()
    Else
        S1 = 1
        N = TextBox1.Text
        x = TextBox2.Text
        For i = 0 To N
            S1 = S1 + (1/x^i)
        Next i
        Label4.Text = S1
    EndIf
EndSub

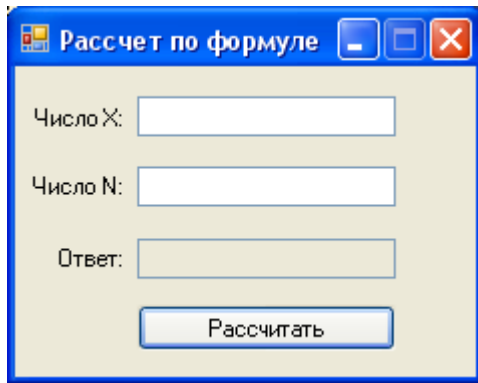
```

Задача № 17

1) **Постановка задачи:** Программа рассчитывает значение по заданной формуле.

$$P = \prod_{i=1}^N (x - i) P = \prod_{i=1}^N (x - i)$$

2) **Интерфейс задачи:**



3) Листинг программы:

```

Dim N As Long, x, S1, S As Single
Dim k As Integer
Dim i As Integer
Dim dial As DialogResult
Public Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    If Not IsNumeric(TextBox1.Text) Or Not IsNumeric(TextBox2.Text) Then
        MessageBox.Show("неправильный формат")
        TextBox1.Focus()
        TextBox2.Focus()
    Else
        S1 = 1
        N = TextBox1.Text
        x = TextBox2.Text
        For i = 1 To N
            S1 = S1 *(x-i)
        Next i
        Label4.Text = S1
    EndIf
EndSub

```

Критерии оценивания:

Оценка «отлично» выставляется обучающемуся, если теоретическое содержание курса освоено полностью, без пробелов; исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует в ответе дополнительный материал все предусмотренные программой задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному; анализирует полученные результаты; проявляет самостоятельность при выполнении заданий.

Оценка «хорошо» выставляется студенту, если теоретическое содержание курса освоено полностью, необходимые практические компетенции в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.

Оценка «удовлетворительно» выставляется студенту, если теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, большинство предусмотренных программой заданий выполнено, но в них имеются ошибки, при ответе на поставленный вопрос студент допускает неточности, недостаточно правильные

формулировки, наблюдаются нарушения логической последовательности в изложении программного материала.

Оценка «неудовлетворительно» выставляется студенту, если он не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, необходимые практические компетенции не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, качество их выполнения оценено числом баллов, близким к минимальному.

Комплект тестовых заданий для проверки сформированности компетенций

Номер задания	Содержание вопроса	Правильный ответ	Компетенция
1.	Слово «алгоритм» происходит +: от имени арабского учёного Абу Абдуллах Мухаммеда ибн Муса аль-Хорезми -: от индийского слова, которое переводится как «цифра» и «шифр» -: от индийского слова «алгорифм», которое впервые встречается в труде «Algoritmi de numero Indorum» («Алгоритмы о счёте индийском»)	+: от имени арабского учёного Абу Абдуллах Мухаммеда ибн Муса аль-Хорезми	ОК 02
2.	Неприменимость алгоритма к допустимым начальным данным заключается в том, что +: алгоритмический процесс никогда не оканчивается, либо его выполнение во время одного из шагов наталкивается на препятствие, заходит в тупик -: алгоритмический процесс не является бесконечным, но он безрезультатно обрывается -: алгоритмический процесс приводит к неверному результату, либо наталкивается на препятствие -: алгоритмический процесс заиклиивается и приводит к неверному результату	+: алгоритмический процесс никогда не оканчивается, либо его выполнение во время одного из шагов наталкивается на препятствие, заходит в тупик	ОК 03
3.	Термин «рекурсия» восходит +: к латинскому слову, которое переводится как «возвращение» -: к греческому слову, которое переводится как «приближение» -: к греческому слову, которое переводится как «вычисление»		ОК 04

4.	Известно, что абстрактный алфавит – это конечная совокупность объектов, называемых буквами этого алфавита. Что может выступать в качестве букв алфавита? +: цифры +: любые значки +: целые фразы +: рисунки +: слова («ключевые» слова) -: средства автоматизации	+: цифры +: любые значки +: целые фразы +: рисунки +: слова («ключевые» слова)	ПК 1.5
