

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
"Казанский (Приволжский) федеральный университет"
Институт вычислительной математики и информационных технологий



УТВЕРЖДАЮ

Проректор
по образовательной деятельности КФУ
Проф. Таюрский Д.А.

_____ 20__ г.

Программа дисциплины

Программирование на Java Б1.В.ДВ.12

Направление подготовки: 01.03.02 - Прикладная математика и информатика

Профиль подготовки: Системное программирование

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Байрашева В.Р.

Рецензент(ы):

Аблаев Ф.М.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Аблаев Ф. М.

Протокол заседания кафедры No ____ от " ____ " _____ 201__ г

Учебно-методическая комиссия Института вычислительной математики и информационных технологий:

Протокол заседания УМК No ____ от " ____ " _____ 201__ г

Регистрационный No

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Байрашева В.Р. кафедра теоретической кибернетики отделение фундаментальной информатики и информационных технологий , Venera.Bajrasheva@kpfu.ru

1. Цели освоения дисциплины

Курс "Программирование на Java" предназначен для ознакомления с так называемыми java-технологиями - информационными технологиями, основанными на использовании языка программирования Java, который уже более 20 лет является популярным средством разработки программного обеспечения в различных программных архитектурах. Курс призван дать представление о современных технологиях разработки на этом языке, включая объектно-ориентированную концепцию разработки, разработку апплетов, разработку пользовательского интерфейса с помощью JavaFX, работу с базами данных, разработку под мобильную платформу Android, а также разработку web-приложений как стандартного вида, так и по технологии web-сервисов. Отдельное место посвящено вопросам обеспечения безопасности информации в java-системах. Курс состоит из лабораторных занятий, на которых студенты осуществляют выполнение индивидуальных заданий, нацеленных на получение навыков разработки java-программ в указанных архитектурах. Часть лабораторных занятий проводится в форме дискуссий и обсуждений, что более напоминает активные формы лекций, в целях систематизированного изложения изучаемых принципов и акцентирования внимания студентов на особенностях тех или иных технологий

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел "Б1.В.ДВ.12 Дисциплины (модули)" основной образовательной программы 01.03.02 Прикладная математика и информатика и относится к дисциплинам по выбору. Осваивается на 4 курсе, 7 семестр.

Изучение дисциплины основывается на результатах изучения дисциплин, связанных с основами информатики, объектно-ориентированного программирования, баз данных. В дальнейшем знания и навыки, которые получают студенты при изучении дисциплины, могут быть применены при написании курсовой работы и выпускной квалификационной работы. В силу популярности java-технологий и разнообразия поддерживаемых архитектур приложений знания этой технологии могут помочь студентам в их будущей профессиональной карьере.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-1 (профессиональные компетенции)	Способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой
ОПК-2 (профессиональные компетенции)	Способность приобретать новые знания, используя современные образовательные и информационные технологии

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-3 (профессиональные компетенции)	Шифр компетенции ОПК-3 Расшифровка приобретаемой компетенции Способность к разработке алгоритмических и программных 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
ПК-7 (профессиональные компетенции)	Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения

В результате освоения дисциплины студент:

1. должен знать:

специфику программирования на языке Java, принципы построения языка, его особенности в сравнении с другими языками, основные этапы и тенденции развития ООП на языке Java, возможности, реализуемые технологией Java в интернете, различные паттерны проектирования программного обеспечения, реализованные в стандартных библиотеках Java, приемы контроля входных данных приложения.

2. должен уметь:

- ориентироваться в технологии JDBC, реализации компонентов JFC1, использовать компоненты других стандартных библиотек Java для решения профессиональных задач,
- создавать иерархию классов приложения, создавать диаграммы UML, извлекать полезную научно-техническую информацию из электронных библиотек, реферативных журналов, сети Интернет.

3. должен владеть:

теоретическими знаниями об основных компонентах языка и их использовании при написании программ, навыками самостоятельной работы при разработке и отладке программ, навыками работы в средах разработки программного обеспечения NetBeans/Eclipse для решения профессиональных задач.

4. должен демонстрировать способность и готовность:

- применять полученные знания в своей дальнейшей профессиональной деятельности

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 3 зачетных(ые) единиц(ы) 72 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 7 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
1.	Тема 1. Введение. Особенности языка Java.	7	1-2	0	0	2	Тестирование
2.	Тема 2. Конструкции языка. Прimitives и ссылочные типы данных.	7	3-4	0	0	4	Компьютерная программа
3.	Тема 3. ООП в Java. Пакет java.lang.	7	5-7	0	0	2	Контрольная работа Компьютерная программа
4.	Тема 4. Коллекции.	7	8-9	0	0	2	Компьютерная программа
5.	Тема 5. Исключения. Модульное тестирование.	7	10-12	0	0	2	Компьютерная программа
6.	Тема 6. Ввод и вывод.	7	13-18	0	0	4	Контрольная работа Компьютерная программа
7.	Тема 7. Java Foundation Classes (JFC).	7		0	0	6	Компьютерная программа
8.	Тема 8. Функциональное программирование. Лямбда-выражения.	7		0	0	2	Компьютерная программа
9.	Тема 9. Многопоточные приложения	7		0	0	2	Компьютерная программа
10.	Тема 10. Работа с базами данных средствами JDBC.	7		0	0	4	Компьютерная программа

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
11.	Тема 11. Система сборки Maven	7		0	0	2	Компьютерная программа
12.	Тема 12. Возможности Stream API и их использование.	7		0	0	2	Компьютерная программа
13.	Тема 13. Фреймворк Spring.	7		0	0	2	Компьютерная программа
.	Тема . Итоговая форма контроля	7		0	0	0	Зачет
	Итого			0	0	36	

4.2 Содержание дисциплины

Тема 1. Введение. Особенности языка Java.

