

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Java – программирование

Кафедра дискретной математики и информатики
факультета математики и компьютерных наук

**Образовательная программа бакалавриата
02.03.01 – Математика и компьютерные науки**

Направленность (профиль) программы:
Математический анализ и приложения

Форма обучения
очная

Статус дисциплины:
дисциплина по выбору

Махачкала, 2022

Рабочая программа дисциплины «Java – программирование» составлена в 2022 году в соответствии с требованиями ФГОС ВО – бакалавриат по направлению подготовки 02.03.01 – Математика и компьютерные науки от 23.08.2017 г. № 807

Разработчик: кафедра дискретной математики и информатики, Якубов А.З., к. ф.-м. н., доцент.

Рабочая программа дисциплины одобрена:
на заседании кафедры дискретной математики и информатики

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

Зав. кафедрой МВ Магомедов А.М.

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

Председатель М.К. Ризаев М.К.

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

Начальник УМУ А.Г. Гасангаджиева А.Г.

Аннотация рабочей программы дисциплины

Дисциплина “Java-программирование” является дисциплиной по выбору по направлению подготовки 02.03.01 – Математика и компьютерные науки.

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

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

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных - ОПК-4, ОПК-5, ОПК-6, профессиональных - ПК-1.

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

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости: в форме 2-х контрольных работ в конце каждого модуля и итогового экзамена в конце семестра.

Объем дисциплины – 3 зачетные единицы, в том числе в академических часах по видам учебных занятий:

Се- местр	Учебные занятия						СРС	Форма промежуточной аттестации
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Все го	из них						
Лек- ции		Лабораторные занятия	Практические занятия	КСР	консультации			
8	108	12		24			36	36 (экзамен)

1. Цели освоения дисциплины

Целями освоения дисциплины Java-программирование 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 Java;
- изучить базовые элементы языка Java: типы, операторы, иерархию классов и интерфейсов;
- получить знания об особенностях объектно-ориентированного программирования на Java;
- изучить средства и методы, предоставляемые языком Java для Web-программирования;
- освоить современные методы программирования апплетов, сервлетов, способы их взаимодействия в сети;
- получить необходимые знания о протоколах передачи информации глобальных сетей;
- приобрести навыки и умения в постановке и решении задач разработки динамических Web-страниц.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина входит в вариативную по выбору часть образовательной программы бакалавриата по направлению 02.03.01 – Математика и компьютерные науки и изучается в соответствии с графиком учебного процесса в восьмом семестре. Изучение предмета производится в течение одного семестра и заканчивается экзаменом.

Дисциплина опирается на знания, полученные в 1,2,3 семестре в процессе изучения дисциплин «Технологии программирования и работа на ЭВМ». В свою очередь, на знания и навыки полученные при изучении данной дисциплины, используются, закрепляются и развиваются при проведении преддипломной практики, выполнении 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-4. Способен находить, анализировать, реализовывать программно и	ОПК-4.1. Знает базовые основы современного математического аппарата, связанного с проектированием,	<i>Знает: общие вопросы теории интеллектуальных систем, различные методы обработки информации, способы их программной реализации.</i>	Конспектирование и проработка теоретического материала. Участие в

<p>использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем.</p>	<p>разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности.</p>	<p><i>Умеет: применять методы машинного обучения в задачах обработки информации, распознавания образов и в других областях человеческой деятельности.</i> <i>Владеет: основными разделами и важнейшими методами обработки информации для возможности их применения при решении научных и научнообразовательных задач.</i></p>	<p>практических занятиях. Реализация проектов. Самостоятельная работа.</p>
	<p>ОПК-4.2. Умеет использовать этот математический аппарат в профессиональной деятельности.</p>	<p><i>Знает: теоретические основы использования информационных технологий в науке и образовании; основные методы работы с ресурсами сети Интернет.</i> <i>Умеет: применять современные методы и средства автоматизированного анализа и систематизации научных данных; практически использовать научно-образовательные ресурсы Интернет в повседневной профессиональной деятельности исследователя и педагога.</i> <i>Владеет: навыками использования информационных технологий в организации и проведении научного исследования; навыками использования современных баз данных; навыками</i></p>	

