

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 18.06.2026 14:18:48

Уникальный программный ключ:

990bea3aecc48c23bd8699e48288f8d19d0c800

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института
перспективной инженерии,
кандидат технических наук,
доцент

А.А. Порохня

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Объектно-ориентированное программирование

Направление подготовки	09.03.02 Информационные системы и технологии
Направленность (профиль)	Прикладное программирование в интеллектуальных информационных системах
Форма обучения	очная
Год начала обучения	2026
Реализуется в 5 семестре	

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимся установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Объектно-ориентированное программирование»

3. Разработчик Николаев Е.И., доцент департамента цифровых, робототехнических систем и электроники

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Дроздова В. И., профессор департамента цифровых, робототехнических систем и электроники

Члены комиссии:

Крахоткина Е.В., доцент департамента цифровых, робототехнических систем и электроники

Николаев Е.И., доцент департамента цифровых, робототехнических систем и электроники

Представитель организации-работодателя: А. В. Новиков, начальник отдела АСУТП ООО «ЕНДС-Ставрополь»

Экспертное заключение: ФОС по дисциплине «Объектно-ориентированное программирование» позволяет оценить уровень сформированности компетенций. Рекомендовать к использованию в учебном процессе.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
<i>Компетенция: ПК-1</i>				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-1 _{ид-1.1}	Понимает основные понятия и принципы алгоритмов. Знает наиболее распространенные алгоритмы и их применение в различных областях. Способен применять готовые алгоритмы для решения простых практических задач. Не имеет опыта в адаптации алгоритмов к конкретным практическим задачам.	Обладает глубоким пониманием основных алгоритмов и их принципов работы. Умеет адаптировать и модифицировать готовые алгоритмы для решения задач конкретной области. Имеет опыт применения алгоритмов в практических проектах. Способен обосновывать применение алгоритмов и доказывать их эффективность.	Владеет широким спектром алгоритмов и глубоко разбирается в принципах их работы. Умеет эффективно адаптировать алгоритмы к самым сложным практическим задачам. Имеет опыт разработки собственных алгоритмов и методов их оптимизации. Может проводить обучение и консультирование других специалистов по алгоритмам.	Создает уникальные и высокоэффективные алгоритмические решения для сложных профессиональных задач. Имеет публикации и научные работы по алгоритмам и их применению. Способен выступать на конференциях и вести обучающие курсы по алгоритмике.
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-1 _{ид-1.2}	Понимание основных понятий и принципов языка высокого уровня (например, переменные, операторы, функции). Умение разрабатывать простые приложения с использованием языка высокого уровня. Знание основных интеллектуальных технологий и их возможностей.	Демонстрирует глубокое понимание синтаксиса и особенностей языка высокого уровня; умение разрабатывать сложные приложения с использованием языка высокого уровня. Знает основные алгоритмы и структуры данных. Способен использовать интеллектуальные технологии (например, машинное обучение или нейронные сети) в разработке приложений.	Демонстрирует глубокое знание языка высокого уровня, включая продвинутые возможности и функции. Способен проектировать и реализовывать сложные приложения, используя язык высокого уровня и интеллектуальные технологии. Способен оптимизировать и улучшать производительность приложений. Способен анализировать и решать сложные	Демонстрирует глубокое и полное владение языком высокого уровня и его возможностями. Способен разрабатывать сложные и инновационные приложения, используя язык высокого уровня и интеллектуальные технологии. Способен создавать собственные алгоритмы и модели для решения

			задачи с использованием интеллектуальных технологий.	сложных задач. Способен применять передовые исследования и технологии в области интеллектуальных технологий в разработке приложений.
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-1 _{ид-1.3}	Знание основных принципов и методов тестирования программного обеспечения. Способен применять базовые инструменты и технологии для автоматизации тестирования. Способен проводить ручное тестирование программного обеспечения и отлаживать проблемы	Демонстрирует понимание и применяет различных методы тестирования, такие как модульное тестирование, интеграционное тестирование, функциональное тестирование и др. Способен создавать и поддерживать автоматизированные тестовые скрипты и сценарии. Применяет инструменты для отладки и профилирования программного обеспечения	Способен создавать и управлять комплексными автоматизированными тестовыми системами. Демонстрирует навыки использования инструментальных средств для контроля качества и управления дефектами, таких как JIRA, QC/ALM и другие. Демонстрирует глубокое понимание принципов разработки и отладки программного обеспечения, чтобы эффективно выявлять и исправлять ошибки	Демонстрирует экспертное знание и опыт применения современных автоматизированных технологий тестирования, таких как контейнеризация, облачное тестирование, DevOps и другие. Способен создавать и реализовывать стратегии тестирования программного обеспечения. Владеет современными инструментами и методами для поиска и устранения уязвимостей безопасности в программном обеспечении
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-1 _{ид-1.4}	Понимает основные принципы непрерывного развертывания и доставки программного обеспечения. Имеет представление о современных инструментах и технологиях, используемых для автоматизации процессов разверты-	Обладает углубленными знаниями о принципах и процессах непрерывного развертывания и доставки ПО. Владеет различными инструментами и технологиями, используемыми для автоматизации процессов	Обладает глубокими знаниями и опытом в области непрерывного развертывания и доставки ПО. Владеет широким набором инструментов и технологий для автоматизации процессов разверты-	Обладает экспертным уровнем по непрерывному развертыванию и доставке ПО и вносит значительный вклад в развитие данной области. Владеет

