

МИНОБРНАУКИ РОССИИ
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
факультет математики и компьютерных наук

Образовательная программа
02.03.01 Математика и компьютерные науки

Профиль подготовки
Математический анализ и его применения

Уровень высшего образования
бакалавриат

Форма обучения
очная

Статус дисциплины: вариативная по выбору

Махачкала, 2018

Рабочая программа дисциплины «Java – программирование» составлена в 2018 году в соответствии с требованиями ФГОС ВО по направлению подготовки – 02.03.01 Математика и компьютерные науки (уровень бакалавриата) от 07.08.2014 № 949.

Разработчик: к.ф.-м.н., доцент по специальности 01.01.07 - «вычислительная математика и математическая кибернетика» Шихиев Шукур Бабаевич.

Рабочая программа дисциплины одобрена:

На заседании кафедры дискретной математики и информатики от 27.04.2018 протокол № 8

Зав. кафедрой  Магомедов А. М.

на заседании Методической комиссии факультета математики и компьютерных наук от протокол №

Председатель  Бейбалаев В. Д.

Рабочая программа дисциплины согласована с учебно-методическим управлением 28.06.2018

Аннотация рабочей программы дисциплины

Дисциплина “Java-программирование” входит в вариативную по выбору часть образовательной программы бакалавриата по направлению 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: ОПК-2, ОПК-4 и ПК-9.

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						Форма проме- жуточной атте- стации	
	в том числе							
	Контактная работа обучающихся с преподавателем					СРС в том числе экзамен		
	Всего	из них						
Лек- ции		Лабора- торные занятия	Практи- ческие занятия	КСР	консуль- тации			
8	108	30		30		0	12+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 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 используются, закрепляются и развиваются при проведении преддипломной практики, выполнении выпускной квалификационной работы.

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

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
ОПК-2	способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;	Знать: основы языка Java, алгоритмические и программные решения в области системного и прикладного программирования. Уметь: использовать типы, операции, управляющие структуры компоненты Javav разработке приложений, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей. Владеть: навыками объектно-ориентированного программирования, создания консольных программ, и GUI-приложений и апплетов.
ОПК-4	способностью находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем;	Знать: элементы языка Java: типы, операторы, иерархию классов и интерфейсов; основы алгоритмизации, основы оптимального представления входных данных с использованием наиболее подходящих структур. Уметь: разрабатывать простые алгоритмы и воплощать в программы, использовать современные операционные системы и оболочки при создании программных приложений.

		Владеть: развитыми навыками объявления и применения
ПК-9	способностью к организации учебной деятельности в конкретной предметной области (математика, физика, информатика)	Знать: основы современных методов программирования, механизмы взаимодействия web-сервера и клиента. Уметь: применять методы объектно-ориентированного, визуального и событийно управляемого программирования. Владеть: начальными навыками разработки программ с применением методов объектно-ориентированного, визуального и событийно-управляемого программирования навыками работы в различных программных средах.

4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет 3 зачетные единицы, 108 академических часов: 30 ч. лекций, 30 ч. практических занятий, 12 ч. – СРС, 36 ч. - экзамен.

4.2. Структура дисциплины

Структура и содержание дисциплины «Java-программирование»

№ п.п.	Разделы и темы дисциплины	Се- ме- ст- р	Не- де- ля- се- ме- ст- ра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Ле- к- ци- и	Пра- ктич- заня- тия	Са- мо- ст- ра- ба	Кон- троль сам- ост.раб.	
Модуль 1. Основы языка Java. GUI-программирование и апплеты.								
1	Тема 1. Введение в программирование на Java. Инструментальные средства разработки Java-программ.	8	1	1	1		Устный опрос	Текущий контроль-тест1.1
2	Тема 2. Базовые элементы языка. Типы.	8	2	1	1		Дом. Самост	Текущий контроль-тест2.1
3	Тема 3.Базовые элементы языка. Операторы.	8	2	1	1		Письм опрос	Текущий контроль-тест3.1
4	Тема 4. Основы объектно-	8	3	1	1		Устный опрос	Текущий контроль-тест4.1