		<p><i>применения мультимедийных технологий обработки и представления информации;</i></p> <p><i>навыками автоматизации подготовки документов в различных текстовых и графических редакторах.</i></p>	
	<p>ОПК-4.3.Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программ</p> <p>Знает: основные направления использования информационных технологий в научных исследованиях и в образовании; методики и технологии проведения обучения с использованием информационных технологий.</p> <p>Умеет: использовать современные информационные технологии для подготовки традиционных и электронных учебных продуктов и программных комплексов в различных областях человеческой деятельности.</p>	<p><i>Знает: основные направления использования информационных технологий в научных исследованиях и в образовании; методики и технологии проведения обучения с использованием информационных технологий.</i></p> <p><i>Умеет: использовать современные информационные технологии для подготовки традиционных и электронных учебно-методических и научных публикаций.</i></p> <p><i>Владеет: навыками получения научных доказательств и проведения научноисследовательских работ с использованием компьютерного моделирования.</i></p>	
<p>ОПК-5. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности</p>	<p>ОПК-5.1.Знает основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и</p>	<p><i>Знает: основные принципы документационного обеспечения профессиональной деятельности с учетом требований информационной безопасности; алгоритмы решения стандартных организационных задач; основные понятия, теоретические положения и методы программирования на языках</i></p>	<p>Конспектирование и проработка теоретического материала. Участие в практических занятиях. Реализация. Самостоятельная работа.</p>

	<p>эксплуатации программных продуктов и программных комплексов.</p>	<p><i>высокого уровня.</i> <i>Умеет: применять методы программирования при решении разнообразных задач</i> <i>теоретического и практического содержания.</i> <i>Владеет: методами программирования на различных языках высокого уровня для решения теоретических и практических задач.</i></p>	
	<p>ОПК-5.2. Умеет использовать современные информационнокоммуникационные технологии в профессиональной деятельности.</p>	<p><i>Знает: основные направления применения информационно-коммуникационных технологий в науке и образовании; принципы построения сетей; локальные и глобальные сети; сеть Интернет; безопасность компьютерных сетей.</i> <i>Умеет: выбирать эффективные информационные технологии для использования в научных исследованиях и учебном процессе.</i> <i>Владеет: методами математического и алгоритмического моделирования и информационно-коммуникационных технологий в науке и образовании.</i></p>	
	<p>ОПК-5.3. Имеет практические навыки разработки программного обеспечения.</p>	<p><i>Знает: теоретические положения и методы программирования на языках высокого уровня.</i> <i>Умеет: выбирать эффективные информационные технологии для использования в научных исследованиях и учебном процессе.</i> <i>Владеет: навыками построения алгоритмов и программ различных явлений и процессов, навыками использования информаци-</i></p>	

		<i>онных технологий для обработки данных.</i>	
ОПК-6. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1. Знает методы теории алгоритмов, методы системного и прикладного программирования.	<i>Знает: алгоритмы решения стандартных задач математического моделирования; основные понятия, теоретические положения и методы программирования на языках высокого уровня. Умеет: анализировать типовые языки программирования, составлять алгоритмы и компьютерные программы. Владеет: навыками решения задач анализа и интеграции различных типов алгоритмов и компьютерных программ.</i>	Конспектирование и проработка теоретического материала. Участие в практических занятиях. Реализация. Самостоятельная работа.
	ОПК-6.2. Умеет: разрабатывать алгоритмы и компьютерные программы в области математических и информационных моделей, создавать информационные ресурсы глобальных сетей.	<i>Знает: основные направления применения компьютерных программ в области математических и информационных моделей; принципы построения локальных и глобальных сетей. Умеет: выбирать эффективные алгоритмы и компьютерные программы для практического применения. Владеет: навыками разработки алгоритмов и компьютерных программ с применением информационных ресурсов глобальных сетей.</i>	
	ОПК-6.3. Имеет практический опыт разработки алгоритмов и компьютерных программ для практического применения.	<i>Знает: теоретические положения теории алгоритмов и методы программирования на языках высокого уровня. Умеет: путем достаточно</i>	