	вания и доставки ПО. Способен использовать базовые функции инструментов непрерывного развертывания, таких как системы контроля версий и среды разработки	развертывания и доставки ПО. Может использовать продвинутое функции инструментов непрерывного развертывания, таких как CI/CD платформы и системы управления конфигурациями. Способен проектировать и настраивать сложные цепочки развертывания и доставки ПО. Понимает преимущества и ограничения контейнеризации и виртуализации и может использовать их эффективно в своей работе	вания и доставки ПО. Способен разрабатывать и реализовывать инновационные подходы к непрерывному развертыванию и доставке ПО. Может управлять сложными проектами по развертыванию и доставке ПО, включая управление ресурсами, распределением задач и контролем качества. Обладает глубоким пониманием принципов контейнеризации и виртуализации и может использовать их для оптимизации процессов развертывания и доставки ПО	всеми современными инструментами и технологиями, используемыми в непрерывном развертывании и доставке ПО. Способен разрабатывать и реализовывать стратегии и методологии развертывания и доставки ПО для организаций любого масштаба. Может эффективно управлять сложными проектами и командами, занимающимися развертыванием и доставкой ПО. Вносит вклад в научные исследования в области непрерывного развертывания и доставки ПО и активно развивает эту область
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ПК-1_{ИД-1.5}	Основные знания в области разработки приложений для различных платформ. Умение использовать основные инструменты и технологии для разработки приложений (например, различные языки программирования, фреймворки, библиотеки). Способность разрабатывать простые приложения для од-	Демонстрирует знания в области разработки приложений для различных платформ. Способен использовать различные инструменты и технологии для разработки полноценных приложений. Способен разрабатывать сложные приложения для одной или нескольких платформ	Демонстрирует глубокие знания в области разработки приложений для различных платформ. Способен применять передовые инструменты и технологии для разработки масштабных приложений (например, работа с облачными сервисами, многопоточность,	Демонстрирует обширные знания и использует практический опыт в области разработки приложений для различных платформ. Демонстрирует глубокое понимание принципов работы различных

	ной или нескольких платформ.		оптимизация производительности). Способен разрабатывать сложные и инновационные приложения для одной или нескольких платформ	платформ и их особенностей. Способен разрабатывать сложные и высоконагруженные приложения для различных платформ. Способен анализировать и решать сложные технические проблемы, связанные с разработкой приложений для различных платформ
--	------------------------------	--	--	---

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Но- мер зада- ния	Правильный ответ	Содержание вопроса	Компетен- ция
		Вопросы к экзамену:	
1.		1. История развития технологий программирования Windows-приложений 2. Обобщенная структура .NET Framework 3. Основные понятия, механизмы и соглашения технологии .NET 4. Особенности выполнения приложений .NET 5. Общие сведения о системе типов .NET 6. Компоновочные блоки, пространства имен, типы 7. Спецификаторы доступа типов и членов типов 8. Объявление и инициализация переменных и констант в C# 9. Условный оператор 10. Оператор выбора 11. Оператор цикла for 12. Цикл с предусловием while 13. Цикл с постусловием do ... while 14. Простые массивы 15. Многомерные массивы 16. Зубчатые массивы 17. Классы и структуры 18. Данные-члены класса 19. Функции-члены класса 20. Методы классов 21. Операции. Перегрузка операций 22. Наследование реализации 23. Наследование интерфейсов 24. Интерфейсы и типы коллекций 25. Списки 26. Очередь	ПК-1

27. Стек

1. Напишите функцию для вычисления значения выражения. Перечень необходимых формальных параметров необходимо определить самостоятельно:

$$Z = \begin{cases} 1 - \frac{X}{2} + \frac{Y^2}{6} - \frac{X^3}{24} + \frac{Y^4}{120} - \dots; \\ \sum_{i=1}^N \sum_{j=1}^R \frac{3 + a \cdot j}{b \cdot i}. \end{cases}$$