5.	Общая теория алгоритмов начала развиваться в связи с необходимостью решить вопросы +: о строгой выводимости и эффективном вычислении -: об алгоритмическом процессе как совокупности действий -: о массовых задачах, имеющих одинаковые способы решения	+: о строгой выводимости и эффективном вычислении	ОК 02
6.	Определение «конечность действий алгоритма решения задач, позволяющая получить желаемый результат при допустимых исходных данных за конечное число шагов» относится к свойству алгоритмов, которое называется +: выполнимость -: определенность -: дискретность -: массовость	+: выполнимость	ОК 03
7.	Свойствами алгоритмов являются +: определённости +: дискретность +: результативность -: значимость +: массовость	+: определённость +: дискретность +: результативность	ПК 3.8
8.	Алгоритм можно записать +: словами (в виде словесного описания) +: в виде блок- схемы +: на алгоритмическом языке -: любыми символами	+: словами (в виде словесного описания) +: в виде блок- схемы +: на алгоритмическом языке	ПК 1.5

9.	Алгоритмы бывают +: линейными -: эллипсовидными +: ветвящимися +: циклическими	+: линейными +: ветвящимися +: циклическими	ОК 02
10.	Выберите интуитивно описательное определение понятия «алгоритм» -: алгоритм – это совокупность методов и инструментов для достижения желаемого результата, представленная на языке высокого уровня с описанием последовательности действий для решения задачи или достижения поставленной цели, в широком смысле – применение научного знания для решения практических задач -: алгоритм – это правила выполнения основных операций обработки данных или способ достижения какой-либо цели, решения конкретной задачи; совокупность приемов или операций практического или теоретического освоения действительности -: алгоритм – это описание вычислений по математическим формулам +: алгоритм – это чёткая система инструкций, определяющая дискретный детерминированный процесс, ведущий от варьируемых начальных данных (входов) к искомому результату (выходу), если таковой существует, через конечное число тактов работы алгоритма; если же искомого результата не существует, то вычислительный процесс либо никогда не оканчивается, либо попадает в тупик	+: алгоритм – это чёткая система инструкций, определяющая дискретный детерминированный процесс, ведущий от варьируемых начальных данных (входов) к искомому результату (выходу), если таковой существует, через конечное число тактов работы алгоритма; если же искомого результата не существует, то вычислительный процесс либо никогда не оканчивается, либо попадает в тупик	ОК 04
11.	Определение «однозначность выполнения каждого отдельного шага преобразования информации» относится к свойству алгоритмов, которое называется +: определенность -: выполнимость -: дискретность -: массовость	+: определенность	ПК 2.4

12.	Расположите команды так, чтобы получился алгоритм «Посади саженец» 1: Возьми лопату 2: Выкопай яму 3: Посади саженец 4: Засыпь яму землей 5: Возьми лейку и полей саженец 6: Поставь лопату и лейку на место	1: Возьми лопату 2: Выкопай яму 3: Посади саженец 4: Засыпь яму землей 5: Возьми лейку и полей саженец 6: Поставь лопату и лейку на место	ПК 3.8
13.	Основным критерием правильного выполнения алгоритма является +: порядок команд -: способ представления алгоритма -: название алгоритма -: краткость записи действий алгоритма	+: порядок команд	ОК 02
14.	Расположите команды так, чтобы получился правильный алгоритм «Подготовить сообщение по предмету и выступить перед аудиторией» 1: Выбрать тему сообщения 2: Подобрать нужную литературу 3: Из источников отобрать информацию 4: Выступить с сообщением	1: Выбрать тему сообщения 2: Подобрать нужную литературу 3: Из источников отобрать информацию 4: Выступить с сообщением	ПК 1.5
15.	Какие из документов являются алгоритмом? +: инструкция получения денег в банкомате +: рецепт приготовления пирога +: схема сбора модели самолета -: правила техники безопасности -: атлас автодорог -: проездной билет на троллейбус	+: инструкция получения денег в банкомате +: рецепт приготовления пирога +: схема сбора модели самолета	ПК 3.8
