МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное учреждение
высшего профессионального образования
"Казанский (Приволжский) федеральный университет"
Высшая школа информационных технологий и информационных систем

УТВЕРЖДАЮ

Проректор по образовательной деятельности КФУ

Проф. Таюрский Д.А.

_____ _____ 20___ г.

подписано электронно-цифровой подписью

Программа дисциплины

Информатика Б1.Б.10

Направление подготовки: 09.03.03 - Прикладная информатика
Профиль подготовки: не предусмотрено
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: русский

Автор(ы):

Абрамский М.М.

Рецензент(ы):

Таланов М.О.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Хасьянов А. Ф.
Протокол заседания кафедры No ___ от "____" _____ 201__г

Учебно-методическая комиссия Высшей школы информационных технологий и информационных систем:
Протокол заседания УМК No ____ от "____" _____ 201__г

Регистрационный No 689510117

Казань
2017

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

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

Программу дисциплины разработал(а)(и) старший преподаватель, б/с Абрамский М.М.
Кафедра программной инженерии Высшая школа информационных технологий и информационных систем , Michael.Abramsky@kpfu.ru

1. Цели освоения дисциплины

Цель дисциплины - ввести в круг понятий и задач, возникающих в области программирования, для того, чтобы студенты могли самостоятельно создавать программы, приложения и комплексы программ. Задача курса состоит в выработке у студентов навыков использования языков, технологий и парадигм программирования для создания приложений, ориентированных на различные предметные области.

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

Данная учебная дисциплина включена в раздел " Б1.Б.10 Дисциплины (модули)" основной образовательной программы 09.03.03 Прикладная информатика и относится к базовой (общепрофессиональной) части. Осваивается на 1, 2 курсах, 1, 2, 3, 4 семестры.

Начальные навыки в области программирования и алгоритмизации, а также в знании определенных языков не требуются. Для освоения курса достаточно знаний в области математики, данных в средней школе.

Курс является тесно связанным с курсом дискретной математики, содержит задачи, связанные с областями линейной алгебры.

Освоение данного курса необходимо для изучения всех курсов, связанных с программированием различных задач.

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

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-17 (профессиональные компетенции)	способностью принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла
ПК-2 (профессиональные компетенции)	способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение
ПК-8 (профессиональные компетенции)	способностью программировать приложения и создавать программные прототипы решения прикладных задач

В результате освоения дисциплины студент:

4. должен демонстрировать способность и готовность:

- Знать основные парадигмы программирования (процедурная, объектно-ориентированная, функциональная), основные структуры данных и структуры управления, базовые понятия баз данных и веб-технологий и владеть основными инструментами, реализующими в себе концепцию этих парадигм

- Уметь проектировать и разрабатывать программные системы,

- Выбирать средства программирования, структуры данных, паттерны проектирования, необходимые для оптимального решения поставленной задачи,

- Понимать и анализировать программный код.

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 21 зачетных(ые) единиц(ы) 756 часа(ов).

Форма промежуточного контроля дисциплины экзамен в 1 семестре; экзамен во 2 семестре; зачет в 3 семестре; экзамен в 4 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	Практические занятия	Лабораторные работы	
1.	Тема 1. Введение в алгоритмизацию	1	1-2	2	0	4	Письменное домашнее задание
2.	Тема 2. Введение в язык Java и среду JDK. Используемые типы данных.	1	3-4	2	0	8	Письменное домашнее задание
3.	Тема 3. Управляющие операторы языка.	1	5-6	2	0	8	Письменное домашнее задание
4.	Тема 4. Базовые алгоритмы обработки данных.	1	7-8	2	0	8	Письменное домашнее задание Контрольная работа
5.	Тема 5. Массивы, действия с ними.	1	9-10	2	0	12	Письменное домашнее задание
6.	Тема 6. Символы и строки	1	11-12	2	0	8	Письменное домашнее задание