		<p><i>глубокого анализа выбирать эффективные алгоритмы и компьютерные программы для практического применения.</i></p> <p><i>Владеет: методами построения алгоритмов и программ различных явлений и процессов, навыками использования информационных технологий для обработки данных для практического применения.</i></p>	
ПК-1. Способен демонстрировать базовые знания математических и естественных наук, основ программирования и информационных технологий	ПК-1.1. Обладает базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий.	<p><i>Знает: основы математического анализа и различные приложения дифференциального и интегрального исчисления в математических и естественных науках; современные языки программирования и современные информационные технологии.</i></p> <p><i>Умеет: применять дифференциальное и интегральное исчисления для решения различных задач математических и естественных наук; составлять программы на современных языках программирования.</i></p> <p><i>Владеет: базовыми методами дифференциального и интегрального исчислений; навыками программирования на современных языках.</i></p>	Конспектирование и проработка теоретического материала. Участие в практических занятиях. Реализация. Самостоятельная работа.
	ПК-1.2. Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и	<p><i>Знает: области применения дифференциального и интегрального исчисления; различные языки программирования.</i></p> <p><i>Умеет: решать задачи, связанные: с</i></p>	

	информатике.	<p><i>исследованием свойств функций и их производных, с изучением функциональных рядов, с оценкой погрешности аппроксимации функций; применять различные языки программирования в численном анализе.</i></p> <p><i>Владеет: методами дифференциального исчисления для исследования функций и навыками приложения интегрального исчисления к геометрии, физике.</i></p>	
	ПК-1.3. Имеет практический опыт научно-исследовательской деятельности в математике и информатике.	<p><i>Знает: методы исследования функций с помощью производных, вычисления интегралов; методы исследования сходимости рядов; численные методы анализа; современные информационные технологии.</i></p> <p><i>Умеет: применять методы исследования функций с помощью производных, вычисления интегралов и методы исследования сходимости рядов в численном анализе с использованием современных информационных технологий.</i></p> <p><i>Владеет: навыками решения задач численного анализа с использованием методов дифференциального и интегрального исчисления.</i></p>	

4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 3 зачетные единицы, 108 академических часов: 12ч. лекций, 24 ч. практических, 36ч. – СРС, 36 ч. экзамен.

4.2. Структура дисциплины

Структура и содержание дисциплины «Java-программирование»

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Лаборат.зан.	Практические занятия	Контроль самост. раб.		
Модуль 1. Основы языка Java.									
1	Тема 1_1. Введение в программирование на Java. Инструментальные средства разработки Java-программ.	8	1	1		2		2	Текущий контроль - тест 1.1
2	Тема 2_1. Базовые элементы языка. Типы.		2	1		2		2	Текущий контроль - тест 2.1
3	Тема 3_1. Базовые элементы языка. Операторы.		3	1		2		4	Текущий контроль - тест 3.1
4	Тема 4_1. Основы объектно-ориентированного программирования для Java. Классы и объекты.		4	1		4		4	Текущий контроль - тест 4.1
5	Тема 5_1. Работа со строками.		5	1		2		2	Текущий контроль - тест 5.1
6	Тема 6_1. Обработка исключений на Java.		6	1		2		2	Текущий контроль - тест 6.1
	<i>Итого по модулю 1:</i>		1-6	6		14		16	Контрольная работа №1
Модуль 2. GUI-программирование и апплеты									
1	Тема 1_2. Создание апплета.	8	7	1		2		2	Текущий контроль - тест 1.2

2	Тема 2_2. Компиляция и выполнение апплета. Передача параметров в апплет.		8	1		2		4	Текущий контроль - тест 2.2
3	Тема 3_2. Средства пользовательского интерфейса. Компоненты и контейнеры.		9	1		2		4	Текущий контроль - тест 3.2
4	Тема 4_2. События. Обработка событий от компонент.		10-12	1		2		6	Текущий контроль - тест 4.2
5	Тема 5_2. Растровые изображения и анимация в апплетах. Звук в апплетах Java.		13-14	2		2		4	Текущий контроль - тест 5.2
	<i>Итого по модулю 2:</i>		10-12	6		10		20	Коллоквиум
	<i>Модуль 3:</i>								36 ч. экзамен
	ИТОГО:			12		24		36	36 ч. Экзамен

4.3. Содержание дисциплины, структурированное по темам

4.3.1. Содержание лекционных занятий по дисциплине

Модуль 1. Основы языка Java

Тема 1_1. Введение в программирование на Java. Инструментальные средства разработки Java-программ.