16.	Свойство алгоритма, из которого следует, что алгоритм должен представлять процесс решения задачи как последовательное выполнение простых шагов, называется +: дискретность -: однозначность -: массовость -: результативность	+: дискретность	ОК 03

17.	Свойство алгоритма, из которого следует, что каждое правило алгоритма должно быть четким и однозначным, называется +: однозначность -: дискретность -: массовость -: результативность	+: однозначность	ПК 3.8
18.	Результативность – это свойство алгоритма, которое означает +: обязательное получение результата за конечное число шагов -: однозначное понимание и исполнение алгоритма исполнителем -: разбиение алгоритма на конечное число элементарных шагов -: применение алгоритма к целому классу однотипных задач	+: обязательное получение результата за конечное число шагов	ПК 3.8
19.	Свойство алгоритма, которое состоит в том, что за конечное число шагов алгоритм должен приводить к решению задачи или после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов, называется +: результативность -: однозначность -: дискретность -: массовость	+: результативность	ОК 04
20.	Свойство алгоритма, которое означает, что алгоритм решения задачи должен быть применим для некоторого класса задач, различающихся лишь исходными данными, называется +: массовость -: результативность -: однозначность -: дискретность	+: массовость	ОК 03

21.	Дискретность – свойство алгоритма, которое подразумевает +: разбиение алгоритма на конечное число элементарных шагов -: что команды должны следовать последовательно друг за другом -: что каждая команда должна быть описана в расчете на конкретного исполнителя	+: разбиение алгоритма на конечное число элементарных шагов	ОК 01
22.	Однозначность – свойство алгоритма, которое означает +: однозначное понимание и исполнение алгоритма исполнителем -: разбиение алгоритма на конечное число элементарных шагов -: обязательное получение результата за конечное число шагов; -: применение алгоритма к целому классу однотипных задач;	+: однозначное понимание и исполнение алгоритма исполнителем	ПК 2.4
23.	Массовость – свойство алгоритма, которое означает +: применимость алгоритма к целому классу однотипных задач -: однозначное понимание и исполнение алгоритма исполнителем -: обязательное получение результата за конечное число шагов -: разбиение алгоритма на конечное число элементарных шагов	+: применимость алгоритма к целому классу однотипных задач	ПК 1.5
24.	Способ представления алгоритма, который содержит описание алгоритма на естественном языке, называется +: словесным -: псевдокодом -: графическим -: программой на алгоритмическом языке	+: словесным	ОК 03
25.	Способ представления алгоритма, с помощью изображения последовательности связанных функциональных блоков, называется +: графическим -: словесным	+: графическим	ОК 04

	-: псевдокодом -: программой на алгоритмическом языке		
26.	Графический способ представления алгоритма – это +: представление алгоритма с помощью изображения последовательности связанных функциональных блоков -: представление алгоритма в форме таблиц и расчетных формул -: система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения	+: представление алгоритма с помощью изображения последовательности связанных функциональных блоков	ПК 3.8
27.	Способ представления алгоритма, в виде последовательности операторов языка программирования, называется +: программой на алгоритмическом языке -: графическим -: словесным -: псевдокодом	+: программой на алгоритмическом языке	ПК 2.4
28.	Алгоритм может быть выполнен процессором компьютера, если +: алгоритм записан в двоичном коде (на машинном языке) -: алгоритм записан на английском языке -: алгоритм записан на естественном языке -: алгоритм зафиксирован в виде блок-схемы	+: алгоритм записан в двоичном коде (на машинном языке)	ПК 3.8
29.	Для изображения начала алгоритма в блок-схеме используется геометрическая фигура +: овал (прямоугольник с закруглёнными углами) -: ромб -: прямоугольник -: параллелограмм -: шестиугольник	+: овал (прямоугольник с закруглёнными углами)	ОК 02
30.	К какой форме представления алгоритма можно отнести схему сборки модели самолета +: графической -: пошаговой	+: графической	ПК 3.8