	ориентированного программирования для Java. Классы и объекты.							
5	Тема 5. Работа со строками.	8	4	1	1		Дом. Самост	Текущий контроль-тест5.1
6	Тема 6. Обработка исключений на Java.	8	4	1	1	1	Письм опрос	Текущий контроль-тест6.1
7	Тема 7. Создание апплета.	8	5	1	1	1	Устный опрос	Текущий контроль-тест1.2
8	Тема 8. Компиляция и выполнение апплета. Передача	8	6	1	1	1	Дом. Самост	Текущий контроль-тест2.2
9	Тема 9. Средства пользовательского интерфейса. Компоненты и контейнеры.	8	6	2	2	1	Письм опрос	Текущий контроль-тест.
10	Тема 10. События. Обработка событий от компонент.	8	7	2	2	1	Устный опрос	Текущий контроль-тест.
11	Тема 11. Растровые изображения и анимация в апплетах. Звук в апплетах Java.	8	8	2	4	1	Дом. Самост	Текущий контроль-тест.
	<i>Итого по модулю 1:</i>		1-8	14	16	6	Письм опрос	Текущий контроль-тест.
Модуль 2. Сетевые приложения на Java								
12	Тема 12. Создание, выполнение и синхронизация потоков. Многопоточность. Приоритеты потоков.	8	9	2	2	1	Устный опрос	Текущий контроль-тест.
13	Тема 13. . Создание потоков связанных с файлами. Файловый ввод/вывод.	8	10	2	2	1	Дом. Самост	Текущий контроль-тест.

14	Тема 14. Программирование меню на Java.		11	2	2	1	Письм опрос	Текущий контроль- тест.
15	Тема 15. Понятие сервлета.		12	2	2	1	Устный опрос	Текущий контроль- тест.
16	Тема 16. Архитектура, жизненный цикл, размещение сервлетов.		13	2	2	1	Дом. Самост	Текущий контроль- тест.
17	Тема 17. Технология передачи файлов из браузера в сервлет. Доступ к параметров в апплет		14	2	2	1	Письм опрос	Текущий контроль- тест.
18	Тема 18. Особенности создания сетевых приложений. Класс URL в библиотеке классов Java.		15	4	2		Устный опрос	Текущий контроль- тест.
	<i>Итого по модулю 2</i>		12- 15	16	14	6		
	<i>Экзамен</i>						36	
	Всего:		108	30	30	12	36	

4.3. Содержание дисциплины, структурированное по темам (разделам).

4.3.1 Содержание лекционных занятий по дисциплине.

Модуль 1. Основы языка Java. GUI-программирование и апплеты.

Тема 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.

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. 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.
Тема 5. Работа со строками. Классы String, StringBuffer. Доступ к символам.
Равенство строк. Сравнение. Упорядочение. Методы работы со строками.
Методы indexOf, lastIndexOf.
Тема 6. Обработка исключений на Java. Типы исключений.
Вложенные операторы try. Блок finally. Операторы throw, catch.
Подклассы Exception.
Тема 7. Создание апплета. Создание апплета на Java. Параметры апплета.
Исходный текст апплета. Методы init, destroy, start, stop.
Тема 8. Компиляция и выполнение апплета. Передача параметров в апплет.
Компиляция и выполнение апплета. Выполнение апплета в отдельном потоке.
Параметры апплета. Компоненты в окне апплета.
Тема 9. Средства пользовательского интерфейса. Компоненты и контейнеры.
Кнопки, переключатели. Списки класса Choice. Списки класса List.
Текстовое поле классов Label, TextField.
Многострочное текстовое поле класса TextArea.
Тема 10. События. Обработка событий от компонент. Обработка событий.
Метод handleEvent. Класс MouseEvent и интерфейс MouseListener.
Движение мыши, перетаскивание. Событие от клавиатуры.
Обработка событий, инициированных компонентами.
Тема 11. 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.
Модуль 3. Сетевые приложения на Java
Тема 12. 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.
Тема 13. . Создание потоков связанных с файлами. Файловый ввод/вывод.
Потоки ввода и вывода. Файловый ввод/вывод.
Построчный и побайтный ввод/вывод.
Тема 14. Программирование меню на Java. Классы Menubar, Menu и MenuItem.
Создание меню в окне типа Frame. Отображение меню.
Тема 15. Понятие сервлета. Класс HttpServlet. Структура сервлета.
Создание и инициализация сервлета. API для работы с сервлетами.
Обработка клиентских запросов.
Тема 16. Архитектура, жизненный цикл, размещение сервлетов.
Многопоточность и сервлеты. Размещение сервлетов.

