

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и компьютерных наук

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

*Программирование на основе классов*

Кафедра дискретной математики и информатики  
факультета математики и компьютерных наук

**Образовательная программа бакалавриата  
02.03.02 - Фундаментальная информатика и информационные  
технологии**

Направленность (профиль) программы:  
**Информатика и компьютерные науки**

Форма обучения:  
**очная**

Статус дисциплины: входит в часть, формируемую  
участниками образовательных отношений

Махачкала, 2022

Рабочая программа дисциплины «Программирование на основе классов» составлена в 2022 году в соответствии с требованиями ФГОС ВО - бакалавриат по направлению подготовки 02.03.02 - Фундаментальная информатика и информационные технологии.

Приказ № 808 Минобрнауки от 23.08.2017 г.

Разработчик(и): кафедра дискретной математики и информатики, Ханикалов Х.Б., старший преподаватель.

Рабочая программа дисциплины одобрена:

на заседании кафедры дискретной математики и информатики

от «28» февраля 2022 г., протокол № 6.

Зав. кафедрой  Магомедов А.М.  
(подпись)

на заседании Методической комиссии факультета математики и компьютерных наук от 24 марта 2022, протокол № 4.

Председатель  Ризаев М.К.  
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением ДГУ «31» марта 2022 г.

Начальник УМУ  Гасангаджиева А.Г.  
(подпись)

## Аннотация рабочей программы дисциплины

Дисциплина “Программирование на основе классов” входит в часть ОПОП, формируемую участниками образовательных отношений образовательной программы бакалавриата по направлению 02.03.02 - Фундаментальная информатика и информационные технологии.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с объектно-ориентированным программированием, созданием консольных и графических приложений.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных – ОПК-1, профессиональных – ПК-4.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: в форме контрольных работ в конце каждого модуля и итогового экзамена в конце семестра.

Объем дисциплины – 4 зачетных единицы, в том числе в академических часах по видам учебных занятий:

Семестр	Учебные занятия						Форма промежуточной аттестации	
	в том числе							
	Всего	Контактная работа обучающихся с преподавателем			СРС, в том числе экзамен			
		Всего	из них					
Лекции	Лабораторные занятия		Практические занятия					
3	144	72	36		36		72	экзамен

### 1. Цели освоения дисциплины

Целями освоения дисциплины «Программирование на основе классов» являются овладение знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования. Формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения.

Основные задачи дисциплины: овладение методами структурного и объектно-ориентированного программирования; закрепление навыков алгоритмизации и программирования, полученных в предыдущих семестрах; создание практической базы для написания качественной выпускной квалификационной работы.

Ожидаемые результаты:

- усвоение базовых типов и операторов;
- получить знания об особенностях объектно-ориентированного программирования, изучить иерархию классов и интерфейсов;
- освоить современные методы и технологии программирования;
- получить необходимые знания о протоколах передачи информации глобальных се-

- приобрести навыки и умения в постановке и решении задач разработки приложений на основе классов.

## 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина входит «Программирование на основе классов» в часть, формируемую участниками образовательных отношений ОПОП бакалавриата по направлению 02.03.02 - Фундаментальная информатика и информационные технологии и изучается в соответствии с графиком учебного процесса в 3 семестре. Изучение предмета производится в течение одного семестра и заканчивается экзаменом.

Дисциплина опирается на знания, полученные в процессе изучения дисциплин «Основы программирования», «Языки программирования». «Java-программирование», «Программная инженерия» и «Введение в Case-технологии»; знания, умения и навыки, полученные при изучении данной дисциплины, используются, закрепляются и развиваются при проведении учебной практики, выполнении курсовой и выпускной квалификационной работы.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения)

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенций	Планируемые результаты обучения	Процедура освоения
ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК 1.1. Знает основные положения и концепции в области математических и естественных наук, базовые теории коммуникации; знает основную терминологию.	<i>Знает:</i> теоретические основы базовых математических дисциплин (математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов), а также теоретической механики, физики. <i>Умеет:</i> решать задачи, связанные с исследованием свойств функций и их производных, с	Устный опрос. Конспектирование и проработка лекционного материала.