лабораторная работа (2 часа(ов)):

Выполнение лабораторной работы "Принципы объектно-ориентированного программирования в Java". Работа включает решение задач с использованием разработки собственных классов, использовании библиотек классов языка Java (классов-коллекций, файлов и пр.).

Тема 2. Конструкции языка. Прimitives и ссылочные типы данных.

лабораторная работа (4 часа(ов)):

Выполнение лабораторной работы "Создание программ с графическим пользовательским интерфейсом". Изучаются технологии создания пользовательского интерфейса Swing и JavaFX. Лабораторная работа включает разработки оконного приложения или апплета с несколькими окнами, возможностями рисования и управления объектами с помощью мыши и клавиатуры.

Тема 3. ООП в Java. Пакет java.lang.

лабораторная работа (2 часа(ов)):

Выполнение лабораторной работы "Создание приложения по работе с базами данных". Изучаются способы подключения к СУБД и выполнении запросов. Лабораторная работа включает разработку программы с использованием стандартной технологии связи с базами данных JDBC и программы с применением шаблонов объектно-реляционного связывания.

Тема 4. Коллекции.

лабораторная работа (2 часа(ов)):

Выполнение лабораторной работы "Создание мобильного Android-приложения". Изучаются особенности Android-приложений. Лабораторная работа включает разработку мобильного приложения (например, игра) с использованием элементов интерфейса и рисования.

Тема 5. Исключения. Модульное тестирование.

лабораторная работа (2 часа(ов)):

Выполнение лабораторной работы "Создание web-приложение с помощью Java EE и технологии сервлетов". Изучается архитектура web-приложений с использованием сервера Apache Tomcat. Лабораторная работа включает разработку web-приложения с обязательным использованием базы данных (авторизация, просмотр каталогов, заказ). Разработка осуществляется с помощью сервлетов или jsp-сценариев.

Тема 6. Ввод и вывод.

лабораторная работа (4 часа(ов)):

Выполнение лабораторной работы "Создание web-приложение с помощью технологии GWT". Изучается архитектура GWT-приложений. Лабораторная работа заключается в разработке web-приложения в сервис-ориентированной архитектуре GWT, включающей несколько обращений к сервисам.

Тема 7. Java Foundation Classes (JFC).

лабораторная работа (6 часа(ов)):

Компоненты JFC: AWT, Java2D, Swing и т.д. Основные свойства Swing: компоненты, контейнеры, панели. Основные компоненты (Buttons, Labels, Text fields, Text areas, Check boxes, Radio buttons, Drop-down lists, List boxes, Tabbed panes, Menus, Message Boxes, Dialog Boxes). Обработка основных событий. Иконки и изображения. Layers, Panels, использование Layout Managers. Модель обработки событий. Создание окон, создание меню. Swing компоненты JTree, JTable, JSlider, JProgressBar.

Тема 8. Функциональное программирование. Лямбда-выражения.

лабораторная работа (2 часа(ов)):

Функциональные интерфейсы и их синтаксис. Типы ссылок. Присвоение лямбда-выражения. Синтаксис лямбда-выражения. Ключевые моменты понятия лямбды. Примеры использования лямбда-выражения. Зачем нужны лямбды. Ссылки на методы. Ссылки на конструктор. Область действия переменной. Захват значений в лямбда-выражении.

Тема 9. Многопоточные приложения

лабораторная работа (2 часа(ов)):

Многопоточность в Java. Понятие потока, исполнители, получение значений из потоков, потоки-демоны, присоединение к потоку, взаимодействие потоков. Класс Thread и интерфейс Runnable. Создание потока. Завершение потока. Приоритеты потоков. Синхронизация. Взаимодействие потоков. Методы notify(), wait(), notifyAll() java.util.concurrent ? библиотека для многопоточного программирования. Concurrent Collections. Callable и Future.

Тема 10. Работа с базами данных средствами JDBC.

лабораторная работа (4 часа(ов)):

JDBC и его архитектура, конфигурирование. Подключение драйверов. Важнейшие интерфейсы и их функциональность: DriverManager, Driver, Connection, Statement, Metadata, ResultSet. Структура стандартной программы обработки БД. Выполнение операторов SQL, анализ исключений, транзакции. java.sql.PreparedStatement и java.sql.CallableStatement. ORM и Hibernate (библиотека).

Тема 11. Система сборки Maven

лабораторная работа (2 часа(ов)):

Основные инструменты для сборки на платформе Java. Структура проекта Maven. Основные понятия Maven: POM (Project Object Model), зависимости, Plugins (плагины), Artefact (артефакт), Repository (репозиторий), Coordinates (координаты), Archetype (архетип). Жизненный цикл сборки: фазы сборки. Основные преимущества Maven.

Тема 12. Возможности Stream API и их использование.

лабораторная работа (2 часа(ов)):

Способы создание стримов. Классический способ: Создание стрима из коллекции, создание стрима из значений, создание стрима из массива, создание стрима из файла (каждая строка в файле будет отдельным элементом в стриме, создание стрима из строки), с помощью Stream.builder, создание параллельного стрима, создание бесконечных стрима с помощью Stream.iterate, создание бесконечных стрима с помощью Stream.generate . Методы работы со стримами: конвейерный и терминальный. Примеры использования различных стримов.

Тема 13. Фреймворк Spring.

лабораторная работа (2 часа(ов)):

Что такое framework Spring. Модули Spring. Внедрение зависимостей, модули Beans и Core. Использование контейнера. Именованные бины и старт контекста. Внутренние бины. Инициализация и удаление. Возможности контекста. Именованные бины и старт контекста. Внедрение через property. Внедрение коллекций. Конфигурация с помощью аннотаций.