Технология передачи файлов из браузера в сервлет.

Доступ к базам данных из сервлета Java.

Тема 17. Технология передачи файлов из браузера в сервлет.

Доступ к базам данных из сервлета Java. Доступ к БД из сервлета.

Взаимосвязь апплет-сервлет.

Тема 18. Особенности создания сетевых приложений.

Класс URL в библиотеке классов Java.

Приемы построения сетевого приложения. Класс URL

4.3.2 Содержание практических 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.

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.

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.

Создание выполнение и синхронизация поток. Многопоточность. Приоритеты потоков.

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

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.

Программирование меню на Java.

1. Написать апплет создающий стандартное меню.

Понятие сервлета. Архитектура, жизненный цикл, размещение сервлетов

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

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 различные образовательные технологии и методы обучения:

- 5.1. Процесс изложения учебного материала сопровождается систематическими (на каждом занятии) компьютерными презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного оборудования.

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

5.3. Отличительные элементы используемых образовательных технологий: в обеспечении преподавания дисциплины используется ряд компьютерных программ, разработанных специально для обеспечения курса.

6. Учебно-методическое обеспечение самостоятельной работы студентов

6.1. 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

6.2. Порядок контроля:

1. Опрос на лабораторном занятии
2. Проверка выполнения домашних заданий и контрольных работ
3. Коллоквиум
4. Зачет.

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
1.	Виртуальная машина Java. JIT-компиляция. Создание простейшей программы на Java, её компиляция в байт-код и запуск. Исполнение программы на виртуальных машинах.	1. теоретических знаний на устном опросе и коллоквиуме. Проверка решенных задач.	риалы сайтов: https://ru.wikipedia.org/wiki/Java http://citforum.ru/
2	Средства разработки Java-приложений. Интегрированные среды разработки.	2. неделя устный опрос	[2] с. 40 – 54
3	Встроенные типы данных. Способы задания литералов различных типов. Хранение данных в памяти ЭВМ. Приведение типов (явное и автоматическое)	3 неделя письменный опрос	[2] с. 23 – 24 [2] с. 25 - 28 [2] с. 31 – 34
4	Константы и переменные. Оператор присваивания. Порядок действий (приоритет операторов). Арифметические операторы. Операторы инкремента и декремента.	4 неделя Составление и отладка программ	[2] с. 34 – 36 [2] с. 54 - 67
5	Встроенный класс Math. Псевдослучайные числа. Операторы сравнения и логические операторы.	5 неделя Коллоквиум	[1] с. 124 - 135 [1] с. 108 - 121 [2] с. 68 – 71,

	Операторы ветвления. Условный оператор. Минимизация количества проверок.		
6	Операторы ветвления. Оператор множественного выбора. Его сравнение с условным оператором.	6 неделя письменный опрос	[1] с. 138 – 151,
7	Основные понятия ООП. Объекты и классы. Абстракция данных.	7. Составление и отладка программ	[1] с. 156 – 179
8.	Сценарий построения объектно-ориентированной программы. Члены классов. Методы и поля...	8. Проверка теоретических знаний на устном опросе	[2] с. 78 - 84 [2] с. 88 - 93 [3] с. 43 – 48
9	Специальные методы классов (конструкторы)	9 неделя И на коллоквиуме.	[3] с. 50 – 53 [3] с. 54 – 58
10	Конструктор по умолчанию. Модификаторы уровня доступа (default, public, protected, private)	10. Проверка выполнения компьютерных программ	[3] с. 77 – 79 [3] с. 87 – 92
11	Основополагающие принципы ООП. Инкапсуляция. основополагающие принципы ООП. Наследование. Управление наследованием.	11 неделя письменный опрос	[2] с. 155 – 169
12	Интерфейсы как средство реализации множественного наследования.	12. Составление и отладка программ	[2] с. 169 – 177
13.	Основополагающие принципы ООП. Полиморфизм.	13 неделя Проверка теоретических знаний на устном опросе	[2] с. 182 – 188
14	Средства реализации полиморфизма.	13 неделя на коллоквиуме.	[3] с. 150 – 153 [3] с. 154 – 158
15	Иерархия классов Java.	14. Проверка выполнения компьютерных программ	[3] с. 177 – 179 [3] с. 87 – 92
16	Коренной класс Object и его методы.	14 письменный опрос	[2] с. 201 – 209
17	Исключительные ситуации. Обработка исключительных ситуаций.	15 неделя Составление и отладка программ	[2] с. 211 – 218
18	Приложения с графическим интерфейсом с использованием GUI-пакетов и апплеты.	15 неделя Повторение материала	[2] с. 220 – 225

Текущий контроль:

1. Проверка хода выполнения Лабораторных работ;

2. Проверка выполнения домашних заданий;

3. Промежуточная аттестация в форме письменной работы.

