

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
02.03.02 - Фундаментальная информатика и информационные
технологии

Направленность (профиль) подготовки:
Информатика и компьютерные науки

Уровень высшего образования
бакалавриат

Форма обучения
очная

Статус дисциплины:
входит в часть ОПОП, формируемую участниками образовательных
отношений, как дисциплина по выбору

Махачкала, 2021

Рабочая программа дисциплины «Java – программирование» составлена в 2021 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02 - Фундаментальная информатика и информационные технологии. Приказ № 807 Минобрнауки России от 23.08.2017 г.

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 30.05.2021 г., протокол № 9.

Зав. кафедрой  Магомедов А.М.

и
на заседании Методической комиссии факультета математики и компьютерных наук от 23.06.2021 г., протокол № 6.

Председатель  Бейбалаев В.Д.

Рабочая программа дисциплины согласована с учебно-методическим управлением « 09 » 07 2021 г.

Начальник УМУ  Гасангаджиева А.Г.

Аннотация рабочей программы дисциплины

Дисциплина “Java-программирование” входит в часть ОПОП, формируемую участниками образовательных отношений, как дисциплина по выбору 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, разработкой Web-приложений, созданием консольных и GUI-приложений.

Дисциплина нацелена на формирование следующих компетенций выпускника: ОПК-3, ПК-4.

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 зачета в конце семестра.

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

Се- местр	Учебные занятия						Форма промежуточной аттестации
	в том числе						
	Контактная работа обучающихся с преподавателем					СРС	
	Все го	из них					
Лек- ции		Лабораторные занятия	Практи- ческие занятия	КСР	консультации		
5	108	30	30			48	зачет

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 02.03.02 - Фундаментальная информатика и информационные технологии и изучается в соответствии с графиком учебного процесса в пятом семестре. Изучение предмета производится в течение одного семестра и заканчивается зачетом.

Дисциплина опирается на знания, полученные в 1 семестре в процессе изучения дисциплин «Основы программирования», во 2 семестре «Языки программирования», в 3 семестре «Программирование на основе классов», в 4 семестре «Основы Web-программирования». В свою очередь, на материал данной дисциплины опираются дисциплины «Программная инженерия» и «Введение в Case-технологии»; знания, умения и навыки, полученные при изучении данной дисциплины, используются, закрепляются и развиваются при проведении преддипломной 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-3. Способен к разработке алгоритмических и про-	ОПК-3.1. Знает методы теории алгоритмов, методы си-	<i>Знает:</i> базовые основы современного математического ап-	Конспектирование и проработка теоретического материала.

граммных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.	<p>стемного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей.</p>	<p>парата, связанного с проектированием программных продуктов и программных комплексов, основы дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов.</p> <p><i>Умеет:</i> использовать математический аппарат в профессиональной деятельности.</p> <p><i>Владеет:</i> практическим опытом применения современного математического аппарата, связанного с проектированием и разработкой программных продуктов и программных комплексов.</p>	Участие в практических занятиях. Реализация проектов на лабораторных занятиях. Самостоятельная работа.
	<p>ОПК-3.2. Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем.</p>	<p><i>Знает:</i> базовые понятия в области математических наук и программирования.</p> <p><i>Умеет:</i> находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности.</p> <p><i>Владеет:</i> практическим опытом научно-исследовательской деятельности в математике и информатике.</p>	
	<p>ОПК-3.3. Имеет практический опыт применения разработки программного обеспечения.</p>	<p><i>Знает:</i> основные методы проектирования и производства программного</p>	

		<p>продукта.</p> <p><i>Умеет:</i> использовать методы проектирования и производства программного продукта.</p> <p><i>Владеет:</i> практическим опытом применения указанных выше методов и технологий.</p>	
<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.</p>	<p>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>	<p><i>Знает:</i> основные методы решения прикладных задач, современные методы информационных технологий, языки программирования и современные информационные технологии.</p> <p><i>Умеет:</i> корректно оформить результаты научного труда в соответствии с современными требованиями; составлять программы на современных языках программирования.</p> <p><i>Владеет:</i> Имеет практический опыт использования сети Интернет, аннотирования, реферирования, библиографического разыскания и описания, опыт работы с научными источниками; навыками программирования на современных языках</p>	<p>Конспектирование и проработка теоретического материала. Участие в практических занятиях. Реализация проектов на лабораторных занятиях. Самостоятельная работа.</p>
	<p>ПК-4.2. Умеет реализовывать численные методы решения прикладных задач в профессиональной сфере деятельности, пакеты программного обеспечения, операционные системы,</p>	<p><i>Знает:</i> Знает методику установки и администрирования программных систем.; различные языки программирования.</p> <p><i>Умеет:</i> реализовывать техническое со-</p>	