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Введение. Особенности языка Java.	7	1-2	подготовка к тестированию	2	Тестирование
2.	Тема 2. Конструкции языка. Примитивные и ссылочные типы данных.	7	3-4		4	Компьютерная программа
3.	Тема 3. ООП в Java. Пакет java.lang.	7	5-7		2	Компьютерная программа
4.	Тема 4. Коллекции.	7	8-9		2	Компьютерная программа
5.	Тема 5. Исключения. Модульное тестирование.	7	10-12		2	Компьютерная программа
6.	Тема 6. Ввод и вывод.	7	13-18		4	Компьютерная программа
7.	Тема 7. Java Foundation Classes (JFC).	7			6	Компьютерная программа
8.	Тема 8. Функциональное программирование. Лямбда-выражения.	7			2	Компьютерная программа
9.	Тема 9. Многопоточные приложения	7			2	Компьютерная программа
10.	Тема 10. Работа с базами данных средствами JDBC.	7			2	Компьютерная программа

№	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
11.	Тема 11. Система сборки Maven	7			2	Компьютерная программа
12.	Тема 12. Возможности Stream API и их использование.	7			2	Компьютерная программа
13.	Тема 13. Фреймворк Spring.	7			4	Компьютерная программа
	Итого				36	

5. Образовательные технологии, включая интерактивные формы обучения

Основные образовательные технологии в рамках курса "Программирование на Java" ориентированы на формирования практических навыков разработки Java-приложений. Поэтому все аудиторные занятия являются лабораторными. Ряд занятий имеют форму дискуссии и обсуждения теоретических аспектов разработки приложений некоторых архитектур. Эти занятия также включают совместную со студентами разработку некоторого учебного демонстрационного проекта. Важно, что студенты участвуют в этой разработке и видят все промежуточные состояния программы и принимают решения о способах ее развития. Большая часть занятий проводится в форме выполнения индивидуальных заданий студентами, их консультации с преподавателем, демонстрации и презентации работы программ.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Тема 1. Введение. Особенности языка Java.

Тестирование , примерные вопросы:

Вопрос 1. Каков будет результат выполнения программы `public class Test { public Test() { } public static void main(String[] args) { Test test = new Test(); int i = 5; while(i = 5){ System.out.println(i++); } }` Варианты ответов 1. а) Компилятор выдаст сообщение об ошибке в строке 8 2. б) На консоль будут последовательно выведены значения 01234 3. в) На консоль будут последовательно выведены значения 43210 4. г) Программа откомпилируется, но на консоль ничего выведено не будет

Вопрос 2. Приведенная ниже программа должна вывести на консоль Hello World!

Выберите строки, которые нужно модифицировать в программе, что бы получить правильный результат. `public class Test { public Test() { } public static void main(String[] args) { Test test = new Test(); String [] arr = {"H","e","l","l","o"," ","w","o","r","l","d","!"}; String result = ""; int i= 0; for(;;){ . result += arr[i++]; } System.out.println(result); }` Варианты ответов 1. а) Заменить строку 9 на `for (i = 0; i < arr.length ;){` 2. б) Заменить строку 9 на `for(int int i = 0; i < arr.length;){` 3. в) Заменить строку 9 на `for(i = 0; i < arr.length;++){` 4. г) Заменить строку 9 на `for(i = 1; i <= arr.length;++){`

Вопрос 3. Какая строка будет выдана на консоль после выполнения фрагмента кода приведенного ниже. `public class Test { public Test () { } public static void main (String [] args) { int i, j; lab : for (i = 0; i < 6; i++){ for (j = 3; j > 1; j--){ if(i == j){ System.out.println(" " + j); break lab; } } } }` Варианты ответов 1. а) 2345 2. б) 234 3. в) 3 4. г) 2

Вопрос 4. Какой результат следует ожидать при компиляции и запуске приведенного кода: `String str=new String("Java"); int i=1; char j=3; System.out.println(str.substring(i,j));` Варианты ответов 1. а) Выведено: Ja 2. б) Выведено: av 3. в) Выведено: ava 4. г) Ошибка: не существует метода `substring(int,char)`.

Вопрос 5. Какой метод следует использовать, чтобы обнаружить позицию буквы v в строке `str= "Java"`? Варианты ответов 1. а) `mid (2, str)` 2. б) `str.charAt (2)` 3. в) `str.indexOf ('v')` 4. г) `indexOf (str,'v')`

Вопрос 6. Что будет выведено в результате компиляции и запуска следующего кода: `String str =" ava "; char ch ='J'; ch += str ; System. out. println (ch);` Варианты ответов 1. а) Java 2. б) ava 3. в) avaJ 4. г) J 5. д) Ошибка во время компиляции.

Вопрос 7. Что будет результатом компиляции и выполнения следующего кода? `StringBuffer s= new StringBuffer("You Java"); s.insert(2, "like "); System.out.print(s);` Варианты ответов 1. а) Youlike u Java 2. б) You like Java 3. в) Ylike ou Java 4. г) You Java like 5. д) Ошибка компиляции: метод `insert()` не объявлен для класса `String-Buffer`

Вопрос 8. Что будет выведено при попытке компиляции и запуска этой программы: `public class Quest6 { public static void main(String[] args){ int a[] = new int[]{1,2,3,}; System.out.print(a[1]); } }` Варианты ответов 1. а) Ошибка компиляции: не определен размер массива 2. б) Ошибка времени выполнения 3. в) Выведено 1 4. г) Выведено 2 5. д) Ошибка компиляции: неправильная инициализация

Вопрос 9. Что будет выведено при попытке компиляции и запуска программы? `public class Quest8{ static int j=2; public static void result(int i){ i *= 10; j += 2; } public static void main(String[] args){ char i = ?1?; result(i); System.out.println(i+" "+j); } }` Варианты ответов 1. а) 1 2 2. б) 10 2 3. в) Ошибка: параметр метода `result()` не сочетается с передаваемой переменной 4. г) 10 4 5. д) 1 4

Тема 2. Конструкции языка. Примитивные и ссылочные типы данных.