JDK – инструментальные средства разработки Java-программ.

Документация, литература.

Понятие байт-кода.

Платформено-независимость Java-приложений.

Мобильность Java.

Пакетные средства, интегрированные средства разработки.

Лабораторная работа.

Текущий контроль - тест 1.1

Тема 2_1. Базовые элементы языка. Типы.

Структура программы.

Типы данных.

Идентификаторы, константы, переменные.

Преобразование типов.

Массивы.

Лабораторная работа.

Текущий контроль - тест 2.1

Тема 3_1. 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.

Лабораторная работа.

Текущий контроль - тест 3.1

Тема 4_1. Основы объектно-ориентированного программирования для Java.

Классы и объекты.

Основные понятия.

Наследование.

Классы, интерфейсы, абстрактные классы.

Реализация классов.

Использование полиморфизма.

Создание объекта, оператор new.

Правила доступа к данным и методам объекта.

Переопределение методов.

Метод-конструктор.

Лабораторная работа.

Текущий контроль - тест1.2

Тема 5_1. Работа со строками.

Классы String, StringBuffer.

Доступ к символам.

Равенство строк. Сравнение.

Упорядочение.

Методы работы со строками.

Методы indexOf, lastIndexOf.

Лабораторная работа.

Текущий контроль - тест2.2

Тема 6_1. Обработка исключений на Java.

Типы исключений.

Вложенные операторы try.

Блок finally.

Операторы throw, catch.

Подклассы Exception.

Лабораторная работа.

Текущий контроль - тест3.2

Модуль 2. GUI-программирование и апплеты.

Тема 1_2. Создание апплета.

Создание апплета на Java.

Параметры апплета.

Исходный текст апплета.

Методы init, destroy, start, stop.

Лабораторная работа.

Текущий контроль - тест1.3

Тема 2_2. Компиляция и выполнение апплета. Передача параметров в апплет.

Компиляция и выполнение апплета.

Выполнение апплета в отдельном потоке.

Параметры апплета.

Компоненты в окне апплета.

Лабораторная работа.

Текущий контроль - тест2.3

Тема 3_2. Средства пользовательского интерфейса. Компоненты и контейнеры.

Кнопки, переключатели.

Списки класса Choice.

Списки класса List.

Текстовое поле классов Label, TextField.

Многострочное текстовое поле класса TextArea.

Лабораторная работа.

Текущий контроль - тест3.3

Тема 4_2. События. Обработка событий от компонент.

Обработка событий.

Метод handleEvent.

Класс MouseEvent и интерфейс MouseListener.

Движение мыши, перетаскивание.

Событие от клавиатуры.

Обработка событий, инициированных компонентами.

Лабораторная работа.

Текущий контроль - тест4.3

Тема 5_2. Растровые изображения и анимация в апплетах. Звук в апплетах Java.

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.

Лабораторная работа.

Текущий контроль - тест5.3

4.3.2. Содержание практических занятий по дисциплине

Введение в программирование на Java. Инструментальные средства разработки Java-программ.

1. JDK – инструментальные средства разработки Java-программ.

2. В чем заключается платформу-независимость Java-приложений?
3. Что представляет собой исходный файл на языке Java?
4. Написать программу выводящую на экран строку HelloWorld.

Базовые элементы языка. Типы, операторы.

1. Описать основные стандартные типы данных в языке Java.
2. Описать операторы выбора, цикла.
3. Написать программу, выводящую на экран месяцы года, время года и количество дней в каждом из них, используя, оператор цикла.