	<p>электронные библиотеки, сетевые технологии</p>	<p>провожение информационных систем; применять различные языки программирования в численном анализе. <i>Владеет:</i> Имеет практический опыт разработки интеграции информационных систем с использованием аппаратно-программных комплексов</p>	
	<p>ПК-4.3. Имеет практический опыт разработки интеграции информационных систем.</p>	<p><i>Знает:</i> современные языки программирования и методы параллельной обработки данных. Знаком с содержанием Единого Реестра Российских программ для электронных вычислительных машин и баз данных; современные информационные технологии. <i>Умеет:</i> реализовывать численные методы решения прикладных задач в профессиональной сфере деятельности, пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии, с использованием современных информационных технологий. <i>Владеет:</i> навыками решения задач с использованием различных информационных технологий, имеет практический опыт разработки интеграции информационных систем.</p>	

4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 3 зачетные единицы, 108 академических часов: 30ч. лекций, 30 ч. лабораторных занятий, 48ч. – СРС.

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	Практические занятия	Лабораторные занятия	Контроль самост. раб.		
Модуль 1. Основы языка Java.									
1	Тема 1_1. Введение в программирование на Java. Инструментальные средства разработки Java-программ.	5	1	1		1		2	Текущий контроль - тест 1.1
2	Тема 2_1. Базовые элементы языка. Типы.	5	2	1		1		2	Текущий контроль - тест 2.1
3	Тема 3_1. Базовые элементы языка. Операторы.	5	3,4	2		2		4	Текущий контроль - тест 3.1
4	Тема 4_1. Основы объектно-ориентированного программирования для Java. Классы и объекты.	5	5,6,7	2		2		4	Текущий контроль - тест 4.1
5	Тема 5_1. Работа со строками.	5	8	2		2		2	Текущий контроль - тест 5.1
6	Тема 6_1. Обработка исключений на Java.	5	9	2		2		2	Текущий контроль - тест 6.1

	<i>Итого по модулю 1:</i>		1-9	10		10		16	Контрольная работа №1
Модуль 2. GUI-программирование и апплеты									
1	Тема 1_2. Создание апплета.	5	10	2		2		2	Текущий контроль - тест 1.2
2	Тема 2_2. Компиляция и выполнение апплета. Передача параметров в апплет.	5	10	2		2		2	Текущий контроль - тест 2.2
3	Тема 3_2. Средства пользовательского интерфейса. Компоненты и контейнеры.	5	11	2		2		2	Текущий контроль - тест 3.2
4	Тема 4_2. События. Обработка событий от компонент.	5	11	2		2		6	Текущий контроль - тест 4.2
5	Тема 5_2. Растровые изображения и анимация в апплетах. Звук в апплетах Java.	5	12	2		2		4	Текущий контроль - тест 5.2
	<i>Итого по модулю 2:</i>		10-12	10		10		16	Коллоквиум
Модуль 3. Сетевые приложения на Java									
6	Тема 6_2. Создание, выполнение и синхронизация потоков. Многопоточность. Приоритеты потоков.	5	13	2		2		2	Текущий контроль - тест 6.2
7	Тема 7_2. . Создание потоков связанных с файлами. Файловый ввод/вывод.	5	14	1		1		2	Текущий контроль - тест 7.2
8	Тема 8_2. Программирование меню на Java.	5	15	1		1		2	Текущий контроль - тест 8.2
9	Тема 9_2. Понятие	5	16	2		2		2	Текущий контроль -

	сервлета.								тест 9.2
10	Тема 10_2. Архитектура, жизненный цикл, размещение сервлетов.	5	16	2		2		4	Текущий контроль - тест 10.2
11	Тема 11_2. Технология передачи файлов из браузера в сервлет. Доступ к базам данных из сервлета Java.	5	17	1		1		6	Текущий контроль - тест 11.2
12	Тема 12_2. Особенности создания сетевых приложений. Класс URL в библиотеке классов Java.	5	18	1		1		6	Текущий контроль - тест 12.2
	<i>Итого по модулю 3:</i>		13-18	10		10		16	Контрольная работа №2
	ИТОГО:			30		30		48	Зачет

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

Модуль 3. Сетевые приложения на Java

Тема 1_3. Создание, выполнение и синхронизация потоков. Многопоточность

Процессы, потоки и приоритеты. Создание потоков.