2. Напишите функцию для вычисления значения выражения. Перечень необходимых формальных параметров необходимо определить самостоятельно:

$$Z = \begin{cases} -\frac{1}{2} + \frac{Y}{6} - \frac{X^2}{24} + \frac{Y^3}{120} - \dots; \\ \sum_{i=1}^N \sum_{j=1}^R \frac{i^2 + j}{i^j}. \end{cases}$$

3. Напишите функцию для вычисления значения выражения. Перечень необходимых формальных параметров необходимо определить самостоятельно:

$$Z = \begin{cases} -\frac{1}{1} + \frac{Y}{2} - \frac{X^2}{3} + \frac{Y^3}{4} - \dots; \\ \sum_{i=1}^N \sum_{j=1}^R \frac{j^2 + h \cdot i}{i^j + c \cdot j}. \end{cases}$$

4. Напишите функцию для вычисления значения выражения. Перечень необходимых формальных параметров необходимо определить самостоятельно:

$$Z = \begin{cases} -\frac{p^2}{2} + \frac{p^3}{6} - \frac{p^4}{24} + \frac{p^5}{120} - \dots; \\ \sum_{i=1}^N \sum_{j=1}^R \frac{i^2 + b \cdot j}{c \cdot i^3}. \end{cases}$$

5. Напишите функцию для вычисления значения выражения. Перечень необходимых формальных параметров необходимо определить самостоятельно:

$$Z = \begin{cases} -\frac{X}{2} + \frac{p \cdot X^2}{6} - \frac{p^2 \cdot X^3}{24} + \frac{p^3 \cdot X^4}{120} - \dots; \\ \sum_{i=1}^N \sum_{j=1}^R \frac{i^2 + a \cdot j^2}{i^3 + j^3}. \end{cases}$$

6. Напишите функцию для вычисления значения выражения. Перечень необходимых формальных параметров необходимо определить самостоятельно:

$$Z = \begin{cases} -\frac{X}{2} + \frac{X^2}{6} - \frac{X^3}{24} + \frac{X^4}{120} - \dots; \\ \sum_{i=1}^N \sum_{j=1}^R \frac{a \cdot i^2}{i^3 + b \cdot j^3}. \end{cases}$$

7. Напишите функцию для вычисления значения выражения. Перечень необходимых формальных параметров необходимо определить самостоятельно:

$$Z = \begin{cases} -\frac{f}{2} + \frac{X \cdot f^2}{6} - \frac{X^2 \cdot f^3}{24} + \frac{X^3 \cdot f^4}{120} - \dots; \\ \sum_{i=1}^N \sum_{j=1}^R \frac{a \cdot (i+j)}{i \cdot (i+2)}. \end{cases}$$

8. Приведен фрагмент программы с объявлением классов. Поясните назначение кода. Найдите ошибки компиляции (если имеются) и поясните пути устранения ошибок:

```
public class MainObject: IString, IDraw
{
    int data;

    public string GetStr()
    {
        return "Данные: " + data.ToString();
    }

    public void Draw()
    {
        Console.WriteLine(data.ToString());
    }
}
```

```
public interface IDraw
{
    void Draw();
}

public interface IString
{
    string GetStr();
}
```

```
static void Main(string[] args)
{
    MainObject ob = new MainObject();
    ob.Draw();
    Console.WriteLine(ob.GetStr());
}
```

9. Приведен фрагмент программы с объявлением классов. Поясните назначение кода. Найдите ошибки компиляции (если имеются) и поясните пути устранения ошибок:

```
public interface IDraw
{
    void Draw();
}

public interface IString
{
    string GetStr();
}
```

```
public class MainObject: IString, IDraw
{
    int data;

    public string GetStr()
    {
        return "Данные: " + data.ToString();
    }
}
```

10. Приведен фрагмент программы с объявлением классов. Поясните назначение кода. Найдите ошибки компиляции (если имеются) и поясните пути устранения ошибок:

```
public interface IString
{
    string GetStr();
}

public class Data1 : IString
{
    int data;

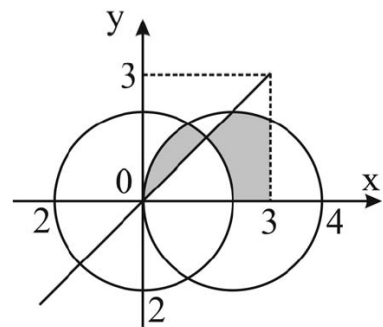
    public string GetStr()
    {
        return "Данные: " + data.ToString();
    }
}
```

```
public class Data2 : IString
{
    string data;

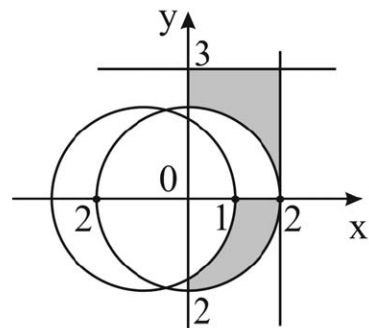
    public string GetStr()
    {
        return data;
    }
}
```