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
7.	Тема 7. Объектно-ориентированный подход (ООП).	1	13-14	2	0	6	Письменное домашнее задание
8.	Тема 8. Реализация ООП в Java.	1	15-16	2	0	16	Контрольная работа Письменное домашнее задание
9.	Тема 9. Пакеты в Java	1	17-18	2	0	2	Письменное домашнее задание
10.	Тема 10. Интерфейсы	2	1-2	2	0	8	Письменное домашнее задание
11.	Тема 11. Исключения	2	3	2	0	4	Письменное домашнее задание
13.	Тема 13. Generics	2	4-5	2	0	6	Письменное домашнее задание
14.	Тема 14. Классы-коллекции	2	5-12	6	0	30	Контрольная работа Письменное домашнее задание
15.	Тема 15. Ввод и вывод в Java	2	13-15	2	0	12	Письменное домашнее задание
16.	Тема 16. Многопоточность	2	16-18	4	0	12	Контрольная работа Письменное домашнее задание
17.	Тема 17. Регулярные выражения	3	2-3	0	0	6	Письменное домашнее задание
17.	Тема 17. Основы работы веб-приложений.	3	1-2	2	0	6	Письменное домашнее задание

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
18.	Тема 18. Разработка серверных приложений на Java	3	4-7	4	0	16	Контрольная работа Письменное домашнее задание
19.	Тема 19. Событийно-ориентированное программирование.	3	8-9	2	0	6	Письменное домашнее задание
20.	Тема 20. Работа с базами данных	3	10-12	2	0	10	Контрольная работа Письменное домашнее задание
21.	Тема 21. Работа с сетью в Java	3	12-15	2	0	14	Письменное домашнее задание
22.	Тема 22. Рефлексия в Java	3	15-16	2	0	6	Письменное домашнее задание
23.	Тема 23. Паттерны проектирования	3	17-18	4	0	8	Устный опрос
24.	Тема 24. Фреймворк Spring. Инъекция зависимостей	4	1-3	4	10	0	Письменное домашнее задание
25.	Тема 25. Фреймворк Spring. Аспектно-ориентированное программирование.	4	3-5	2	10	0	Письменное домашнее задание
26.	Тема 26. Фреймворк Spring. MVC-приложения.	4	6-10	2	20	0	Контрольная работа Письменное домашнее задание
27.	Тема 27. ORM-модель баз данных. DAO. Фреймворк Hibernate	4	11-16	8	24	0	Контрольная работа Письменное домашнее задание
28.	Тема 28. Тестирование приложений.	4	17-18	2	8	0	Письменное домашнее задание

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
.	Тема . Итоговая форма контроля	1		0	0	0	Экзамен
.	Тема . Итоговая форма контроля	2		0	0	0	Экзамен
.	Тема . Итоговая форма контроля	3		0	0	0	Зачет
.	Тема . Итоговая форма контроля	4		0	0	0	Экзамен
	Итого			72	72	216	

4.2 Содержание дисциплины

Тема 1. Введение в алгоритмизацию

лекционное занятие (2 часа(ов)):

Структура компьютера по фон Нейману. Понятие алгоритма. Языки низкого и высокого уровня. Компилируемые и интерпретируемые языки (примеры, различия)

лабораторная работа (4 часа(ов)):

Структурные и неструктурные блок-схемы. Структурирование блок-схем.

Тема 2. Введение в язык Java и среду JDK. Используемые типы данных.

лекционное занятие (2 часа(ов)):

JVM. Этапы компилирования и исполнения программ на Java. Типы данных в Java и их свойства

лабораторная работа (8 часа(ов)):

Структура класса, метод main. Компиляция и исполнение программ. Примитивные типы данных Java (int, double, float, boolean, char), различия между примитивными и ссылочными типами данных. Приведение типов. Арифметические и логические операции (+,-,*,/,%, &&, ||, !).

Тема 3. Управляющие операторы языка.

лекционное занятие (2 часа(ов)):

Обзор условных и циклических операторов в языке Java. Обзор задач, решаемых через управляющие конструкции.

лабораторная работа (8 часа(ов)):

Задачи на ввод-вывод данных (args, Scanner), условный оператор if, оператор switch, сокращенный if, циклы while и do while, циклы for и for each, операторы break и continue.

Тема 4. Базовые алгоритмы обработки данных.

лекционное занятие (2 часа(ов)):

Подходы к построению алгоритмов. Проектирование алгоритмов TOP-DOWN и BOTTOM-UP.