	<p>интегрированием, с изучением функциональных рядов, с дифференциальными уравнениями, с численным решением дифференциальных уравнений, с алгебраическими уравнениями и их системами.</p> <p><i>Владеет:</i> базовыми методами современного математического анализа по исследованию математических и естественнонаучных задач.</p>	
<p>ОПК 1.2. Умеет осуществлять первичный сбор и анализ материала, интерпретировать различные математические объекты.</p>	<p><i>Знает:</i> способы использования знаний в различных областях математики при решении конкретных задач в области математики и естественных наук.</p> <p><i>Умеет:</i> применять различные методы современного математического анализа по исследованию математических и естественнонаучных задач.</p> <p><i>Владеет:</i> навыками применения методов современного математического анализа при решении конкретных задач в области математики и естественных наук.</p>	Контрольная работа
<p>ОПК 1.3. Имеет практический опыт работы с решением стандартных математических задач и применяет его в профессиональной деятельности</p>	<p><i>Знает:</i> различные методы современного математического анализа по исследованию математических и естественнонаучных задач.</p> <p><i>Умеет:</i> корректно выбрать методы решения конкретной задачи в области математики и естественных наук.</p> <p><i>Владеет:</i> навыками выбора методов решения задач современного математического анализа.</p>	

ПК-4. Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии.	ПК-4.1. Знает современные языки программирования и методы параллельной обработки данных. Знаком с содержанием Единого Реестра Российских программ для электронных вычислительных машин и баз данных.	<p><i>Знает:</i> основы математического анализа и различные приложения дифференциального и; современные языки программирования интегрального исчисления в математических и естественных науках и современные информационные технологии.</p> <p><i>Умеет:</i> применять дифференциальное и интегральное исчисления для решения различных задач математических и естественных наук; составлять программы на современных языках программирования.</p> <p><i>Владеет:</i> базовыми методами дифференциального и интегрального исчислений; навыками программирования на современных языках.</p>	Контрольная работа
	<p><b>ПК-4.2.</b></p> <p><i>Умеет реализовывать численные методы решения прикладных задач в профессиональной сфере деятельности, пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии.</i></p>	<p><i>Знает:</i> области применения дифференциального и интегрального исчисления; различные языки программирования.</p> <p><i>Умеет:</i> решать задачи, связанные с исследованием свойств функций и их производных, с изучением функциональных рядов, с оценкой погрешности аппроксимации функций; применять различные языки программирования в численном анализе.</p> <p><i>Владеет:</i> методами дифференциального исчисления для исследования функций и навыками приложения интегрального исчисления к геометрии, физике.</p>	
	ПК-4.3. Имеет практический опыт разработки интеграции информационных систем.	<p><i>Знает:</i> методы исследования функций с помощью производных, вычисления интегралов; методы исследования сходимости рядов; численные методы анализа; современные информационные технологии.</p> <p><i>Умеет:</i> применять методы исследования функций с помощью производных, вычисления интегралов и методы исследования сходимости рядов в численном анализе с использова-</p>	

		нием современных информационных технологий. <i>Владеет:</i> навыками решения задач численного анализа с использованием методов дифференциального и интегрального исчисления.	
--	--	---	--

#### 4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 4 зачетных единицы, 144 академических часа: 36 ч. лекций, 36 ч. практ, 36 ч. СРС, 36 ч. контроль в виде экзамена.

4.2. Структура дисциплины.

4.2.1. Структура дисциплины в очной форме

№ п/п	Разделы и темы дисциплины по модулям	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Формы текущего контроля успеваемости и промежуточной аттестации
			Всего	Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа в т. ч зачет, экзамен	
<b>Модуль 1. Основы объектно-ориентированных языков программирования</b>								
1.	Тема 1. Введение в программирование на языке высокого уровня. Инструментальные средства разработки.	3	6	2	2		2	Устный опрос
2.	Тема 2. Базовые элементы языка. Типы данных.	3	6	2	2		2	Устный опрос
3.	Тема 3. Базовые элементы языка. Операторы.	3	12	4	4		4	Устный опрос
4.	Интегрированная		12	4	4		4	Контрольная