Синхронизация. Методы synchronized. Операторы synchronized.

Методы wait и notify. Планирование потоков.

Взаимная блокировка. Приостановка потоков. Прерывание потока.

Завершение работы потока. Завершение приложения.

Использование Runnable. Ключевое слово volatile.

Безопасность потоков и ThreadGroup. Отладка потоков.

Реализация многопоточности в Java.

Лабораторная работа.

Текущий контроль - тест1.4

Тема 2_3. . Создание потоков связанных с файлами. Файловый ввод/вывод.

Потоки ввода и вывода.

Файловый ввод/вывод.

Построчный и побайтный ввод/вывод.

Лабораторная работа.

Текущий контроль - тест2.4

Тема 3_3. Программирование меню на Java.

Классы MenuBar, Menu и MenuItem.

Создание меню в окне типа Frame.

Отображение меню.

Лабораторная работа.

Текущий контроль - тест3.4

Тема 4_3. Понятие сервлета.

Класс HttpServlet. Структура сервлета.

Создание и инициализация сервлета.

API для работы с сервлетами.

Обработка клиентских запросов.

Лабораторная работа.

Текущий контроль - тест1.5

Тема 5_3. Архитектура, жизненный цикл, размещение сервлетов.

Многопоточность и сервлеты.

Размещение сервлетов.

Технология передачи файлов из браузера в сервлет.

Доступ к базам данных из сервлета Java.

Лабораторная работа.

Текущий контроль - тест2.5

Тема 6_3. Технология передачи файлов из браузера в сервлет. Доступ к базам данных из сервлета Java.

Доступ к БД из сервлета.

Взаимосвязь апплет-сервлет.

Лабораторная работа.

Текущий контроль - тест3.5

Тема 7_3. Особенности создания сетевых приложений. Класс URL в библиотеке классов Java.

Адрес IP и класс InetAddress.

Универсальный адрес ресурсов URL.

Класс URL в библиотеке классов Java.

Связь приложений Java с расширениями сервера Web.

Приложения SocketServ и SocketClient.

Лабораторная работа.

Текущий контроль - тест4.5

4.3.2. Содержание лабораторных занятий по дисциплине

Лабораторная работа по теме:

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.

Лабораторная работа по теме:

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.

Лабораторная работа по теме:

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.

Лабораторная работа по теме:

Работа со строками.

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

Лабораторная работа по теме:

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.

Лабораторная работа по теме:

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.

Лабораторная работа по теме:

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.

Лабораторная работа по теме:

События. Обработка событий от компонентов.

1. Описать процесс обработки событий от компонентов.

2. Написать программу демонстрирующую работу обработчиков событий.

Лабораторная работа по теме:

Растровые изображения и анимация в апплетах.

Звук в апплетах Java.

1. Написать апплет, выполняющий загрузку изображения.
2. Написать апплет, выполняющий вывод анимации.
3. Написать программу воспроизводящую аудио файл.

Лабораторная работа по теме:

Создание, выполнение и синхронизация поток. Многопоточность. Приоритеты потоков.

1. Описать класс Thread, на примере управления выполняющимся в данный момент подпроцессом.
2. Написать программу создающую два подпроцесса.
3. Написать программу иллюстрирующую работу оператора synchronized.

Лабораторная работа по теме:

Создание потоков связанных с файлами. Файловый ввод/вывод.

1. Написать программу, считывающую данные с файла, используя класс FileInputStream.
2. Написать программу, иллюстрирующую работу с записью данных в файлы, используя класс FileOutputStream.
3. Описать классы для работы с побайтным вводом/выводом.

Лабораторная работа по теме

Программирование меню на Java.

Написать апплет создающий стандартное меню.

Лабораторная работа по теме:

Понятие сервлета. Архитектура, жизненный цикл, размещение сервлетов

1. Составить описание сервлетов.
2. Создать и выполнить сервлет.