лабораторная работа (8 часа(ов)):

Принадлежность точки фигуре, заданной системой неравенств, НОД, схема Горнера, вычисление конечных сумм, вычисление бесконечных сумм с некоторой точностью, вычисление последовательностей, заданных рекуррентной формулой, задачи на проверку условий 'для любого' и 'существует'.

Тема 5. Массивы, действия с ними.

лекционное занятие (2 часа(ов)):

Необходимость массивов. Хранение в памяти. Одномерные массивы, алгоритмы обработки массивов, многомерные массивы, ступенчатые массивы

лабораторная работа (12 часа(ов)):

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

Алгоритмы поиска максимума/минимума, алгоритмы сортировки (выбор, обмен, слиянием), многомерные массивы - умножение, сложение матриц, транспонирование матриц.

Тема 6. Символы и строки

лекционное занятие (2 часа(ов)):

Тип char, номер символа в ASCII. Класс String. Объявление, использование (ввод-вывод, конкатенация, методы charAt и length).

лабораторная работа (8 часа(ов)):

Реализация строковых алгоритмов (поиск подстроки, сравнение)

Тема 7. Объектно-ориентированный подход (ООП).

лекционное занятие (2 часа(ов)):

Принципы ООП - абстракция, 3 кита ООП (инкапсуляция, наследование, полиморфизм), классы и объекты.

лабораторная работа (6 часа(ов)):

Проектирование объектов из различных предметных областей

Тема 8. Реализация ООП в Java.

лекционное занятие (2 часа(ов)):

Методы. Объявление и вызов. Параметры. Перегрузка. Классы. Объявление. Члены класса (атрибуты, методы), оператор this, конструктор класса, модификаторы (public, private, protected, по умолчанию, static, abstract). Наследование и полиморфизм. Восходящее преобразование и позднее связывание. Объекты. Создание объекта, оператор new.

лабораторная работа (16 часа(ов)):

Реализация объектов из различных предметных областей

Тема 9. Пакеты в Java

лекционное занятие (2 часа(ов)):

Пакеты в Java. Необходимость. Принципы построения.

лабораторная работа (2 часа(ов)):

Создание пакета. Иерархия. Директивы import и package. Classpath. Компиляция и выполнение классов в командной строке с учетом пакетов

Тема 10. Интерфейсы

лекционное занятие (2 часа(ов)):

Интефейсы и абстрактные классы. Абстрактные методы и абстрактные классы. Интерфейсы, их отличие от абстрактных классов. Интерфейсы и множественное наследование. Интерфейсы как средство адаптации. Клонирование и его реализация в Java. Основные сведения о перечисляемых типах (enums).

лабораторная работа (8 часа(ов)):

Поля в интерфейсах и их инициализация. Вложенные интерфейсы. Интерфейсы и фабрики. Интерфейсы-маркеры. Решение задач на проектирование интерфейсов.

Тема 11. Исключения

лекционное занятие (2 часа(ов)):

Объекты-исключения. Перехват и обработка исключений. Спецификация исключений. Трассировка стека. Повторное возбуждение исключений. Цепочки исключений. Стандартные исключения java и их классификация. Завершение обработки с помощью finally и его использование. Ограничения при использовании исключений. Исключения и конструкторы.

лабораторная работа (4 часа(ов)):

Создание исключений. Использование исключений в работе программ.

Тема 13. Generics

лекционное занятие (2 часа(ов)):

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

Параметризованные типы (Generics). История и цели внедрения параметризованных типов в Java. Простая параметризация. Параметризованные интерфейсы. Параметризованные методы. Параметризация и вложенные классы. Ограничения параметризации. Сосуществование параметризованного и непараметризованного кода. Массивы параметризованного типа. Метасимволы. Преобразования типов и предупреждения.

лабораторная работа (6 часа(ов)):

Разработка параметризованных классов и методов. Реализация параметризованных интерфейсов.

Тема 14. Классы-коллекции

лекционное занятие (6 часа(ов)):

Коллекции объектов. Параметризованные и типизированные контейнеры. Основные понятия. Списки, итераторы, множества, отображения. Collection и Iterator. Foreach и итераторы. Хеширование. Особенности использования hashCode() и equals().