Текущий контроль включает, кроме еженедельного опроса и проверки знаний по текущему материалу, ведение электронного журнала посещаемости, проверку выполнения компьютерных программ. Подразумевается непрерывное общение по электронной почте (общение по скайпу не целесообразно, т.к. не позволяет осуществлять доскональную проверку заданий).

Промежуточный контроль проводится в виде письменной работы, рассчитанной на 20 - 30 минут.

Итоговый контроль проводится в виде письменной работы с обязательным устным собеседованием по результатам предварительной проверки.

Критерии выставления оценок «отлично», «хорошо», «удовлетворительно» определяются степенью владения материалом и достигнутым уровнем компетентности в решении задач дискретной математики. В исключительных случаях учитываются успехи на всероссийских олимпиадах и конкурсах по номинации данной дисциплины.

Для обеспечения самостоятельной работы используется разработанный на кафедре пакет заданий и методических указаний, издано учебное пособие с алгоритмами решения базовых заданий по дискретной математике и соответствующими программами на языке Дельфи. Самостоятельная работа студентов складывается из проработки лекционного материала, материала учебника и соответствующих форумов интернет, решения всех заданий из индивидуальных заданий, решения рекомендуемых задач, подготовки к сдаче промежуточных форма контроля, и выполнения Лабораторных работ.

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 выведете 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.

Отчет принимается в виде реферата с выполненными Лабораторными работами

№	Семестр	Виды и содержание контрольных мероприятий
1	2	3
Модуль 1	2	Сдача лабораторных работ № 1,2,3,4
Модуль 2	2	Сдача лабораторных работ № 5,6,7
Модуль 3	2	Сдача лабораторных работ № 8,9,10

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

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Процедура освоения
ОПК-2	способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с	Знать: основы языка Java, алгоритмические и программные решения в области системного и прикладного программирования. Уметь: использовать типы, операции, управляющие структуры компоненты Javav разработке приложений, информационных и имитационных моделей, созданию	Изучение тем: 1,2,3,4,5 модуля №1. Выполнение тестовых заданий на каждом лабораторном занятии для проверки усвоения самостоятельных заданий.

	применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;	информационных ресурсов глобальных сетей. Владеть: навыками объектно-ориентированного программирования, создания консольных программ, и GUI-приложений и апплетов.	
ОПК-4	способностью находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем;	Знать: элементы языка Java: типы, операторы, иерархию классов и интерфейсов; основы алгоритмизации, основы оптимального представления входных данных с использованием наиболее подходящих структур. Уметь: разрабатывать простые алгоритмы и воплощать в программы, использовать современные операционные системы и оболочки при создании программных приложений. Владеть: развитыми навыками объявления и применения	Составление программ с использованием различных типов и с применением всех структур управления. Составление, отладка и тестирование программ с использованием нескольких структурированных типов (с обязательным присутствием файловых данных)
ПК-9	способностью к организации учебной деятельности в конкретной предметной области (математика, физика, информатика)	Знать: основы современных методов программирования, механизмы взаимодействия web-сервера и клиента. Уметь: применять методы объектно-ориентированного, визуального и событийно управляемого программирования. Владеть: начальными навыками разработки программ с применением методов объектно-ориентированного, визуального и событийно-управляемого программирования навыками работы в различных программных средах.	Отладка и тестирование программ по темам 1,2,3,4,5 модуля №1. Составление программ с использованием в отдельности файлов, массивов, строк, списков и коллекций. Изучение визуальных компонентов и действий с формами

7.2. Типовые контрольные задания Примеры тестовых заданий: Вопросы межсессионной аттестации: I - III модуль.

1. Темы рефератов и курсовых работ:
2. Типы данных. Ссылочные типы. Классы. Интерфейсы.
3. Логические операции. Преобразование типа. Переменные.
4. Операторы. Логические, присваивания. Приоритеты.
5. Операторы. Условные, выбора.
6. Операторы цикла. Три инструкции перехода.