Компьютерная программа , примерные вопросы:

1. Ввести n строк с консоли, найти самую короткую строку. Вывести эту строку и ее длину.
2. Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания их длин, а также (второй приоритет) значений этих их длин.
3. Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше средней, также их длины.
4. В каждом слове текста k-ю букву заменить заданным символом. Если k больше длины слова, корректировку не выполнять.
5. В русском тексте каждую букву заменить ее номером в алфавите. В одной строке печатать текст с двумя пробелами между буквами, в следующей строке внизу под каждой буквой печатать ее номер.
6. Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.
7. Из текста удалить все слова заданной длины, начинающиеся на согласную букву.
8. В тексте найти все пары слов, из которых одно является об?ращением другого.

Тема 3. ООП в Java. Пакет java.lang.

Компьютерная программа , примерные вопросы:

1. Найти и напечатать, сколько раз повторяется в тексте каждое слово. 2. Найти, каких букв, гласных или согласных, больше в каждом предложении текста. 3. Выбрать три разные точки заданного на плоскости множества точек, составляющие треугольник наибольшего периметра. 4. Найти такую точку заданного на плоскости множества точек, сумма расстояний от которой до остальных минимальна. 5. Выпуклый многоугольник задан на плоскости перечислением координат вершин в порядке обхода его границы. Определить площадь многоугольника. 6. Ввести строки из файла, записать их в стек. Вывести строки в файл в обратном порядке. 7. Ввести число, занести его цифры в стек. Вывести в число, у которого цифры идут в обратном порядке. 8. Сложить два многочлена заданной степени, если коэффициенты многочленов хранятся в объекте HashMap. 9. Создать стек из элементов каталога. 10. Не используя вспомогательных объектов, переставить отрицательные элементы данного списка в конец, а положительные - в начало этого списка. 11. Организовать вычисления в виде стека. 12. Выполнить попарное суммирование произвольного конечного ряда чисел следующим образом: на первом этапе суммируются попарно рядом стоящие числа, на втором этапе суммируются результаты первого этапа и т.д. до тех пор, пока не останется одно число. 13. Задать два стека, поменять информацию местами. 14. Определить класс Stack. Объявить объект класса. Ввести последовательность символов и вывести ее в обратном порядке. 15. Умножить два многочлена заданной степени, если коэффициенты многочленов хранятся в списках.

Тема 4. Коллекции.

Компьютерная программа , примерные вопросы:

Задача 1. Создать коллекции HashSet, TreeSet, LinkedHashSet. Сложить в них порядка 50 чисел из заданного диапазона. Вывести на печать. Создать еще одну коллекцию HashSet из явно указанных элементов (одной операцией). Проверить, содержит ли ее TreeSet. Удалить из TreeSet элементы, входящие в последнюю коллекцию. Результаты вывести на печать. Задача 2. Построить диаграмму распределения случайных чисел, т.е. сосчитать количество выпадений каждого выдаваемого генератором случайных чисел значения из заданного диапазона. Общее количество не менее 100000. (Предварительно построить оболочку для числа вхождений). Подобрать нужную структуру данных. Задача 3. Вычислить сколько раз каждая буква встречается в заданном тексте. Задача 4. Имеется в виду отдел кадров некоторого предприятия, выпускающего некий вид продукции. Сотрудники делятся на руководителей и исполнителей. Сотрудник имеет: индекс, ФИО, должность, возраст, оклад, стаж работы, продолжительность отпуска. Руководитель имеет: рейтинг популярности у сотрудников. Исполнитель имеет: процент брака при изготовлении продукции. Действия ОК: 1. Поступает из вышестоящей организации список сотрудников на сокращение в виде их индексов. Сокращают согласно списку. 2. Для отчета находят самого старого, самого молодого, долго работающего, мало работающего. Данные отправляют на сервер. Сервер сортирует сотрудников по возрасту и пересылает обратно клиенту. Клиент выводит всю коллекцию на консоль. Общие требования: 1. Использовать коллекции, компараторы, итераторы. 2. Числовые данные задавать случайными числами. Задача 5. Сложить два многочлена заданной степени, если коэффициенты многочленов хранятся в объекте HashMap. Задача 6. Определить класс Set на основе множества целых чисел, n = размер. Создать методы для определения пересечения и объединения множеств. Задача 6. Имеется в виду отдел кадров некоторого предприятия, выпускающего некий вид продукции. Сотрудники делятся на руководителей и исполнителей. Сотрудник имеет: индекс, ФИО, должность, возраст, оклад, стаж работы, продолжительность отпуска. Руководитель имеет: рейтинг популярности у сотрудников. Исполнитель имеет: процент брака при изготовлении продукции. Действия ОК: 1. Поступает из вышестоящей организации список сотрудников на сокращение в виде их индексов. Сокращают согласно списку. 2. Для отчета находят самого старого, самого молодого, долго работающего, мало работающего. Данные отправляют на сервер. Сервер сортирует сотрудников по возрасту и пересылает обратно клиенту. Клиент выводит всю коллекцию на консоль. Общие требования: 1. Использовать коллекции, компараторы, итераторы. 2. Числовые данные задавать случайными числами.

Тема 5. Исключения. Модульное тестирование.

Компьютерная программа , примерные вопросы:

Задание. Контейнеры, IO потоки, классы Object и String. Написать программу, которая будет принимать в качестве аргумента имя текстового файла, и выводить файл с колонками: 1. Слово. 2. Частота. 3. Частота (в %). Файл должен быть упорядочен по убыванию частоты, то есть самые частые слова должны идти в начале. Разделителями считать все символы кроме букв и цифр. Методические указания: - Использовать класс `java.lang.StringBuilder` для построения слов. - Для чтения из файла удобно использовать: `java.io.InputStreamReader`, например: `Reader reader = null; try { reader = new InputStreamReader(new FileInputStream("FILE NAME")); //read the data here } catch (IOException e) { System.err.println("Error while reading file: " + e.getLocalizedMessage()); } finally { if (null != reader) { try { reader.close(); } catch (IOException e) { e.printStackTrace(System.err); } } }` - Для определения класса символа использовать метод `Character.isLetterOrDigit`. - Для хранения статистики в памяти можно использовать одну из реализаций интерфейса `java.util.Set`, который должен будет хранить объекты специального класса. Данный класс должен содержать слово и счётчик. В случае использования `java.util.HashSet` класс также должен реализовать методы `equals`, `hashCode`. Теоретические сведения: Контейнеры стандартной библиотеки расположены в пакете `java.util`. IO классы (потоки ввода-вывода) располагаются в пакете `java.io`. Основные интерфейсы: 1. `Set` - множество без дубликатов и без доступа по индексу. 2. `Map` - множество пар ключ-значение, где ключи не повторяются. Их основные реализации: 1. `HashMap`, `HashSet` - реализации на основе функции `hashCode`. 2. `TreeMap`, `TreeSet` - реализация на основе бинарного дерева. Ключи (элементы) должны реализовывать интерфейс `Comparable`, иначе необходимо передавать в контейнер при его создании объект, реализующий интерфейс `Comparator`. Хранимые в данных контейнерах данные упорядочены. Лучшее время поиска, но большее накладные расходы на вставку, чем на основе функции `hashCode`.

Тема 6. Ввод и вывод.

Компьютерная программа , примерные вопросы:

Задача 1. Создать текстовый файл, считать все байты из файла, не учитывая повторений - отсортировать их по байт-коду в убывающем порядке, вывести на экран, закрыть поток ввода-вывода. Текстовый файл положить в проект на одном уровне вложенности с `src`, чтобы можно было не указывать длинный путь, а просто имя файла с его расширением. Задача 2. На вход программы подается 2 файла. Первый содержит английский текст. Второй ? строки вида: Строка1 = Строка2. Где строка1 ? слово на английском языке, а строка2 ? его перевод на русский. Вывести на экран дословный перевод текста из первого файла с помощью перевода из второго, который нужно хранить в `Map`. Задача 3. Откройте текстовый файл так, чтобы вы смогли прочесть его построчно. Читайте каждую строку, как `String`, и поместите этот объект `String` в `LinkedList`. Распечатайте все строки из `LinkedList` в обратном порядке Задача 4. Создайте класс, называемый `SortedDirList` с конструктором, который принимает информацию о пути к файлу и строит хранящийся список директории из файлов по этому пути. Создайте два перегруженных метода `list()`, которые будут производить либо полный список, или подмножество из списка, основываясь на аргументе. Добавьте метод `size()`, который принимает имя файла и возвращает размер этого файла.

Тема 7. Java Foundation Classes (JFC).

Компьютерная программа , примерные вопросы:

Задание. Шаблон проектирования ?MVC?, графический интерфейс пользователя (GUI).
Вариант ♦1 (Тетрис). Постановка задачи Написать аналог игры ?Тетрис? (—TetrisI). Набор фигур стандартный - все возможные вариации связанных многоугольников, составленных из 4-х квадратов. Архитектура программы должна быть основана на паттерне MVC (Mode-View-Controller). Требования к программе 1. Игра должна поддерживать таблицу рекордов. 2. Пользователю должны быть доступны команды: Exit, About, New Game, High Scores. Вариант ♦2 (Сапёр). Постановка задачи Написать аналог игры ?Сапёр? (—MinesweeperII) из состава стандартных программ для Windows OS. Архитектура программы должна быть основана на паттерне MVC (ModeView-Controller). Программа должна иметь два интерфейса - текстовый и графический, причем оба интерфейса должны использовать одну и ту же игровую модель. Т.е. классы данных и логики должны быть одинаковые для текстового и графического интерфейсов. Требования к программе: 1. Размер поля и количество мин можно изменить. По умолчанию поле размером 9x9 и количество мин 10. 2. Игра должна поддерживать таблицу рекордов. 3. Пользователю должны быть доступны команды: Exit, About, New Game, High Scores. 4. Отчет времени должен быть реализован отдельным потоком. Реализация текстового UI 1. Команды пользователя вводятся с консоли, ячейки нумеруются от нуля 2. После каждого хода игрока все игровое поле распечатывается на экран целиком Реализация графического UI 1. Мины и флажки отображать с помощью картинок. 2. При формировании окна игры использовать класс LayoutManager. Для расположения элементов на игровой панели рекомендуется использовать класс GridBagLayout. Для расположения ячеек поля рекомендуется использовать класс GridLayout. Методические указания: - Для реализации пользовательского интерфейса использовать библиотеку Swing (классы из пакета javax.swing.*). - Работа с компонентами пользовательского интерфейса (классами библиотеки Swing) должна проходить только из UI потока. - Для отображения диалоговых окон рекомендуется использовать класс JOptionPane.

Тема 8. Функциональное программирование. Лямбда-выражения.

Компьютерная программа , примерные вопросы:

Задача 1. Создать коллекцию музыкальных инструментов, реализующих интерфейс, содержащий информацию о названии инструмента, стоимости инструмента и его звуковых способностях. Создать интерфейс, который по номеру (текстовое поле) инструмента выводит его название (текстовое поле), специальные кнопки выводит на консоль стоимость и звучание. Также реализовать возможность получения номера инструмента, его звучание и стоимость по названию. Если такого инструмента нет, то вывести сообщение об ошибке. Задача 2. Директор автобусного парка решил обновить подвижной состав, заменив 5 самых старых автобусов новыми. При выборе новых автобусов он исходил из минимальной стоимости имеющихся в продаже автобусов. Осуществить эти операции по обновлению автобусного парка. При выезде на работу механик дает разрешение на выезд каждому автобусу. Разрешение дается при полном бензобаке (емкость ? 50 л). Если бензобак не полон, то требуется дозаправка автобуса. Каждый раз добавляется 0.1 объема бензобака. Информация о готовности каждого из автобусов помещается в текстовом поле. Заправка осуществляется с помощью нажатия соответствующей кнопки. Уровень наполненности бензобака показывает индикатор (?градусник?)