лабораторная работа (30 часа(ов)):

Решение задач на библиотеку java.util. Линейные структуры данных, древовидные структуры данных. Реализация абстрактных типов данных и структур данных в Java

Тема 15. Ввод и вывод в Java

лекционное занятие (2 часа(ов)):

История развития средств ввода/вывода в Java. Новый ввод/вывод (nio). ByteBuffer. Вопросы производительности. Сжатие данных. JAR-файлы. Сериализация, десериализация и их разновидности.

лабораторная работа (12 часа(ов)):

Класс File. Потоки данных: InputStream и OutputStream и их классы реализации. Классы Reader и Writer. java.NIO и задачи на нем.

Тема 16. Многопоточность

лекционное занятие (4 часа(ов)):

Многопоточная архитектура java. Класс Thread. Интерфейс Runnable. Ключевое слово synchronized и совместное использование ресурсов. Ключевое слово synchronized как модификатор методов. Потокобезопасное программирование.

лабораторная работа (12 часа(ов)):

Решение задач на многопоточность. Способы синхронизации в решении задач, проектирование модели Producer-Consumer, Java Concurrent API.

Тема 17. Регулярные выражения

лабораторная работа (6 часа(ов)):

Синтаксис регулярных выражений. Классы Pattern, Matcher, методы compile, match, find, group. Жадные, ленивые регулярные выражения.

Тема 17. Основы работы веб-приложений.

лекционное занятие (2 часа(ов)):

OSI модель, HTTP - клиент, сервер, запросы, параметры, response, HTML, динамическая генерация содержимого

лабораторная работа (6 часа(ов)):

Верстка html страниц. Языки HTML, CSS.

Тема 18. Разработка серверных приложений на Java

лекционное занятие (4 часа(ов)):

Сервлеты как способ реализации взаимодействия клиента и сервера. Редирект. Сессии. Файлы Cookie. Контекст сервлета.

лабораторная работа (16 часа(ов)):

Реализация основных средств http-протокола в сервлетах. Принцип работы JSP и других шаблонизаторов и их базовый синтаксис. Шаблонизаторы JSTL, Freemarker, Velocity.

Тема 19. Событийно-ориентированное программирование.

лекционное занятие (2 часа(ов)):

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

Понятие события, обработчика событий, GUI. Пример реализации GUI на swing.

лабораторная работа (6 часа(ов)):

Основные компоненты пакета Swing. Javascript как браузерный язык. DOM. Ajax-запросы.

Тема 20. Работа с базами данных

лекционное занятие (2 часа(ов)):

Модель данных. Запросы. Нормальные формы БД. ORM.

лабораторная работа (10 часа(ов)):

Драйвер базы данных. JDBC. Классы Connection, Statement, ResultSet. Простейший SQL. ORM-маппинг объектов. Репозиторий.

Тема 21. Работа с сетью в Java

лекционное занятие (2 часа(ов)):

Модель OSI, пакет java.net для реализации сетевых приложений.

лабораторная работа (14 часа(ов)):

URL, Socket, ServerSocket, Datagram. Работа с сетью с помощью NIO.

Тема 22. Рефлексия в Java

лекционное занятие (2 часа(ов)):

Аннотации в Java, необходимость, применение. Основы рефлекции.

лабораторная работа (6 часа(ов)):

Рефлексия - классы Class, Method, Field и другие. Использование рефлексии для метапрограммирования.

Тема 23. Паттерны проектирования

лекционное занятие (4 часа(ов)):

Понятие паттерна. Порождающие паттерны, структурные паттерны и паттерны поведения.

лабораторная работа (8 часа(ов)):

Разбор примеров по порождающим паттернам, структурным паттернам и паттернам поведения.

Тема 24. Фреймворк Spring. Инъекция зависимостей

лекционное занятие (4 часа(ов)):

Что такое Spring. Понятие сильной связанности. Инъекция зависимостей и Inversion of Control. Конфигурация Spring, основные xml-теги и аннотации.