7. Свойства ООП.
8. Классы. Передача параметров по значению и по ссылке. Ссылка this.
9. Модификаторы доступа. Типы методов: перегруженные, переопределенные. Ключевое слово super.
10. Пакеты. Включение пакета, класса пакета. Пакет AbstractWindowToolkit.
11. Контейнеры: фрейм, панель.
12. Компоненты GUI: Label, TextField, TextArea.
13. Компоненты GUI: Button, Checkbox, радиокнопки, списки.
14. Типы макетов.
15. Обработка событий от кнопки, от мыши. Реализация интерфейсов.
16. Апплеты - создание, жизненный цикл, вывод апплета.
17. Апплеты – вывод изображений, передача параметров.
18. Классы Graphics, Colors, Fonts.
19. Обработка исключений. Классы Error и Exception.
20. Ключевые слова: try, catch, throw, throws, finally.

7.2.2. Вопросы для оценки качества освоения теории

1. Найти сумму цифр заданного двузначного числа.
2. Переставить местами цифры двузначного числа.
3. Найти сумму цифр заданного трехзначного числа.
4. Найти сумму цифр заданного натурального числа.
5. Циклическая перестановка элементов массива
6. Найти делители заданного натурального числа
7. Является ли простым заданное натуральное число
8. Перечислить все трехразрядные двоичные числа.
9. Перечислить все трехразрядные троичные числа.
10. Найти сумму цифр заданного натурального числа. Выполнить форматированный вывод промежуточных вычислений.
11. Вывести на экран монитора все n-разрядные двоичные числа.
12. Вывести на экран монитора все n-разрядные двоичные числа k единицами.
13. Сколько существует целочисленных решений уравнения: $x_1 + x_2 + x_3 + x_4 = 7$?
Ответ: 120 (N=4, k=7)
14. Сколькими способами можно выбрать 13 из 52 игральные карты, различая их только по масти? Ответ: 560 (N=4, k=13).
15. Написать программу, которая по дате (числу и месяцу) определяет день недели (для этой даты).
16. Написать программу, которая распознает числовое арифметическое выражение. Например, $((123+66)-(88-64))$ – правильное выражение. А $((123+66)?-(88-64))$, $((123+66)-(88-64)$ – неправильные выражения.

7.2.3. Примерные упражнения и задания к практическим занятиям и для самопроверки

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.

4. 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).

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

6. 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.

7.2.4. Примеры вариантов к текущему контролю

1. Алфавит языка. Символы. Ключевые слова
2. Литералы (константы). Типы данных. Тип литерала по умолчанию.
3. Имена (идентификаторы). Переменные. Объявление переменных.
4. Арифметическое выражение. Числовые типы.
5. Арифметические операции. Арифметические функции (*Math.*).
6. Логическое выражение. Логические операции. Логические операции. Побитовые операции - логические операции над битами чисел.
7. Явное приведение и автоматическое расширение типов.
8. Сокращённые формы операторов (присваивания)

9. Управляющие инструкции: if; switch – case; while; do – while; for;
10. Массив
11. Классы и объекты.
12. Свойство классов: наследование. Примеры.
13. Свойство классов: инкапсуляция. Примеры.
14. Свойство классов: полиморфизм. Перегрузка метода. Примеры.
15. Создание копии объекта. Анонимный объект.
16. Статистические (static) поля. ([1] стр. 145)
17. Статистические (static) методы. ([1] стр. 148)
18. Ввод и вывод данных.
19. Форматированный вывод.
20. Обработка исключений. Пример на ArithmeticExpretion. ([1] стр. 227)
21. Простой и параметризованный конструктор. Примеры.
22. Абстрактный класс. Примеры.
23. Интерфейс. Примеры.
24. Графика на Java: цвета и рисование геометрических фигур.
25. Библиотеки awt и swing. Примеры работы с JFrame; JLabel; JTextField
26. Создание и исполнение апплета (Applet). Примеры.
27. Аргументы командной строки. Значения массива args из метода main().