Продвинутый:

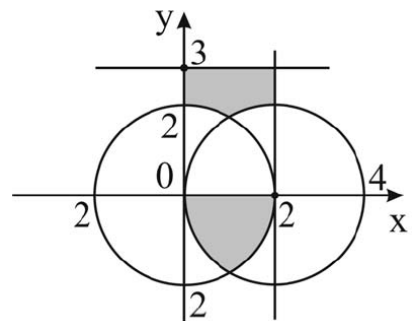
1. Напишите функцию для проверки принадлежности точки закрашенной области.
Схема формирования закрашенной области представлена на рисунке:



2. Напишите функцию для проверки принадлежности точки закрашенной области.
 Схема формирования закрашенной области представлена на рисунке:

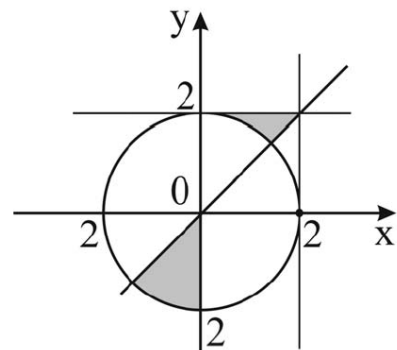


3. Напишите функцию для проверки принадлежности точки закрашенной области.
 Схема формирования закрашенной области представлена на рисунке:



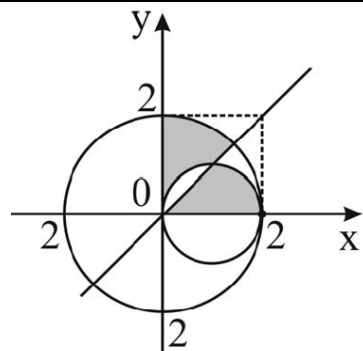
4. Напишите функцию для проверки принадлежности точки закрашенной области.

Схема формирования закрашенной области представлена на рисунке:



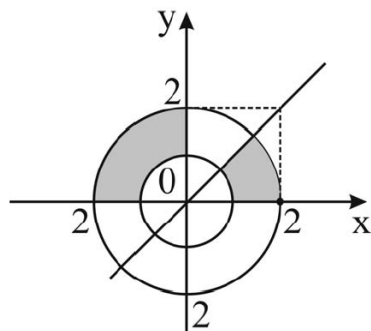
5. Напишите функцию для проверки принадлежности точки закрашенной области.

Схема формирования закрашенной области представлена на рисунке:



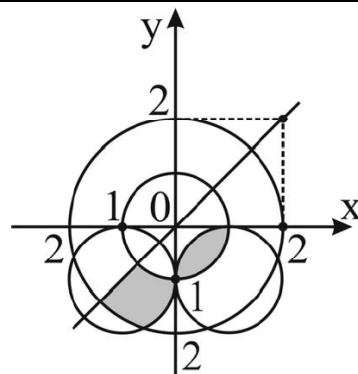
6. Напишите функцию для проверки принадлежности точки закрашенной области.

Схема формирования закрашенной области представлена на рисунке:

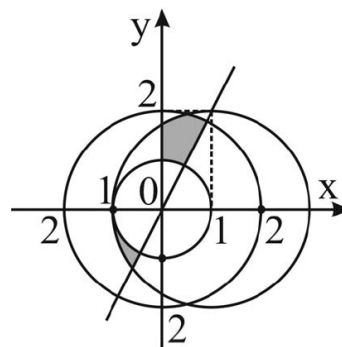


7. Напишите функцию для проверки принадлежности точки закрашенной области.