Основы объектно-ориентированного программирования для Java.

Классы и объекты.

1. Описать реализацию класса.
2. Описать оператор new.
3. Объявление методов.
4. Написать программу использующую разную реализацию методов в классе.

Работа со строками.

1. Описание класса String.
2. Описание класса StringBuffer. Примеры программ.

Обработка исключений на Java.

1. Описать типы исключений.
2. Описать работу операторов try, throw, catch, finally.
3. Написать программу использующую операторы исключения для выхода из метода.

Создание Java-приложения. Компиляция и выполнение апплета. Передача параметров в апплет.

1. Описать строение апплета.
2. Компиляция и выполнение апплета на Java.
3. Создать апплет с получением параметров.

Средства пользовательского интерфейса. Компоненты и контейнеры.

1. Вспомнить определения компонентов и контейнеров.
2. Написать апплеты, демонстрирующие примеры работы с классами Label, Button, Choice, List, TextField, TextArea.

События. Обработка событий от компонентов.

1. Описать процесс обработки событий от компонентов.
2. Написать программу демонстрирующую работу обработчиков событий.

5. Образовательные технологии

Для эффективной реализации целей и задач ФГОС, для претворения компетентностного подхода в преподавании дисциплины «Java программирование», используются следующие образовательные технологии и методы обучения:

Вид занятия	Технология	Цель	Формы и методы обучения
1	2	3	4

Лекции	Технология проблемного обучения	Усвоение теоретических знаний, развитие мышления, формирование профессионального интереса к будущей деятельности	Мультимедийные лекции-объяснение, лекция-визуализация, с привлечением формы тематической дискуссии, беседы, анализа конкретных ситуаций
Практические занятия	Технология проблемного, модульного, дифференцированного и активного обучения, деловая игра	Развитие творческой и познавательной самостоятельности, обеспечение индивидуального подхода с учетом базовой подготовки. Организация активности студентов, обеспечение лично-деятельного характера усвоения знаний, приобретения навыков, умений.	Индивидуальный темп обучения. Инновационные интерактивные методы в обучении: использование Web-ресурсов для подготовки компьютерных презентаций, использование off-line (электронная почта) для обмена информацией, консультаций с преподавателем, работа с электронными пособиями, возможность самотестирования. Постановка проблемных познавательных задач. Методы активного обучения: «круглый стол», игровое производственное
Самостоятельная работа	Технологии концентрированного, модульного, дифференцированного обучения	Развитие познавательной самостоятельности, обеспечение гибкости обучения, развитие навыков работы с различными источниками информации, развитие умений, творческих навыков	Индивидуальные, групповые, интерактивные (в режимах on-line и offline).

6. Учебно-методическое обеспечение самостоятельной работы студентов

6.1. Примерное распределение времени самостоятельной работы студентов

Раздел (модуль, тема)	Вид самостоятельной работы - практическое содержание	Контрольные сроки (в нед.) и вид контроля	Уч.-мет. обеспечение (указаны источники из списка основной литературы)
1	2	3	4
Модуль 1. Ос-	Особенности языка и платформы Java.	3 и 9 недели	[1] – [5];

новы языка Java.	Классификация программ по типу исполнения (компилируемые, интерпретируемые, исполняемые на виртуальных машинах). Виртуальная машина Java. JIT-компиляция. Создание простейшей программы на Java, её компиляция в байт-код и запуск. Средства разработки Java-приложений. Интегрированные среды разработки. Встроенные типы данных. Способы задания литералов различных типов. Хранение данных в памяти ЭВМ. Приведение типов (явное и автоматическое). Константы и переменные. Оператор присваивания. Порядок действий (приоритет операторов). Арифметические операторы. Операторы инкремента и декремента. Встроенный класс Math. Псевдослучайные числа. Операторы сравнения и логические операторы. Операторы ветвления. Условный оператор. Минимизация количества проверок. Операторы ветвления. Оператор множественного выбора. Его сравнение с условным оператором.	обучения. Проверка теоретических знаний на устном опросе и коллоквиуме. Проверка решенных задач.	материалы сайтов: https://ru.wikipedia.org/wiki/Java http://citforum.ru/
Модуль 2. GUI-программирование и апплеты.	Основные понятия ООП. Объекты и классы. Абстракция данных. Сценарий построения объектно-ориентированной программы. Члены классов. Методы и поля. Специальные методы классов (конструкторы). Конструктор по умолчанию. Модификаторы уровня доступа (default, public, protected, private). Основопологающие принципы ООП. Инкапсуляция. Основопологающие принципы ООП. Наследование. Управление наследованием.	10 и 15 недели обучения. Проверка теоретических знаний на устном опросе и коллоквиуме. Проверка выполнения компьютерных программ	[3], [4]; материалы сайтов: http://www.emanual.ru/