	среда разработки программ. Основные элементы среды разработки.							работа №1
	Итого по модулю 1		36	12	12		12	
<b>Модуль 2. Введение в ООП</b>								
5.	Тема 1. Принципы ООП.	3	6	2	2		2	Устный опрос
6.	Тема 2. Классы и объекты.		18	6	6		6	Устный опрос
7.	Тема 3. Работа со структурированными типами	3	6	2	2		2	Устный опрос
8.	Тема 4. События. Обработка событий от компонент.		6	2	2		2	Контрольная работа №2
	Итого по модулю 2		36	12	12		12	
<b>Модуль 3. Использование визуальных компонентов</b>								
9.	Тема 1. Графические примитивы. Работа с графикой.	3	12	4	4		4	Устный опрос
10.	Тема 2. Текстовые компоненты Label, Edit, Memo. Списки List Box, Combo Box.	3	12	4	4		4	Устный опрос
11.	Тема 3. Представление и обработка табличных данных. Класс TStringList.		6	2	2		2	Устный опрос
12.	Тема 4. Работа с файлами. Компоненты для работы с файлами.		6	2	2		2	Контрольная работа №3
	Итого по модулю 3		36	12	12		12	
	Модуль 4. Подготовка к экзамену		36				36	Экзамен
	<b>ИТОГО:</b>		144	36	36		72	



### **4.3. Содержание дисциплины, структурированное по разделам и темам**

#### 4.3.1. Содержание лекционных занятий по дисциплине, структурированное по разделам и темам

##### **Модуль 1. Основы объектно-ориентированных языков.**

##### **Тема 1. Введение в ООП. Инструментальные средства разработки.**

Языки программирования высокого уровня.

Визуальная разработка программ в Delphi.

Пакетные средства, интегрированные средства разработки.

##### **Тема 2. Базовые элементы языка. Типы.**

Структура программы.

Типы данных. Идентификаторы, константы, переменные.

Преобразование типов.

Массивы.

##### **Тема 3. Базовые элементы языка. Операторы.**

Операторы выбора, цикла, перехода.

Операторы и блоки. Оператор if-else. Оператор switch. Цикл while и do-while. Оператор for. Метки. Оператор break. Оператор continue. Оператор return.

Использование функций.

##### **Тема 4. Интегрированная среда разработки программ.**

Главное окно интегрированной среды разработки Delphi.

Основные элементы интегрированной среды Delphi.

Проект. Структура проекта.

Конструирование простого приложения.

##### **Модуль 2. Основы объектно-ориентированного программирования.**

##### **Тема 1. Принципы ООП.**

Основные понятия.

Инкапсуляция.

Наследование.

Полиморфизм.

##### **Тема 2. Классы и объекты.**

Классы, интерфейсы, абстрактные классы.

Реализация классов.

Создание объекта, оператор new.

Правила доступа к данным и методам объекта.

Переопределение методов.

Метод-конструктор.

##### **Тема 3. Работа со структурированными типами**

Классы String, String Buffer.

Равенство строк. Сравнение. Упорядочение.

Методы работы со строками.

Методы indexOf, lastIndexOf.

Типы исключений.

Вложенные операторы try.

Блок finally. Операторы throw, catch. Подклассы Exception.

##### **Тема 4. События. Обработка событий от компонент.**

Обработка событий.

Метод HandleEvent.

Класс MouseEventArgs и интерфейс MouseListener.

Движение мыши, перетаскивание.

Событие от клавиатуры.

Обработка событий, инициированных компонентами.

### **Модуль 3. Использование визуальных компонентов.**

#### **Тема 1. Графические примитивы. Работа с графикой.**

Загрузка и рисование растрового изображения.

Класс Image. Метод paint.

Координаты, цвет, шрифт.

Графические элементы. Класс Graphics. Метод setColor.

Просмотр изображения в апплете.

Видео в окне апплета.

Загрузка и проигрывание звуковых файлов.

#### **Тема 2. Компоненты вкладки Standard.**

Текстовые компоненты Label, Edit, Memo.

Списки List Box, Combo Box.

Ввод / вывод массива.

#### **Тема 3. Представление и обработка табличных данных.**

Класс TStringList.

Компонент String Grid – таблица строк.

Ввод / вывод массива.

Переключатели.

#### **Тема 4. Работа с файлами.**

Файлы. Основные понятия и операции.

Типизированные файлы.

Текстовые файлы.

Процедуры работы с файлами.

Компоненты для работы с файлами.

4.3.2. Содержание практических занятий по дисциплине, структурированное по разделам и темам

### **Модуль 1. Основы объектно-ориентированных языков.**

#### **Тема 1. Введение в ООП. Инструментальные средства разработки.**

Понятие байт-кода. Java-машина. Кроссплатформенные программы.

Пакетные средства, интегрированные средства разработки.

#### **Тема 2. Базовые элементы языка. Типы.**

Структура программы, простые типы, формат их представления в памяти.