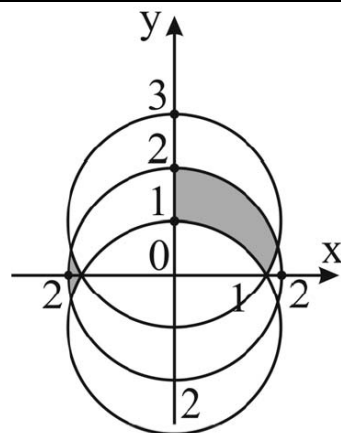
Схема формирования закрашенной области представлена на рисунке:



8. Напишите функцию для проверки принадлежности точки закрашенной области.
 Схема формирования закрашенной области представлена на рисунке:

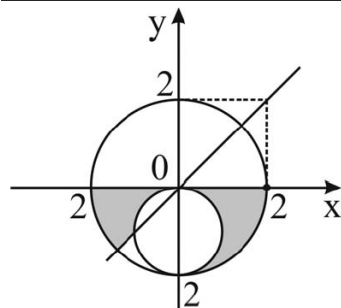


9. Напишите функцию для проверки принадлежности точки закрашенной области.
 Схема формирования закрашенной области представлена на рисунке:



10. Напишите функцию для проверки принадлежности точки закрашенной области.

Схема формирования закрашенной области представлена на рисунке:



Примерные задания для курсовой работы

Примерная тематика курсовых работ (проектов) (для каждого вида деятельности, предусмотренного образовательной программой высшего образования)

Направление деятельности	Примерная тематика
Проектный	1. Разработка приложения «Интерактивная карта»
Проектный	2. Разработка приложения «Менеджер фотографий»
Проектный	3. Разработка приложения «Траектория движения»
Проектный	4. Разработка приложения «Игра «Собирает монеты»

		Проектный 5. Разработка приложения «Свертка изображения» Проектный 6. Разработка приложения «Матричный калькулятор» Проектный 7. Разработка приложения «Аквалайзер» Проектный 8. Разработка приложения «Графический пароль» Проектный 9. Разработка приложения «Игра «Танки» Проектный 10. Разработка приложения «Игра «Змейка» Проектный 11. Разработка приложения «Расчет стоимости поездки» Проектный 12. Разработка приложения «Расчет баллистической траектории» Проектный 13. Разработка приложения «Интерактивный анализ набора данных» Проектный 14. Разработка приложения «Моделирование популяции» Проектный 15. Разработка приложения «Обработка изображения фильтрами» Проектный 16. Разработка приложения «Игра «Цветы против грибов» Проектный 17. Разработка приложения «Фракталы» Проектный 18. Разработка приложения «Искусственный интеллект» Проектный 19. Разработка приложения «Филиалы и сотрудники» Проектный 20. Разработка приложения «Игра «Ракета» Проектный 21. Разработка приложения "Игра «Блиц» Проектный 22. Разработка приложения «Поиск текста в документах» Проектный 23. Разработка приложения «Анализ матричных данных»⁷ Проектный 24. Разработка приложения «Игра «Поиск крота» Проектный 25. Разработка приложения «Вычисления со скобками»	
		Задания закрытого типа	
1.	1, 2	Какие типы относятся к типам-делегатам? 1) Action<T> 2) Func<T> 3) List<T> 4) IEnumerable<T>	
2.	1	Какой метод позволяет реализовать остановку выполнения программы и ожидание продолжения выполнения в течении заданного количества миллисекунд? 1) Thread.Sleep() 2) System.Console()	ПК-1

		3) Math.Wait() 4) Waiting.Stop()	
3.	1	Какой класс используется для представления потока в многопоточном приложении? 1) Thread 2) Task 3) MultiTask 4) Waiting 5) Action<T>	ПК-1
4.	2	Какое пространство имен является корневым в библиотеке .NET 1) Data 2) System 3) Object 4) Input 5) IO	ПК-1
5.	1,2,4	В каких фрагментах кода объявляются переменные? 1) <code>int a;</code> 2) <code>int b = 7;</code> 3) <code>a = 123;</code> 4) <code>string str = "Hello, World!!!";</code>	ПК-1
6.	3	С помощью какого зарезервированного слова происходит объявление пространства имен? 1) using 2) break 3) namespace 4) foreach 5) name	ПК-1
7.	1	Даны определения классов на языке C#.	ПК-1

		<pre> class Child : Parent { int b; void E() { b = a; } } public class Parent { protected int a; } class Child1 : Parent { int c; public Child1() { } } </pre> Существуют ли ошибки в данном фрагменте? 1) ошибок нет 2) переменная a является недоступной в производном классе, следовательно, присвоение b = a некорректно 3) в двух классах отсутствуют конструкторы 4) все поля классов являются закрытыми, поэтому их использование не представляется возможным	
8.	2	Что такое конструктор по умолчанию в языке C#? 1) Конструктор по умолчанию представляет собой метод, который вызывается перед созданием объекта и устанавливает значения всех полей класса в предопределенные значения 2) Программист, определяя класс, может явно не реализовывать конструктор. В этом случае .NET Framework использует пустой метод в качестве конструктора по умолчанию 3) Программист определяет сколько угодно своих конструкторов, но .NET Framework будет игнорировать их, если не отключен конструктор по умолчанию	ПК-1
9.	4	Даны определения классов на языке C#.	ПК-1