6.2. Виды и порядок выполнения самостоятельной работы

1. Изучение конспектов лекций и рекомендованной литературы.
2. Подготовка к опросу на практических занятиях
3. Решение задач и упражнений
4. Подготовка к коллоквиуму и контрольным работам
5. Поиск материала на интернет-форумах
6. Подготовка к экзамену

6.3. Порядок контроля:

1. Опрос на практическом занятии
2. Проверка выполнения домашних заданий и контрольных работ

3. Коллоквиум

4. Зачет.

6.4. Примеры индивидуальных вариантов задач для самостоятельного выполнения:

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.

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

7.1. Типовые контрольные задания

7.1.1. Темы рефератов:

Модуль 1. Основы языка Java

Тема 1_1. Введение в программирование на Java. Инструментальные средства разработки Java-программ.

Тема 2_1. Базовые элементы языка. Типы.

Тема 3_1. Базовые элементы языка. Операторы.

Тема 4_1. Основы объектно-ориентированного программирования для Java.

Тема 5_1. *Работа со строками.*

Тема 6_1. Обработка исключений на Java.

Модуль 2. GUI-программирование и апплеты.

Тема 2_2. Компиляция и выполнение апплета. Передача параметров в апплет.

Тема 3_2. Средства пользовательского интерфейса. Компоненты и контейнеры.

Тема 4_2. События. Обработка событий от компонент.

Тема 5_2. Растровые изображения и анимация в апплетах. Звук в апплетах Java.

7.1.2. Примерные упражнения и задания для текущего контроля

Примеры тестовых заданий:

N: 1

Q: Язык Java является:

-: машинным;

-: процедурным;

+: объектно-ориентированным;

-: процедурным и машинным.

N: 1

Q: Что является выходом компилятора Java?

+: байт-код;

-: выполняемый код;

-: файл класса;

-: машинный код.

N: 1

Q: В чем основное отличие языка Java от других языков высокого уровня?

-: простота кода;

-: объектно-ориентированный язык;

+: платформено-независимость Java-приложений;

-: высокая эффективность.

N: 1

Q: Какая утилита инструментального набора JDK позволяет выполнять Java-программу:

-: Javac;

-: Javadoc;

-: Jar;

+: Java.

Проверочная работа №1 (типы данных)

Вариант 1

Задание 1. Дан фрагмент программы:

```
_____ s =120;  
System.out.println(s-20);
```

Какой тип данных нужно подставить вместо прочерка, чтобы программа работала корректно, переменная s занимала наименьший объем памяти, а в результате исполнения на экран вывелось натуральное число?

Задание 2. Что выведется на экран в результате работы следующего фрагмента программы?

```
float w =2.5F;  
int z =10/(int) w;  
System.out.println(z+w);
```

Задание 3. Что выведется на экран в результате работы следующего фрагмента программы?

```
float y =2F;  
Sys-  
tem.out.println((int)y+"2"+y*1.5);
```

Вариант 3

Задание 1. Дан фрагмент программы:

```
_____ s =990;  
System.out.println(s*2);
```

Какой тип данных нужно подставить вместо прочерка, чтобы программа работала корректно, пе-

Вариант 2

Задание 1. Дан фрагмент программы:

```
_____ s =3;  
System.out.println(10/s);
```

Какой тип данных нужно подставить вместо прочерка, чтобы программа работала корректно, переменная s занимала наименьший объем памяти, а в результате исполнения на экран вывелось вещественное число?