Преобразование типов. Особенности массивных типов.

#### **Тема 3. Базовые элементы языка. Операторы.**

Условный оператор и его развитие – оператор выбора. Различия и взаимозаменяемость операторов цикла. Минусы оператора перехода. Операторы прерывания, продолжения, возврата. Использование функций.

### **Модуль 2. Основы объектно-ориентированного программирования.**

#### **Тема 1. Принципы ООП.**

Три основных принципа ООП. Как инкапсуляция разграничивает права доступа. Применение идеи наследования. Примеры полиморфизма.

#### **Тема 2. Классы и объекты.**

Классы, интерфейсы, абстрактные классы. Реализация классов и создание объекта. Правила доступа к данным и методам объекта. Переопределение методов. Конструктор.

#### **Тема 3. Работа со структурированными типами**

Классы String, StringBuffer. Равенство строк. Сравнение. Упорядочение.

Методы работы со строками. Методы indexOf, lastIndexOf.

Типы исключений. Вложенные операторы try. Блок finally. Операторы throw, catch. Подклассы прерываний.

**Тема 4.** События. Обработка событий от компонент.

Обработка событий. Метод handleEvent. Класс MouseEvent и интерфейс MouseListener. События от мыши и клавиатуры. Обработка событий, инициированных компонентами.

### **Модуль 3. Использование визуальных компонентов.**

**Тема 1.** Графические примитивы. Работа с графикой.

Загрузка и отображения растрового изображения. Класс Image. Метод paint.

Координаты, цвет, шрифт. Графические элементы. Класс Graphics. Метод setColor. Просмотр изображения в апплете. Видео в окне апплета.

Загрузка и проигрывание звуковых файлов.

**Тема 2.** Компоненты вкладки Standard.

Текстовые компоненты Label, Edit, Memo.

Списки List Box, Combo Box.

Ввод / вывод массива.

**Тема 3.** Представление и обработка табличных данных.

Класс TStringList.

Компонент StringGrid – таблица строк.

Ввод / вывод массива.

Переключатели.

**Тема 4.** Работа с файлами.

Файлы. Основные понятия и операции.

Типизированные файлы.

Текстовые файлы.

Процедуры работы с файлами.

Компоненты для работы с файлами.

### **5. Образовательные технологии**

Процесс изложения учебного материала сопровождается систематическими (на каждом занятии) компьютерными презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного оборудования, предусмотрено широкое использование в учебном процессе активных и интерактивных форм проведения занятий.

Студентам регулярно сообщаются те из формулировок задач, предложенных на собеседовании в ведущие IT-компании (Microsoft, Apple, Google, Uber, Amazon.com и др.), которые соответствуют текущим темам.

. Предусмотрено регулярное общение и консультации с представителями российских и зарубежных компаний (из числа выпускников кафедры) по электронной почте и по скайпу.

### **6. Учебно-методическое обеспечение самостоятельной работы студентов**

Для обеспечения самостоятельной работы используется разработанный на кафедре пакет заданий и методических указаний. Самостоятельная работа студентов складывается из проработки лекционного материала и материалов форумов программирования, материала учебника, решения всех заданий из индивидуальных заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных форм контроля.

Пример задания на самостоятельную работу.

**Задание**

Выведите на экран все двузначные члены последовательности  $2a_{n-1}+50$ , где  $a_1 = -26$ .  
Создайте массив из 11 случайных целых чисел из отрезка  $[-1;1]$ , выведите массив на экран в строку. Определите какой элемент встречается в массиве чаще всего и выведите об этом сообщение на экран. Если два каких-то элемента встречаются одинаковое количество раз, то не выводите ничего.

Создайте класс углов, отложенных против часовой стрелки от положительного направления оси абсцисс, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий задают ли углы перпендикулярные прямые.

С использованием построенного класса создайте угол в  $10^\circ$  и второй угол в  $280^\circ$ . Проверьте с помощью созданного метода задают ли углы перпендикулярные прямые и если да, то выведите соответствующее сообщение на экран.

## **7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины**

### **7.1. Типовые контрольные задания**

#### **Вариант 1**

1. В переменной  $n$  хранится четырехзначное число. Создайте программу, вычисляющую и выводящую на экран сумму цифр  $n$ .
2. Создайте массив из 15 случайных целых чисел из отрезка  $[0;9]$ . Выведите массив на экран. Подсчитайте сколько в массиве чётных элементов и выведите это количество на экран на отдельной строке.
3. Создайте класс окружностей на плоскости, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий имеют ли две окружности равную площадь.