Тема 9. Многопоточные приложения

Компьютерная программа , примерные вопросы:

Задание. Многопоточное программирование. ?Эмулятор работы фабрики? 1. Постановка задачи Напишите приложение, эмулирующее работу фабрики по сборке автомашин. Машина состоит из 3-х частей: кузов, двигатель и аксессуары. Машину надо собрать и отвезти на склад, откуда она поступает дилерам. 2. Требования к программе 1. Все склады имеют определенный размер, который нельзя превышать. Размеры складов, количество сборщиков, поставщиков и дилеров задаются в конфигурационном файле. Приложение предоставляет графический интерфейс (библиотека Swing), где можно смотреть основные параметры работы фабрики и контролировать процесс. 2. Каждый сборщик, поставщик и дилер должен работать в отдельном потоке. Для синхронизации и ожидания событий необходимо использовать мониторы синхронизации (notify(), notifyAll(), wait()). Наличие процедуры ожидания в виде цикла автоматически ведет к непринятию задания. Каждая деталь - это отдельный объект. . 3. Потоки, которые представляют поставщиков деталей, поставляют одну деталь раз в N миллисекунд. Если какой-то склад деталей полон, то поставщик ожидает освобождения места для деталей (используя методы wait(), notify()). Скорость работы поставщиков определяется 3-мя ползунками (для каждого типа деталей). Должно отображаться кол-во деталей на каждом из складов в текущий момент и кол-во деталей, произведенных поставщиками (для поставщиков аксессуаров можно общий показывать). 4. Потоки, которые представляют дилеров, запрашивают со склада готовой продукции 1 машину в M миллисекунд. Скорость запрашивания машин можно регулировать ползунком в интерфейсе окна. Интерфейс также должен показывать кол-во произведенных машин (вообще) и кол-во машин на складе в данный момент. При отправке машины дилеру информация о покупке должна писаться в лог работы фабрики (в файл) в виде строки: <Time>: Dealer <Number>: Auto <ID> (Body: <ID>, Motor: <ID>, Accessory: <ID>) Включение/отключение лога контролируется с помощью специального параметра в конфигурационном файле. 5. Поток контроллера склада готовой продукции просыпается при любом отправке машины со склада продукции. Он анализирует состояние склада и передает запрос на изготовление новых машин (в случае необходимости) на фабрику. 6. На фабрике работает несколько потоков (сборщиков) в рамках ThreadPool. Задачами для ThreadPool являются запросы на создание новых машин (от контроллера склада готовых изделий). При выполнении такой задачи сборщик должен взять по одной детали, необходимой для сборки машины, с соответствующих складов. Если на складе нет нужной детали, то поток ждет поставки. Собирая новую машину, рабочий создает новый объект и с помощью всех необходимых объектов, представляющих детали. После этого объект отправляется на склад готовой продукции. Если склад полон, то рабочий ждет освобождения места для новой машины. Интерфейс должен отображать, сколько всего было сделано машин и сколько задач еще ждут исполнителя (в очереди задач ThreadPool). 7. Конфигурационный файл должен предоставлять настройки для задания вместимости всех складов и количестве всех типов потоков. Примерный список параметров в конфигурационном файле (просьба использовать свои имена): StorageBodySize=100 StorageMotorSize=100 StorageAccessorySize=100 StorageAutoSize=100 AccessorySuppliers=5 Workers=10 Dealers=20 LogSale=true 3. Особенности реализации 1. Классы объектной модели (деталь, склад, поставщик, сборщик и т.д.) не должны зависеть от библиотеки графического интерфейса SWING. 2. Пул потоков (ThreadPool) нужно реализовать в отдельном пакете. ThreadPool должен выполнять абстрактные задачи и не зависеть от реализации (задача на сборку автомашины в фабрике). 3. Программа должна корректно завершаться. При получении сигнала о закрытии окна все потоки должны корректно прерываться, и лог-файл закрываться.

Тема 10. Работа с базами данных средствами JDBC.

Компьютерная программа , примерные вопросы:

Построение собственной базы данных и демонстрация возможностей ее обработки
Варианты заданий
1 База данных ?Студенческая библиотека?. Ориентировочные таблицы: 2 База данных ?Страховая фирма?. Ориентировочные таблицы: ?Виды страховок?, ?Клиенты\объекты?, ?Страховая деятельность?
3 База данных ?Агентство недвижимости?. Ориентировочные таблицы: ?Объекты недвижимости?, ?Продажи?, ?Покупки?.
4 База данных ГИБДД (Государственная инспекция безопасности дорожного движения)
5 База данных ?Деканат ВУЗа?. Ориентировочные таблицы: ?Список студентов?, ?Список предметов?, ?Сессия?
6 База данных отдела кадров производственного предприятия. Ориентировочные таблицы: ?Сотрудники?, ?Штатное расписание?, ?Отделы?, ?Цеха?.
7 База данных фирмы покупки и продажи автомобилей Ориентировочные таблицы: ?Продажи?, ?Покупки?, ?Автомобили?
8 База данных ?Гостиница? Ориентировочные таблицы: ?Номера?, ?Счета?, ?Клиенты?
9 База данных ?Расчет квартплаты ТСЖ? Ориентировочные таблицы: ?Список жильцов?, ?Оплаты?, ?Тарифы?.
10 База данных ?Железнодорожные кассы? Ориентировочные таблицы: ?Продажи?, ?Посадочные места?, ?Направления?
11 База данных ?Авиапассажирские перевозки? Ориентировочные таблицы: ?Рейсы?, ?Самолеты?, ?Продажи?
12 База данных музея. Ориентировочные таблицы: ?Экспонаты?, ?Авторы?, ?Экспозиции?.
13 База данных ?Спортивные комплексы района? Ориентировочные таблицы: ?Нормативы?, ?Спортсмены?, ?Соревнования?
14 База данных ?Экзаменационная сессия?. Ориентировочные таблицы: ?Предметы?, ?Оценки?, ?Студенты?
15 База данных ?Турагентство?. Ориентировочные таблицы: ?Туры?, ?Продажи?,
16 База данных Аптека?. Ориентировочные таблицы: ?Товары?, ?Поставщики?, ?Продажи?
17 База данных ?Сборка и реализация компьютеров?. Ориентировочные таблицы: ?Продукция?, ?Клиенты?, ?Заказы?.
18 База данных Продуктовые магазины района Ориентировочные таблицы: ?Продажи?, ?Отделы?, ?Товары?.
19 База данных больницы (одного отделения). Ориентировочные таблицы: ?Больные?, ?Диагнозы?, ?Врачи?
20 База данных ?Видеотека?. Ориентировочные таблицы: ?Артисты?, ?Фильмы?, ?Продажи?