Задание 2. Что выведется на экран в результате работы следующего фрагмента программы?

```
short w =9;  
double z = w *1.5;  
System.out.println(w-  
(int) z);
```

Задание 3. Что выведется на экран в результате работы следующего фрагмента программы?

```
double y =3.5;  
System.out.println("7"+y*3);
```

Вариант 4

Задание 1. Дан фрагмент программы:

```
_____ s =3;  
System.out.println(7.5/s);
```

Какой тип данных нужно подставить вместо прочерка, чтобы программа ра-

переменная s занимала наименьший объем памяти, а в результате исполнения на экран вывелось натуральное число?

Задание 2. Что выведется на экран в результате работы следующего фрагмента программы?

```
int w =10;
double z =4;
System.out.println(1+w/z);
```

Задание 3. Что выведется на экран в результате работы следующего фрагмента программы?

```
double y =3;
System.out.println((int)y+6+"y"+6);
```

ботала корректно, переменная s занимала наименьший объем памяти, а в результате исполнения на экран вывелось вещественное число?

Задание 2. Что выведется на экран в результате работы следующего фрагмента программы?

```
double w =1.75;
int z =(int) (2*w);
System.out.println(w*2+z);
```

Задание 3. Что выведется на экран в результате работы следующего фрагмента программы?

```
byte y =14;
System.out.println(2+y+"2"+y);
```

Проверочная работа №2 (условный оператор, генерация псевдослучайных чисел)

Вариант 1

Задание 1. Дан исходный код программы:

```
classExample{
    publicstaticvoid
    main(String[]args) {
        _____ a =0==8%2;
        if(a || (a &&false)) {
            Sys-
            tem.out.println((1+1.5)+"2");
        }else{
            Sys-
            tem.out.println(3+"a"+"2"+a);
        }
    }
}
```

Какой тип данных из набора boolean, int, double нужно подставить вместо прочерка, чтобы программа компилировалась корректно?

Задание 2. Что выведется на экран в результате работы представленной выше программы?

Задание 3. В программе имеется объявленная

Вариант 2

Задание 1. Дан исходный код программы:

```
classExample{
    publicstaticvoid
    main(String[]args) {
        _____ a =5>3;
        if(a || (5+3)*2<16) {
            Sys-
            tem.out.println(4+"12"+2);
        }else{
            Sys-
            tem.out.println(2+12*2);
        }
    }
}
```

Какой тип данных из набора boolean, int, double нужно подставить вместо прочерка, чтобы программа компилировалась корректно?

Задание 2. Что выведется на экран в результате работы представленной выше программы?

переменная `s` типа `int`. Напишите выражение, в результате которого переменной `s` будет присвоено случайное значение из отрезка `[-9;0]`.

Вариант 3

Задание 1. Дан исходный код программы:

```
classExample{
    publicstaticvoid
    main(String[]args){
        _____ b =5.0;
        b = b*2;
        if(b >10|| b <-10){
            Sys-
            tem.out.println("12"+0+4);
        }else{
            Sys-
            tem.out.println(0+4+"12");
        }
    }
}
```

Какой тип данных из набора `boolean`, `int`, `double` нужно подставить вместо прочерка, чтобы программа компилировалась корректно?

Задание 2. Что выведется на экран в результате работы представленной выше программы?

Задание 3. В программе имеется объявленная переменная `s` типа `int`. Напишите выражение, в результате которого переменной `s` будет присвоено случайное значение из отрезка `[-6;2]`.

Задание 3. В программе имеется объявленная переменная `s` типа `int`. Напишите выражение, в результате которого переменной `s` будет присвоено случайное значение из отрезка `[4;12]`.

Вариант 4

Задание 1. Дан исходный код программы:

```
classExample{
    publicstaticvoid
    main(String[]args){
        _____ a =-3.0;
        a =2+ a;
        if(3>2||3>2&&false){
            Sys-
            tem.out.println("4"+a);
        }else{
            System.out.println(-
            a+"4");
        }
    }
}
```

Какой тип данных из набора `boolean`, `int`, `double` нужно подставить вместо прочерка, чтобы программа компилировалась корректно?