практическое занятие (10 часа(ов)):

Проектирование модели данных предметной области. Выявление сильных связностей. Создание приложений, использующих инъекции зависимостей. Обзор возожностей конфигурирования Spring приложений (beans, конструкторы, set-, get-, коллекции, autowiring)

Тема 25. Фреймворк Spring. Аспектно-ориентированное программирование.

лекционное занятие (2 часа(ов)):

AOP. Необходимость. Spring AOP. Основные понятия.

практическое занятие (10 часа(ов)):

Advice, JoinPoint, PointCut, Proxy-объект. Spring и AspectJ. Реализация через аннотации и через xml.

Тема 26. Фреймворк Spring. MVC-приложения.

лекционное занятие (2 часа(ов)):

Spring MVC. Структура простого приложения.

практическое занятие (20 часа(ов)):

Написание приложений на Spring MVC. Dispatcher, обработка запросов. Основные классы. Конфигурация MVC-приложения.

Тема 27. ORM-модель баз данных. DAO. Фреймворк Hibernate

лекционное занятие (8 часа(ов)):

Модель ORM. Преимущества. Технология JPA. Hibernate. Spring Data JPA.

практическое занятие (24 часа(ов)):

Data Access Objecs в Spring Язык HQL. Фреймворк Hibernate и его возможности. Фремворк Spring Data JPA и его возможности.

Тема 28. Тестирование приложений.

лекционное занятие (2 часа(ов)):

Основные unit-тестирования приложений. Библиотеки тестирования

практическое занятие (8 часа(ов)):

Unit-тестирование приложений. Задачи. Основы Test-Driven Development. Виды тестирования: mock-тестирование, поведенческое тестирование и средства их реализации в Java.

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 в алгоритмизацию	1	1-2	подготовка домашнего задания	8	домашнее задание
2.	Тема 2. Введение в язык Java и среду JDK. Используемые типы данных.	1	3-4	подготовка домашнего задания	8	домашнее задание
3.	Тема 3. Управляющие операторы языка.	1	5-6	подготовка домашнего задания	8	домашнее задание
4.	Тема 4. Базовые алгоритмы обработки данных.	1	7-8	подготовка домашнего задания	6	домашнее задание
				подготовка к контрольной работе	4	контрольная работа
5.	Тема 5. Массивы, действия с ними.	1	9-10	подготовка домашнего задания	16	домашнее задание
6.	Тема 6. Символы и строки	1	11-12	подготовка домашнего задания	8	домашнее задание
7.	Тема 7. Объектно-ориентированный подход (ООП).	1	13-14	подготовка домашнего задания	8	домашнее задание
8.	Тема 8. Реализация ООП в Java.	1	15-16	подготовка домашнего задания	8	домашнее задание
				подготовка к контрольной работе	8	контрольная работа
9.	Тема 9. Пакеты в Java	1	17-18	подготовка домашнего задания	8	домашнее задание
10.	Тема 10. Интерфейсы	2	1-2	подготовка домашнего задания	4	домашнее задание

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
11.	Тема 11. Исключения	2	3	подготовка домашнего задания	4	домашнее задание
13.	Тема 13. Generics	2	4-5	подготовка домашнего задания	4	домашнее задание
14.	Тема 14. Классы-коллекции	2	5-12	подготовка домашнего задания	28	домашнее задание
				подготовка к контрольной работе	4	контрольная работа
15.	Тема 15. Ввод и вывод в Java	2	13-15	подготовка домашнего задания	9	домашнее задание
16.	Тема 16. Многопоточность	2	16-18	подготовка домашнего задания	6	домашнее задание
				подготовка к контрольной работе	4	контрольная работа
17.	Тема 17. Регулярные выражения	3	2-3	подготовка домашнего задания	8	домашнее задание
17.	Тема 17. Основы работы веб-приложений.	3	1-2	подготовка домашнего задания	8	домашнее задание
18.	Тема 18. Разработка серверных приложений на Java	3	4-7	подготовка домашнего задания	18	домашнее задание
				подготовка к контрольной работе	6	контрольная работа
19.	Тема 19. Событийно-ориентированное программирование.	3	8-9	подготовка домашнего задания	8	домашнее задание
20.	Тема 20. Работа с базами данных	3	10-12	подготовка домашнего задания	6	домашнее задание
				подготовка к контрольной работе	6	контрольная работа
21.	Тема 21. Работа с сетью в Java	3	12-15	подготовка домашнего задания	16	домашнее задание
22.	Тема 22. Рефлексия в Java	3	15-16	подготовка домашнего задания	6	домашнее задание
23.	Тема 23. Паттерны проектирования	3	17-18	подготовка к устному опросу	8	устный опрос