		<pre> public class Parent { protected int a; Parent() { a = 0; } } class Child1 : Parent { int c; public Child1() { :base() } } </pre> Существуют ли ошибки в данном фрагменте? 1) ошибок нет 2) конструктор базового класса private, а производного – public 3) все поля классов являются закрытыми, поэтому их использование не представляется возможным 4) конструктор базового класса не виден из производного класса	
10.	4	Даны определения классов на языке C#. <pre> public class Parent { protected int a; protected Parent() { a = 0; } } class Child1 : Parent { int c; public Child1() { :base() } } </pre> Существуют ли ошибки в данных фрагментах? 1) конструктор базового класса не виден из производного класса 2) конструктор базового класса private, а производного – public 3) все поля классов являются закрытыми, поэтому их использование не представляется возможным 4) ошибок нет	ПК-1
11.	1	Даны определения классов на языке C#.	ПК-1

		<pre> public class Parent: Child1 { protected int a; protected Parent() { a = 0; } } class Child1 : Child { int c; } class Child : Parent { int b; void E() { b = a; } } </pre> Существуют ли ошибки в данных фрагментах? 1) классы наследуются по кругу 2) неверное использование модификатора доступа protected в классе Parent 3) в двух классах отсутствуют конструкторы 4) все поля классов являются закрытыми, поэтому их использование не представляется возможным 5) ошибок нет	
12.	1	Сколько наследников может быть у представленного класса? <pre> public class Child : Parent { int b; void E() { b = a; } } </pre> 1) столько, сколько реализует программист (принципиальных ограничений нет) 2) количество наследников ограничивается в базовом классе Parent 3) не более 128	ПК-1

13.	2	В языке C# программист определил класс следующим образом) <pre> public class Parent { protected int a; protected Parent() { a = 0; } } </pre> Какой класс является базовым для данного класса? 1) System.Data 2) System.Object 3) System.IO 4) нет базового класса	ПК-1
14.	3	В языке C# программист определил класс следующим образом) <pre> public class Parent { protected int a; protected Parent() { a = 0; } } </pre> Как характеризуется переменная a по отношению ко всем производным классам? 1) в производных классах переменная a является недоступной как внутри класса, так и из объектной переменной 2) в производных классах переменная a является недоступной внутри класса, но доступной из объектной переменной 3) в производных классах переменная a является видимой внутри класса, но	ПК-1

		недоступной из объектной переменной 4) в производных классах переменная а является видимой как внутри класса, так и из объектной переменной	
15.	3	В языке C# программист определил классы и создал их объекты следующим образом) <pre> public class Parent { protected int a; protected Parent(int pA) { a = pA; } } public class Child : Parent { int b; public Child() :base(0) { b = a; } } static void Main(string[] args) { Parent ob1 = new Parent(); Child ob2 = new Child(); } </pre> Какая ошибка была допущена? 1) некорректное использование зарезервированного слова base 2) в конструкторе класса Child нет доступа к переменной а базового класса 3) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора 4) нельзя создать объект базового класса, так как объявлен класс-наследник 5) ошибок нет	ПК-1
16.	1	В языке C# программист определил классы и создал их объекты следующим образом)	ПК-1

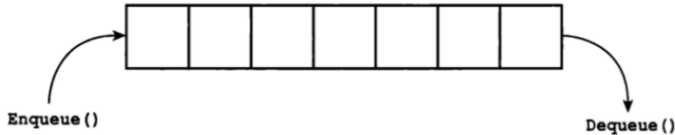
		<pre> public class Parent { protected int a; protected Parent(int pA) { a = pA; } public Parent() : this(0) { } } static void Main(string[] args) { Parent ob1 = new Parent(); Child ob2 = new Child(); } public class Child : Parent { int b; public Child() :base() { b = a; } } </pre> Какая ошибка была допущена? 1) ошибок нет 2) некорректное использование зарезервированного слова base 3) в конструкторе класса Child нет доступа к переменной a базового класса 4) некорректное использование зарезервированного слова this 5) нельзя создавать объект ob1, так как класс Parent не имеет открытого конструктора	
17.	1	В языке C# программист определил классы и интерфейсы следующим образом)	ПК-1