Тема 11. Система сборки Maven

Компьютерная программа , примерные вопросы:

Задача 1. Создать связный список из элементов типа Pet (дом. жив.). Провести следующую обработку: 1) Создать коллекцию. 2) Вывести на печать всю коллекцию как объект. 3) Вывести на печать первого и последнего. 4) Убрать последнего. Сообщить об этом и кого убрали. 5) Вывести на печать результат. 6) Убрать первого. Сообщить об этом и кого убрали. 7) Вывести на печать результат. 8) Найти и удалить головной элемент (poll). Сообщить об этом и кого убрали. 9) Добавить новый элемент в голову. 10) Вывести на печать результат. 11) Добавить специфический элемент в конец коллекции (offer). 12) Вывести на печать результат. 13) Добавить еще один специфический элемент в конец коллекции (offer). 14) Вывести на печать результат.
Задача 2. Необходимо создать класс, содержащий в себе данные о человеке: идентификационный номер, фамилию, имя, отчество, возраст. Имеется коллекция людей, состоящая из преподавателей и студентов. У преподавателей есть информация о кафедре, на которой он работает, а у студента - номер группы. Создать такую коллекцию, заполнив ее данными (не менее 3 преподавателей и 5 студентов). Вывести на экран отсортированный список (сортировка по фамилии, имени и отчеству) в следующем виде: 1. Иванов И.И. - студент (группа: 09-411) 2. Петров П.П. - преподаватель (кафедра: прикладной математики) Также сообщить сведения о самом молодом и самом старом в этой коллекции человеке.

Тема 12. Возможности Stream API и их использование.

Компьютерная программа , примерные вопросы:

Задача 1. Сделать простейший калькулятор. Каждая кнопка соответствует одной цифре. Цифра с нажатой кнопки отображается в текстовом поле. Следующая цифра отображается в другом текстовом поле. Имеются 3 кнопки для выполнения операций сложения, вычитания и умножения. При нажатии на одну из этих кнопок осуществить соответствующую операцию. Результат вывести на консоль. Задача 2. Создать коллекцию музыкальных инструментов, реализующих интерфейс, содержащий информацию о названии инструмента, стоимости инструмента и его звуковых способностях. Создать интерфейс, который по номеру (текстовое поле) инструмента выводит его название (текстовое поле), специальные кнопки выводит на консоль стоимость и звучание. Также реализовать возможность получения номера инструмента, его звучание и стоимость по названию. Если такого инструмента нет, то вывести сообщение об ошибке. Задача 3. Директор автобусного парка решил обновить подвижной состав, заменив 5 самых старых автобусов новыми. При выборе новых автобусов он исходил из минимальной стоимости имеющихся в продаже автобусов. Осуществить эти операции по обновлению автобусного парка. При выезде на работу механик дает разрешение на выезд каждому автобусу. Разрешение дается при полном бензобаке (емкость - 50 л). Если бензобак не полон, то требуется дозаправка автобуса. Каждый раз добавляется 0.1 объема бензобака. Информация о готовности каждого из автобусов помещается в текстовое поле. Заправка осуществляется с помощью нажатия соответствующей кнопки. Уровень наполненности бензобака показывает индикатор (?градусник?)

Тема 13. Фреймворк Spring.

Компьютерная программа, примерные вопросы:

Задача 1. Начальник автопарка решил усовершенствовать работу своего предприятия. Для этого он создал базу данных машин, входящих в его гараж. В состав его автопарка входит 15 автобусов, 10 грузовиков, 8 ?газелей? для перевозки пассажиров и 12 грузовых ?газелей?. У всех машин кроме назначения имеются номера (индексы) и дата последнего техосмотра. При выезде на работу механик дает разрешение на выезд каждому автомобилю. Разрешение дается, если количество дней после последнего техосмотра не превышает некоторое число: для автобусов - 5, для грузовиков -7, для пассажирских ?газелей? - 3, для грузовых ?газелей? - 6. Информация о проверяемом автомобиле помещается в текстовое поле. Команда проверки на готовность осуществляется нажатием кнопки ?Проверка?. Результат - окрашивание кнопки индикации (или фона) в красный или зеленый цвет в случае не разрешения или разрешения соответственно. Задача 2. На предприятие, осуществляющее сборку некоторых устройств, поступают комплектующие детали: ручка, привод поршня, гайка, шайба, фланец, клапан, штуцер, сальник. Детали характеризуются материалом (сталь, железо, медь, резина, пластмасса и т.п.), весом, стоимостью. Вес и стоимость задаются случайным образом. Предусмотреть меню, в котором есть возможность выбора всех деталей с данным названием, из данного материала, а также сортировки по весу и стоимости. Результаты операций выводить на консоль. Задача 3. Поместить во фрейм пять кнопок. Необходимо, чтобы при нажатии на любую из них кнопки меняли свое расположение. Перемена обеспечивается установкой нового менеджера компоновки.