7.2.5. Перечень вопросов к промежуточному (по частям) и итоговому контролю

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	Вариант 2
<p>Задание 1. Дан фрагмент программы: _____ s =120; System.out.println(s-20);</p> <p>Какой тип данных нужно подставить вместо про-черка, чтобы программа работала корректно, переменная s занимала наименьший объем памяти, а в результате исполнения на экран вывелось натуральное число?</p>	<p>Задание 1. Дан фрагмент программы: _____ s =3; System.out.println(10/s);</p> <p>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?</p>
<p>Задание 2. Что выведется на экран в результате работы следующего фрагмента программы?</p> <p>float w =2.5F; int z =10/(int) w; System.out.println(z+w);</p>	<p>Задание 2. Что выведется на экран в результате работы следующего фрагмента программы?</p> <p>short w =9; double z = w *1.5; System.out.println(w-(int)z);</p>
<p>Задание 3. Что выведется на экран в результате работы следующего фрагмента программы?</p> <p>float y =2F; System.out.println((int)y+"2"+y*1.5);</p>	<p>Задание 3. Что выведется на экран в результате работы следующего фрагмента программы?</p> <p>double y =3.5; System.out.println("7"+y*3);</p>

Вариант 1	Вариант 2
<p>Задание 1. Дан фрагмент программы: _____ s =990; System.out.println(s*2);</p> <p>Какой тип данных нужно подставить вместо прочерка, чтобы программа работала</p>	<p>Задание 1. Дан фрагмент программы: _____ s =3; System.out.println(7.5/s);</p>

корректно, переменная s занимала наименьший объём памяти, а в результате исполнения на экран вывелось натуральное число?	Какой тип данных нужно подставить вместо прочерка, чтобы программа работала корректно, переменная s занимала наименьший объём памяти, а в результате исполнения на экран вывелось вещественное число?
Задание 2. Что выведется на экран в результате работы следующего фрагмента программы? <code>int w =10; double z =4; System.out.println(1+w/z);</code>	Задание 2. Что выведется на экран в результате работы следующего фрагмента программы? <code>double w =1.75; int z =(int)(2*w); System.out.println(w*2+z);</code>
Задание 3. Что выведется на экран в результате работы следующего фрагмента программы? <code>double y =3; System.out.println((int)y+6+"y"+6); float y =2F; Sys-tem.out.println((int)y+"2"+y*1.5);</code>	Задание 3. Что выведется на экран в результате работы следующего фрагмента программы? <code>byte y =14; System.out.println(2+y+"2"+y);</code>

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

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.

Вопросы зачета по курсу «Java-программирование» для студентов 4 курса ФМиКН направления МиКН, 8 семестр.

1. Выполнение Java-программы. Средства в составе пакета JDK.
2. Понятие байт кода. Типы Java-программ. Приложения и апплеты.
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.
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. Потoki, 2 способа создания потоков.
26. Состояния потоков, два типа потоков.
27. Пакеты, управление доступом.

7.3. 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. Перечень основной и дополнительной учебной литературы, рекомендуемых для освоения дисциплины

Основная:

1. Вязовик Н.А. Программирование на Java [Электронный ресурс] / Н.А. Вязовик. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет

Информационных Технологий (ИНТУИТ), 2016. — 603 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73710.html>

2. Васильев А.Н. Самоучитель Java с примерами и программами — М. : ПРЕСС, 2011.

3. Яшин А.С. Java на примерах. Практика, практика и только практика [Электронный ресурс]/ Яшин А.С., Сеттер Р.В.— Электрон. текстовые данные.— СПб.: Наука и Техника, 2018.— 256 с.— Режим доступа: <http://www.iprbookshop.ru/78104.html>.— ЭБС «IPRbooks»

Дополнительная литература

1. Джошуа Блох Java. Эффективное программирование [Электронный ресурс] / Блох Джошуа. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 310 с. — 978-5-4488-0127-3. — Режим доступа: <http://www.iprbookshop.ru/64057.html>

2. Евдокимов П.В. С# на примерах [Электронный ресурс]/ Евдокимов П.В.— Электрон. текстовые данные.— СПб.: Наука и Техника, 2018.— 320 с.— Режим доступа: <http://www.iprbookshop.ru/78097.html>.— ЭБС «IPRbooks»

3. Сеттер Р.В. Изучаем Java на примерах и задачах [Электронный ресурс]/ Сеттер Р.В.— Электрон. текстовые данные.— СПб.: Наука и Техника, 2016.— 240 с.— Режим доступа: <http://www.iprbookshop.ru/44025.html>.— ЭБС «IPRbooks»

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.