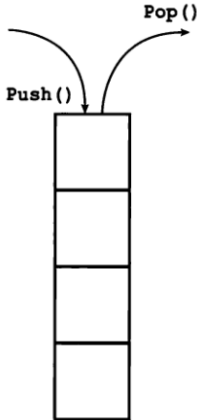
		<pre> public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } public void Draw() { Console.WriteLine(data.ToString()); } } </pre> Какая ошибка была допущена? 1) нет ошибок 2) класс MainObject не может иметь двух базовых классов 3) некорректное использование зарезервированного слова interface 4) в классе IString метод GetStr не определен	
18.	2	На языке C# программист определил класс, интерфейсы и создал объекты следующим образом) <pre> public interface IDraw { void Draw(); } public interface IString { string GetStr(); } public class Parent { protected int a; } </pre> <pre> public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } public void Draw() { Console.WriteLine(data.ToString()); } } </pre>	ПК-1

		<pre>static void Main(string[] args) { MainObject ob1 = new MainObject(); ob1.Draw(); Parent ob2 = new Parent(); ob2.Draw(); }</pre> Какая ошибка была допущена? 1) в классе Parent нет конструктора, поэтому объект ob2 не может быть создан 2) в классе Parent не определен метод Draw, который вызывается через объектную переменную ob2 3) класс MainObject не может иметь двух базовых классов 4) ошибок нет	
19.	2	На языке C# программист определил класс, интерфейсы и создал объекты следующим образом) <pre>public interface IDraw { void Draw(); } public interface IString { string GetStr(); } public class MainObject: IString, IDraw { int data; public string GetStr() { return "Данные: " + data.ToString(); } }</pre> Какая ошибка была допущена? 1) класс MainObject не может иметь двух базовых классов 2) класс MainObject не реализует метод Draw, наследуемого интерфейса IDraw 3) класс MainObject содержит метод GetStr, который дублирует метод базового класса IString 4) ошибок нет	ПК-1

20.	1	На языке C# программист определил класс, интерфейсы) <pre> public interface IString { string GetStr(); } public class Data1 : IString { int data; public string GetStr() { return "Данные: " + data.ToString(); } } </pre> <pre> public class Data2 : IString { string data; public string GetStr() { return data; } } </pre> Затем программист реализовал функцию Main) <pre> static void Main(string[] args) { IString[] mas = new IString[] { new Data1(), new Data2() }; } </pre> Какая ошибка была допущена? 1) нет ошибок 2) в массиве присутствуют элементы различного типа 3) не указан размер массива при его создании 4) тип массива, заявленный при объявлении, и тип добавляемых элементов не совпадают	ПК-1
21.	2	На языке C# программист определил класс, интерфейсы)	ПК-1

		<pre> public interface IString { string GetStr(); } public class Data1 : IString { int data; public string GetStr() { return "Данные: " + data.ToString(); } } </pre>	<pre> public class Data2 : IString { string data; public string GetStr() { return data; } } </pre>	
		Затем программист реализовал функцию Main) <pre> static void Main(string[] args) { IString[] mas = new IString[] { new Data1(), new Data2() }; IString ob = new IString(); } </pre> Какая ошибка была допущена? 1) нет ошибок 2) Невозможно создать объект интерфейса ob 3) в массиве присутствуют элементы различного типа 4) тип массива, заявленный при объявлении, и тип добавляемых элементов не совпадают		
22.	1	Для чего служит оператор break? 1) прерывает выполнение текущей итерации цикла и цикла в целом 2) прерывает выполнение текущей итерации цикла и переходит к следующей итерации 3) прерывает выполнение текущей итерации цикла и переходит к выполнению всего цикла с начала		ПК-1
23.	2	Для чего служит оператор continue?		ПК-1

		1) прерывает выполнение текущей итерации цикла и цикла в целом 2) прерывает выполнение текущей итерации цикла и переходит к следующей итерации цикла 3) прерывает выполнение текущей итерации цикла и переходит к выполнению всего цикла с начала	
24.	3	В переменной Sum по завершении выполнения фрагмента кода будет содержаться... <pre>long Sum = 0; int i; for (i = 1; i <= 1000000; i++) if (i % 3 == 0) Sum += i;</pre> 1) сумма кубов всех чисел от 1 до 1000000 2) 0 3) сумма всех чисел кратных 3 и находящихся в диапазоне от 1 до 1000000	ПК-1
		Задания открытого типа	
25.	очередь	Какая структура данных характеризуется поведением «первым элемент пришел – первым считывается»?	ПК-1
	стек	Какая структура данных характеризуется поведением «последним элемент пришел – первым считывается»?	ПК-1
26.	массив	Какая структура данных характеризуется произвольным доступом к элементам (чаще всего по индексу)?	ПК-1
27.	очередь	На рисунке представлена схема организации структуры данных.  Какому типу коллекции наиболее подходит данная схема?	ПК-1
28.	стек	На рисунке представлена схема организации структуры данных.	ПК-1