С использованием построенного класса создайте две окружности: одну с центром в  $(0;0)$  и радиусом 12, а вторую с центром в  $(3;5)$  и радиусом 11. Проверьте с помощью созданного метода равна ли их площадь и если равна, то выведите соответствующее сообщение на экран.

#### **Вариант 2**

1. В переменных  $q$  и  $w$  хранятся два натуральных числа. Создайте программу, выводящую на экран результат деления  $q$  на  $w$  с остатком. Пример вывода программы (для случая, когда в  $q$  хранится 21, а в  $w$  хранится 8):  $21 / 8 = 2$  и 5 в остатке.
2. Создайте массив из 8 случайных целых чисел из отрезка  $[1;10]$ . Выведите массив на экран в строку. Замените каждый элемент с нечётным индексом на ноль. Снова выведите массив на экран на отдельной строке.
3. Создайте класс отрезков на координатной плоскости, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий равна ли длина двух отрезков.

С использованием построенного класса создайте два отрезка: один от точки (1;1) до точки (2;2) и второй отрезок от точки (-3;0) до точки (1;1). Проверьте с помощью созданного метода равна ли их длина и если равна, то выведите соответствующее сообщение на экран.

### Вариант 3

1. Создать программу, которая будет проверять попало ли случайно выбранное из отрезка [5;155] целое число в интервал (25;100) и сообщать результат на экран.
2. Создайте массив из 8 случайных целых чисел из отрезка [1;10]. Выведите массив на экран в строку. Замените каждый элемент с нечётным индексом на ноль. Снова выведете массив на экран на отдельной строке.
3. Создайте класс прямоугольных треугольников, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, проверяющий являются ли два треугольника подобными.

С использованием построенного класса создайте два треугольника: один с катетами 3 и 4 и второй с катетами 5 и 6. Проверьте с помощью созданного метода подобны ли треугольники и если да, то выведите соответствующее сообщение на экран.

### Вариант 4

1. Создать программу, выводющую на экран случайно сгенерированное трёхзначное натуральное число и его наибольшую цифру.
2. Создайте массив из 4 случайных целых чисел из отрезка [10;99], выведите его на экран в строку. Определить и вывести на экран сообщение о том, является ли массив строго возрастающей последовательностью.
3. Создайте класс комплексных чисел, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных.

Опишите в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства. Если это необходимо, то проверьте допустимость их значений в конструкторе (например, в классе обыкновенных дробей нельзя создавать дробь с нулевым знаменателем).

Создайте в классе метод, вычисляющий модуль суммы двух комплексных чисел.

С использованием построенного класса создайте два комплексных числа:  $3i+1$  и  $2i-1$ . Вычислите с помощью метода и выведите на экран модуль их суммы.

### Вопросы

Вопросы экзамена по курсу «Программирование на основе классов» для студентов 2 курса ФМиКН направления ФИИТ, 3 семестр.

1. Принципы объектно-ориентированного программирования.
2. Понятие кроссплатформенности.
3. Идентификаторы. Комментарий. Управляющие символы.
4. Типы данных. Примитивные типы.
5. Типы данных. Ссылочные типы. Массивы.
6. Типы данных. Ссылочные типы. Классы. Интерфейсы.
7. Логические операции. Преобразование типа. Переменные.
8. Операторы. Логические, присваивания. Приоритеты.
9. Операторы. Условные, выбора.
10. Операторы цикла. Три инструкции перехода.
11. Свойства ООП.
12. Классы. Передача параметров по значению и по ссылке. Ссылка this.
13. Модификаторы доступа. Типы методов: перегруженные, переопределенные. Ключевое слово super.
14. Пакеты. Включение пакета, класса пакета. Пакет AbstractWindowToolkit, GUI.

15. Контейнеры: фрейм, панель.
16. Компоненты GUI: Label, TextField, TextArea.
17. Компоненты GUI: Button, Checkbox, радиокнопки, списки.
18. Типы макетов.
19. Апплеты - создание, жизненный цикл, вывод апплета.
20. Апплеты – вывод изображений, передача параметров.
21. Классы Graphics, Colors, Fonts.
22. Обработка исключений. Классы ErrorException.
23. Ключевые слова: try, catch, throw, throws, finally.
24. Потоки, способы создания потоков.
25. Состояния потоков, два типа потоков.
26. Пакеты, управление доступом.