	-: табличной -: записи на алгоритмическом языке		
31.	Для изображения конца алгоритма в блок-схеме используется геометрическая фигура +: овал -: ромб -: прямоугольник -: параллелограмм -: шестиугольник I:	+: овал	ПК 2.4
32.	Для изображения в блок-схеме ввода/вывода данных используется геометрическая фигура +: параллелограмм -: ромб -: прямоугольник -: овал -: шестиугольник	+: параллелограмм	ОК 01
33.	Какая геометрическая фигура используется для изображения действия в блок-схеме? +: прямоугольник -: ромб -: овал -: параллелограмм -: шестиугольник	+: прямоугольник	ОК 04
34.	Какая геометрическая фигура используется для изображения условия в блок-схеме? +: ромб -: прямоугольник -: овал -: параллелограмм -: шестиугольник	+: ромб	ПК 1.5
35.	Приведите в соответствие термины и определения: L1: форма записи порядка действий исполнителя с помощью графических блоков L2: последовательность указаний исполнителю, записанная на каком-либо языке программирования	L1: форма записи порядка действий исполнителя с помощью графических блоков L2: последовательность указаний исполнителю, записанная на каком-	ОК 01

	L3: конечная последовательность указаний (команд) исполнителю, выполнив которую, он достигает поставленную цель или решает определенную задачу L4: запись порядка действий с помощью последовательности фраз, предложений на естественном языке R1: блок-схема R2: программа R3: алгоритм R4: словесное описание	либо языке программирования L3: конечная последовательность указаний (команд) исполнителю, выполнив которую, он достигает поставленную цель или решает определенную задачу L4: запись порядка действий с помощью последовательности фраз, предложений на естественном языке R1: блок-схема R2: программа R3: алгоритм R4: словесное описание	
36.	Какой алгоритм называется линейным? +: алгоритм, действия которого выполняются друг за другом -: алгоритм, обеспечивающий выбор одного из альтернативных путей работы в зависимости от результата проверки условия (да или нет) -: алгоритм, в котором одни и те же операции выполняются многократно -: алгоритм, который использует вспомогательные алгоритмы	+: алгоритм, действия которого выполняются друг за другом	ОК 01
37.	Алгоритмическая конструкция, в которой действия расположены друг за другом, называется +: следование -: ветвление -: цикл	+: следование	ПК 2.4
38.	Какой алгоритм называется алгоритмом с ветвлением? +: алгоритм, обеспечивающий выбор одного из альтернативных путей работы в зависимости от результата проверки условия (да или нет)	+: алгоритм, обеспечивающий выбор одного из альтернативных путей работы в зависимости от результата проверки условия (да или нет)	ОК 03

	-: алгоритм, действия которого выполняются друг за другом -: алгоритм, в котором одни и те же операции выполняются многократно -: алгоритм, который использует вспомогательные алгоритмы		
39.	Алгоритм, в котором после проверки условия в разных ситуациях выполняются разные наборы команд, называется +: алгоритмом с ветвлением -: линейным алгоритмом -: циклическим алгоритмом	+ : алгоритмом с ветвлением	ОК 01
40.	Какой алгоритм называется циклическим? +: алгоритм, в котором одни и те же операции выполняются многократно -: алгоритм, обеспечивающий выбор одного из альтернативных путей работы в зависимости от результата проверки условия (да или нет) -: алгоритм, действия которого выполняются друг за другом -: алгоритм, который использует вспомогательные алгоритмы	+ : алгоритм, в котором одни и те же операции выполняются многократно	ОК 03

Критерии оценивания:

Оценка «отлично» выставляется студенту, если он правильно ответил на 90-100% от общего числа вопросов тестовых заданий.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 70-89% от общего числа вопросов тестовых заданий.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 50- 69% от общего числа вопросов тестовых заданий.

Оценка «неудовлетворительно» выставляется студенту, если он правильно ответил менее чем на 50% от общего числа вопросов тестовых заданий.