		 Какому типу коллекции наиболее подходит данная схема?	
29.	словарь	Для какого вида коллекции подходит следующее определение: сложная структура данных, позволяющая обеспечить доступ к элементам по ключу?	ПК-1
30.	множество	Для какого вида коллекции подходит следующее определение: коллекция, содержащая неупорядоченный набор различных элементов?	ПК-1
31.	List<T>	Какой класс коллекций можно использовать в качестве динамического массива?	ПК-1
32.	WPF	На рисунке представлен фрагмент кода. <pre> <Window x:Class="WpfApplication1.MainWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="MainWindow" Height="350" Width="525"> <Grid> </Grid> </Window> </pre> Какая технология используется для создания приложения?	ПК-1
33.	Windows Forms	На рисунке представлен фрагмент кода.	ПК-1

		<pre>private void InitializeComponent() { this.SuspendLayout(); // // Form1 // this.AutoScaleMode = new System.Drawing.SizeF(6F, 13F); this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font; this.ClientSize = new System.Drawing.Size(425, 317); this.Name = "Form1"; this.Text = "Form1"; this.ResumeLayout(false); }</pre> Какая технология используется для создания приложения?	
34.	GDI+	Какая технология применяется в Windows Forms для рисования на поверхности экранной формы?	
35.	System.Windows.Forms	В каком пространстве имен расположены классы Windows Forms?	
36.	System.Windows	В каком пространстве имен расположены классы WPF?	
37.	System.Windows.Controls	В каком пространстве имен расположены основные классы элементов управления при использовании технологии WPF?	
38.	System.Windows.Ink	В каком пространстве имен содержатся классы для работы с перьевым вводом?	
39.	System.Windows.Shapes	В каком пространстве имен располагаются классы для рисования двумерной графики: линий, прямоугольников, эллипсов, многоугольников и ломанных в системе WPF?	
40.	0	Какое минимальное значение принимает индексатор массива в C#?	
41.	матрица	Какие математические объекты наилучшим образом представляются двумерными массивами?	
42.	while	Какая конструкция называется оператором цикла с предусловием?	

43.	default	Какое зарезервированное слово не может присутствовать в приведенном фрагменте? <pre> int a = 0; if (a == 0) a = 1; else if (a == 1) a = 2; else if (a == 2) a = 3; default a = 0; </pre>	
44.	1	Сколько итераций цикла будет выполнено? <pre> int a = 1, S = 0; do { S += a; a++; } while (a < 0); </pre>	
45.	100	Сколько итераций цикла будет выполнено? <pre> int a = 0, S = 0; do { S += a; a++; continue; break; } while (a<100); </pre>	
46.	100	Сколько раз выполнится оператор continue?	

		<pre> int a = 0, S = 0; do { S += a; a++; continue; break; } while (a<100); </pre>	
47.	0	Сколько раз выполнится оператор break? <pre> int a = 0, S = 0; do { S += a; a++; continue; break; } while (a<100); </pre>	

2. Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала (контрольных точек), оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

3. Критерии оценивания компетенций заданий открытого типа

Оценка **«отлично»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ не содержит ошибок и решены все задачи. Компетенция ПК-1 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если являются верными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит незначительные ошибки, не влияющие на общий результат и не представлены отдельные задачи, не влияющие на общий результат. Компетенция ПК-1 освоена на среднем уровне

Оценка **«удовлетворительно»** выставляется студенту, если в основном верными являются результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; отчетный документ содержит ошибки, не позволяющие оценить качество разработки; представлены не все данные и задачи решены не в полном объеме. Компетенция ПК-1 освоена на минимальном уровне.

Оценка **«неудовлетворительно»** выставляется студенту, если являются неверными результаты демонстрации знаний об основных платформах, технологиях и инструментальных программно-аппаратных средствах реализации информационных систем, применения современных технологий для реализации информационных систем, выбора платформ для реализации информационных систем, анализа и выбора инструментальных программно-аппаратных средств для реализации информационных систем; представленный документ содержит много грубых ошибок, не представлены ключевые данные и задачи решены недостаточно полно. Компетенция ПК-1 не сформирована.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка **«отлично»** выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ПК-1 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ПК-1 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ПК-1 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ПК-1 не сформирована.