## **7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Общий результат складывается из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий – 10 баллов,
- выполнение текущих практических заданий – 40 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 50 баллов,
- письменная контрольная работа - 50 баллов.

Промежуточный контроль по дисциплине включает экзамен, результаты которого оцениваются по 100-балльной системе ориентировочно по следующим критериям:

1) оценка «отлично», если у студента от 86 до 100 баллов с учетом степени усвоения, высокий уровень знаний по программе дисциплины, отвечает четко и логически обоснованно;

2) оценка «хорошо», если у студента от 66 до 85 баллов с учетом степени усвоения, достаточно высокий уровень знаний по программе дисциплины, отвечает в основном четко и логически обоснованно, но допускает отдельные неточности.

3) оценка «удовлетворительно», если у студента от 51 до 65 баллов с учетом степени усвоения, достаточный уровень знаний по программе дисциплины, отвечает в основном правильно и в логической последовательности, но допускает отдельные неточности;

4) оценка «неудовлетворительно», если у студента от 0 до 50 баллов с учетом степени усвоения, недостаточный уровень знаний по программе дисциплины, имеются существенные пробелы в усвоении важных знаний из программы курса.

## **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

Основная литература:

1. Фаронов В.В. Turbo Pascal 7.0. Учебный курс: учебное пособие / В.В. Фаронов. – СПб.: Питер, 2012. – 368 с.
2. Джошуа Блох Java. Эффективное программирование [Электронный ресурс]/ Джошуа Блох— Электрон. текстовые данные. — Саратов: Профобразование, 2017.— 310 с.— Режим доступа: <http://www.iprbookshop.ru/64057.html>.— ЭБС «IPRbooks».
3. Вязовик Н.А. Программирование на Java [Электронный ресурс]/ Вязовик Н.А.— Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 603 с.— Режим доступа: <http://www.iprbookshop.ru/73710.html>. — ЭБС «IPRbooks»3.
4. Эндрю Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5 (6-е издание). Издательство: Вильямс, 2013.

**Дополнительная литература:**

1. Мухаметзянов Р.Р. Основы программирования на Java [Электронный ресурс]: учебное пособие/ Мухаметзянов Р.Р.— Электрон. текстовые данные. — Набережные Челны: Набережночелнинский государственный педагогический университет, 2017. — 114 с.— Режим доступа: <http://www.iprbookshop.ru/66812.html>.— ЭБС «IPRbooks»
2. В. Пестриков, А. Маслобоев. Delphi на примерах. СПб.: БХВ-Петербург, 2005.

## **9. Перечень рекомендуемых ресурсов информационно-телекоммуникационной сети «Интернет»**

1. <http://www.intuit.ru/> - Интернет-университет информационных технологий
2. <http://www.delphimaster.ru/> - сайт, посвященный программированию в среде Delphi.
3. <http://delphiexpert.ru/> - уроки, видеокурсы по программированию в среде Delphi.
4. <http://sources.ru/> - большой набор статей, исходников по программированию, администрированию и IT-дизайну
5. <http://codeby.net/> - форум разработчиков прикладного программного обеспечения.
6. <http://www.ru-coding.com/> - статьи и книги по программированию сетевых приложений и приложений под Windows.

## **10. Методические указания для обучающихся по освоению дисциплины**

- 1) Помимо выполнения заданий на практических занятиях рекомендуется самостоятельно решить упражнения, предложенных к каждой лекции.
- 2) Самостоятельная работа студентов заключается в решении всех разобранных на занятиях упражнений, материала учебника и соответствующих форумов интернет, решения всех заданий из индивидуальных практических заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных отчетов и экзамена и дополнительной работы в компьютерном классе самостоятельно.

## **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

Для проведения полноценных занятий необходимо следующее программное обеспечение: Операционная система Windows 7, 8.1 и 10, JDK, Microsoft Visual Studio Express, NetBeans, Ubuntu Linux.

## **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Лекции по дисциплине читаются в классе, оборудованном проектором, к каждому занятию имеются презентации, лабораторные работы проходят в компьютерном классе, оборудованном необходимым аппаратными и программными средствами. Часть лекций предоставляется студенту в электронном формате. Практические занятия проводятся в компьютерных классах с современным аппаратным и программным обеспечением. На сайте кафедры размещаются учебные пособия и презентации к лекции.