Итоговая форма контроля

зачет (в 7 семестре)

Примерные вопросы к итоговой форме контроля

Вопросы к зачету:

1. Технология Java, ее возникновение, современное состояние и развитие.
- 2 Структура программы Java. Библиотеки стандартных классов. Основные пакеты. Пример простой программы. Примитивные типы данных.
3. Массивы. Многомерные массивы.
4. Классы. Модификаторы доступа (инкапсуляция). Разграничение доступа в Java. Объявление класса (заголовок + тело). Объявление полей, методов. Объявление конструкторов. Создание объекта. Инициализаторы.
5. Параметры методов. Преобразование и приведение типов. Присвоение значений. Перегрузка методов.

6. Статические элементы. Ключевые слова `this` и `super`. Особенности их применения. Ключевое слово `abstract`.
7. Интерфейсы. Объявление, реализация, применение. Полиморфизм (суть, условия применения).
8. Пакет `java.lang`. Класс `Object` и его основные методы. Класс `Class`
9. Классы-оболочки: `Integer` и другие числовые классы. Их методы. Классы-оболочки `Character`, `Boolean`. Их методы. Класс `Math`.
10. Класс `String`. Конструкторы, методы. Класс `StringBuffer`.
11. Исключения. Иерархия исключений в Java. Генерация исключений. Создание классов исключений.
12. Перехват исключений. Вложенные блоки `try`. Повторная генерация исключений. Блок `finally`. `Try` с ресурсами Как использовать исключения.
13. Классы `Arrays` и `Collections` и их методы. Интерфейсы `Comparator` и `Comparable` и их применение.
14. Интерфейс `List` и классы его реализующие.
15. Интерфейс `Set` и классы его реализующие.
16. Интерфейс `Map` и классы его реализующие.
17. Принципы построения графического интерфейса. Парадигма MVC. Контейнеры. Иерархия компонентов и контейнеров. Компоненты. Атрибуты компонентов. Размер и позиция компонентов.
18. Структура фрейма. Создание простого фрейма. Двумерные фигуры. Иерархия фигур. Определение цвета, выбор шрифта. Вывод графических изображений.
19. Основные менеджеры компоновки.
20. Компоненты фрейма (метки, кнопки и пр.), их описание и использование.
21. Текстовые области и поля. Меню на Java.
22. Многопоточность. Состояние потока. Класс `Thread` и интерфейс `Runnable`. Создание потока. Создание нескольких потоков. Завершение потока.
23. Приоритеты потоков. Синхронизация. Взаимодействие потоков. Потоки-демоны.
24. Система ввода/вывода. Потоки данных. Байтовый ввод/вывод. Иерархия классов входных и выходных потоков. Файловый ввод и вывод. Классы фильтров ввода и вывода. Сериализация.
25. Потоки ввода символов. Консольный ввод и вывод (байтовый и символьный). Файлы произвольного доступа.

7.1. Основная литература:

1. Голощапов А. Л. Google Android: программирование для мобильных устройств. - СПб.:БХВ-Петербург, 2010. - 448 с. Режим доступа: <http://znanium.com/bookread.php?book=351241>
2. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6. - Режим доступа: <http://znanium.com/catalog/product/1020593>
3. Прохоренок Н.А. Основы Java: Самоучитель Учебное пособие / Прохоренок Н.А. - СПб.:БХВ-Петербург, 2017. - 704 с. ISBN 978-5-9775-3785-8 - Режим доступа: <http://znanium.com/catalog/product/978545>

7.2. Дополнительная литература:

1. Машнин Т. С. Современные Java-технологии на практике: Практическое руководство / Машнин Т.С. - СПб.:БХВ-Петербург, 2010. - 560 с. ISBN 978-5-9775-0561-1 - Режим доступа: <http://znanium.com/catalog/product/351236>

2. Хабибуллин, Ильдар Шаукатович .Технология Java : учебно-справочное пособие / И. Ш. Хабибуллин . - Казань : Казанский университет, 2010 . - 210 с.

3.Кубенский А.А. Создание и обработка структур данных в примерах на Java: Пособие / Кубенский А.А. - СПб:БХВ-Петербург, 2015. - 320 с. ISBN 978-5-9775-1820-8 - Режим доступа: <http://znanium.com/catalog/product/940050>

4. Будилов В.А. Интернет-программирование на Java: Пособие / Будилов В.А. - СПб:БХВ-Петербург, 2014. - 698 с. ISBN 978-5-9775-1931-1 - Режим доступа: <http://znanium.com/catalog/product/940239>

7.3. Интернет-ресурсы:

Интернет-портал образовательных ресурсов по ИТ - <http://www.intuit.ru>

Компьютерная энциклопедия - <http://www.computer-encyclopedia.ru>

Официальный сайт Java - <http://www.oracle.com/technetwork/java/index.html>

Сайт по программированию и алгоритмике - <http://algotlist.manual.ru/>

Форум по ИТ - <http://www.citforum.ru>

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Программирование на Java" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Учебно-методическая литература для данной дисциплины имеется в наличии в электронно-библиотечной системе "ZNANIUM.COM", доступ к которой предоставлен студентам. ЭБС "ZNANIUM.COM" содержит произведения крупнейших российских учёных, руководителей государственных органов, преподавателей ведущих вузов страны, высококвалифицированных специалистов в различных сферах бизнеса. Фонд библиотеки сформирован с учетом всех изменений образовательных стандартов и включает учебники, учебные пособия, УМК, монографии, авторефераты, диссертации, энциклопедии, словари и справочники, законодательно-нормативные документы, специальные периодические издания и издания, выпускаемые издательствами вузов. В настоящее время ЭБС ZNANIUM.COM соответствует всем требованиям федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО) нового поколения.

Лекции по дисциплине проводятся в аудитории, оснащенной мультимедийным оборудованием, лабораторные занятия по дисциплине проводятся в компьютерном классе.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 01.03.02 "Прикладная математика и информатика" и профилю подготовки Системное программирование .

Автор(ы):

Байрашева В.Р. _____

"__" _____ 201__ г.

Рецензент(ы):

Аблаев Ф.М. _____

"__" _____ 201__ г.