Лабораторная работа по теме:

Особенности создания сетевых приложений. Класс URL в библиотеке классов Java.

1. Описать адрес IP и класс InetAddress.
2. Описать класс URL и показать работу с ним, на примере программы.
3. Сокеты и работа с ними.

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 обуче-	Усвоение теоретических знаний, развитие	Мультимедийные лекции-объяснение, лекция-

	ния	мышления, формирование профессионального интереса к будущей деятельности	визуализация, с привлечением формы тематической дискуссии, беседы, анализа конкретных ситуаций
Лабораторные 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 личностно-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. Основы языка Java.	Особенности языка и платформы Java. Классификация программ по типу исполнения (компилируемые, интерпретируемые,	3 и 9 недели обучения. Проверка тео-	[1] – [5]; матери-

	<p>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. Интегриро- ванные среды разработки. Встроенные типы данных. Способы задания литералов раз- личных типов. Хранение данных в памяти ЭВМ. Приведение типов (явное и автомати- ческое). Константы и переменные. Оператор присваивания. Порядок действий (приори- тет операторов). Арифметические операто- ры. Операторы инкремента и декремента. Встроенный класс Math. Псевдослучайные числа. Операторы сравнения и логические операторы. Операторы ветвления. Услов- ный оператор. Минимизация количества проверок. Операторы ветвления. Оператор множественного выбора. Его сравнение с условным оператором.</p>	<p>ретических знаний на уст- ном опросе и коллоквиуме. Проверка ре- шенных задач.</p>	<p>алы сай- тов: https://ru.wikipedia.org/wiki/Java http://citforum.ru/</p>
<p>Модуль 2. GUI- программиро- вание и аппле- ты.</p>	<p>Основные понятия ООП. Объекты и классы. Абстракция данных. Сценарий построения объектно-ориентированной программы. Члены классов. Методы и поля. Специальные методы классов (конструкторы). Конструктор по умол- чанию. Модификаторы уровня доступа (default, public, protected, private). Ос- новополагающие принципы ООП. Ин- капсуляция. Основополагающие прин- ципы ООП. Наследование. Управление наследованием.</p>	<p>10 и 15 недели обучения.</p> <p>Проверка тео- ретических знаний на уст- ном опросе и коллоквиуме.</p> <p>Проверка выполнения компьютерных программ</p>	<p>[3], [4]; матери- алы сай- тов: http://www.emanual.ru/</p>
<p>Модуль 3. Се- тевые прило- жения на Java</p>	<p>Интерфейсы как средство реализации мно- жественного наследования. Основополага- ющие принципы ООП. Полиморфизм. Средства реализации полиморфизма. Иерархия классов Java. Коренной класс Object и его методы. Исключительные ситу- ации. Обработка исключительных ситуаций. Приложения с графическим интерфейсом с использованием GUI-пакетов и апплеты.</p>	<p>16 и 18 недели обучения.</p> <p>Проверка тео- ретических знаний на уст- ном опросе и коллоквиуме.</p> <p>Проверка выполнения компьютерных программ</p>	<p>[2], [6] ; матери- алы сай- тов: https://ru.wikipedia.org/wiki/Java http://www.emanual.ru/</p>

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 лабораторном 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.

Модуль 3. Сетевые приложения на Java

Тема 1_3. Создание, выполнение и синхронизация потоков. Многопоточность

Тема 2_3. . Создание потоков связанных с файлами. Файловый ввод/вывод.

Тема 3_3. Программирование меню на Java.

Тема 4_3. Понятие сервлета.

Тема 5_3. Архитектура, жизненный цикл, размещение сервлетов.

Тема 6_3. Технология передачи файлов из браузера в сервлет. Доступ к базам данных из сервлета Java.

Тема 7_3. Особенности создания сетевых приложений. Класс URL в библиотеке классов 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;
- + : платформу-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?

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 фраг-

```
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;
System.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, 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);
```

мента 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 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?

```
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:

Вариант 2

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 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?

```

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?

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 $[-6;2]$.

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 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.

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:

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

8. Учебно-методическое обеспечение дисциплины

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.

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 лабораторных 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 зачета 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. Лабораторные занятия проводятся в компьютерных классах с современным аппаратным и программным обеспечением. На сайте кафедры размещаются учебные пособия и презентации к лекции.