Задание 2. Что выведется на экран в результате работы представленной выше программы?

Задание 3. В программе имеется объявленная переменная `s` типа `int`. Напишите выражение, в результате которого переменной `s` будет присвоено случайное значение из отрезка `[-6;8]`.

Проверочная работа №3

Пример класса №1

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 кон-

структуре (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.

Пример класса №2

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, вычисляющий в радианах угол равный данному и лежащий в пределах $[0;2\pi)$.

С использованием построенного класса создайте угол в 1085° и с помощью созданного метода вычислите и выведите на экран равный ему угол в радианах.

7.1.3. Примерные задания к промежуточному контролю

Вопросы зачета по курсу «Java-программирование» для студентов 3 курса ФМиКН направления ФИИТ, 5 семестр.

1. Выполнение Java-программы. Средства в составе пакета JDK. Кроссплатформенность.
2. Понятие байт кода. Типы Java-программ. Различия между приложениями и апплетами.
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. Обработка событий от кнопки, от мыши. Реализация интерфейсов.
20. Апплеты - создание, жизненный цикл, вывод апплета.
21. Апплеты – вывод изображений, передача параметров.
22. Классы Graphics, Colors, Fonts.
23. Обработка исключений. Классы Error и Exception.
24. Ключевые слова: try, catch, throw, throws, finally.
25. Потоки, 2 способа создания потоков.
26. Состояния потоков, два типа потоков.
27. 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 лабораторных заданий – 40 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов.

Промежуточный контроль по дисциплине включает:

- экзамен - 100 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. Учебно-методическое обеспечение дисциплины

Адрес сайта курса / Основная литература:

1. Патрик Ноутон, Герберт Шилдт. Java™2. Наиболее полное руководство. Санкт-Петербург. “БХВ-Петербург” 2013, 1050 стр.
2. В.Будилов. Интернет-программирование на Java. С.Пб: ВHV-Санкт-Петербург, 2014, 694 стр.
3. Вебер Д. Технология Java в подлиннике. С.Пб: ВHV-Санкт-Петербург, 2012, 1104 стр.
4. Эфеган М. JAVA Справочник. С.Пб: Питер, 2013. 448 стр.
5. Мейнджер Д. JAVA: Основы программирования. С.Пб: ВHV-Санкт-Петербург, 2012, 320 стр.

6. Мейсо Б. JAVA ++: Основы программирования. 2014, 400 стр.
7. Пономарчук Ю. В., Кузнецов И. В. Программирование на языке Java: Учебное пособие [Электронный ресурс]: учебник для бакалавров, магистров — Электрон. текстовые данные - Дальневосточный государственный университет путей сообщения 2021.— 103 с.— Режим доступа: <https://e.lanbook.com/book/259451>.— ЭБС.
8. Гуськова О.И.. **Объектно ориентированное программирование в Java: учебное пособие**: [Электронный ресурс]: учебник для бакалавров.— Электрон. текстовые данные - Московский педагогический государственный университет 2018.— 240 с.— Режим доступа: <https://e.lanbook.com/book/122311>.— ЭБС

Дополнительная литература

1. Крис Джамса Библиотека программиста Java .- Jamsa Press, ООО "Попурри", 2009.
2. И.Ю.Баженова Язык программирования Java .- АО "Диалог-МИФИ", 2011
3. Нейл Бартлетт, Алекс Лесли, Стив Симкин Программирование на Java. Путеводитель .- The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 2012

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

<https://ru.wikipedia.org/wiki/Java>

<http://citforum.ru/>

<http://www.compdoc.ru/>

<http://www.emanual.ru/>

10. Методические указания для обучающихся по освоению дисциплины

1) Помимо выполнения практических заданий рекомендуется решить все упражнения (или большую часть) упражнений, 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 лабораторных 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, библиотеки Java, одна из программных оболочек, к примеру, NetBeans.

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. 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.