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
24.	Тема 24. Фреймворк Spring. Инъекция зависимостей	4	1-3	подготовка домашнего задания	14	домашнее задание
25.	Тема 25. Фреймворк Spring. Аспектно-ориентированное программирование.	4	3-5	подготовка домашнего задания	8	домашнее задание
26.	Тема 26. Фреймворк Spring. MVC-приложения.	4	6-10	подготовка домашнего задания	8	домашнее задание
				подготовка к контрольной работе	4	контрольная работа
27.	Тема 27. ORM-модель баз данных. DAO. Фреймворк Hibernate	4	11-16	подготовка домашнего задания	12	домашнее задание
				подготовка к контрольной работе	4	контрольная работа
28.	Тема 28. Тестирование приложений.	4	17-18	подготовка домашнего задания	4	домашнее задание
	Итого				297	

5. Образовательные технологии, включая интерактивные формы обучения

Парное программирование (один человек набирает код, второй говорит, что писать)

Командное программирование (разделение обязанностей и функций по разработке программ - по модулям или технологиям)

Мастер-классы по определенным темам (повторение практических вещей для преподавателей)

Использование системы контроля версий (технология, используемая во всех компаниях, занимающихся промышленным программированием

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

Тема 1. Введение в алгоритмизацию

домашнее задание , примерные вопросы:

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

1. В Википедии найти код "Hello World" на java. Скомпилировать и запустить с помощью командной строки. Скриншот успешного выполнения прислать на почту. Java-файл с HelloWorld загрузит в репозиторий. 2. Написать на java программу, которая выводит на экран ваше имя, составленное из одного символа. Указание: шрифт в консоли - моноширинный, т.е. каждый символ имеет одну и ту же фиксированную ширину, в т.ч. и пробел. Сделайте такую надпись в блокноте, а потом построчно вставьте ее в операторы s.o.p. в вашей программе. Каждый System.out.println выводит одну строчку, перенося курсор вывода на следующую сроку. 3. На диске C есть папки f1 и f2. На диске D папка g1. Вам надо создать файл 1.txt с текстом "HELLO" в папке f1, переместить его в папку f2. Затем вам надо его скопировать в папку g1 и убедиться, что он там лежит. Напишите команды, которые надо выполнить в командной строке, выполнить описанные действия. ВСЕ действия выполнять только в командной строке. Текстовый файл с командами залить в репозиторий.

Тема 2. Введение в язык Java и среду JDK. Используемые типы данных.

домашнее задание , примерные вопросы:

1. Дан радиус шара. Найти его объем. 2. Вычислить x в степени 100, использовав наименьшее количество операций умножения. Цикл использовать нельзя.

Тема 3. Управляющие операторы языка.

домашнее задание , примерные вопросы:

1. Для цифры k вывести таблицу умножения: 2 x k = ... 3 x k = ... 9 x k = ...

Тема 4. Базовые алгоритмы обработки данных.

домашнее задание , примерные вопросы:

1. Вычислить n! = 1·2·3·...·n для произвольного целого n; 2. Вывести n-е по счету число Фибоначчи. 3. Вычислить по схеме Горнера для произвольного x значение многочлена 4. Разложить введенное число n на множители, вывести их на экран. 5. Проверить среди n числе существование таких, что в них все цифры - четные.

контрольная работа , примерные вопросы:

1. Табулирование кусочно-заданной функции 2. Подсчет бесконечной суммы (ряда) 3. Задача с цифрами числа. 4. Кванторы

Тема 5. Массивы, действия с ними.

домашнее задание , примерные вопросы:

1. Вводятся два массива, представляющие собой подстановку. Пример: (0,1,2,3,4) (4,1,3,0,2) где первая строка - индексы, вторая - элементы массива. Умножить одну подстановку на другую, результат записать в первый массив, результат вывести на экран. 2. Развернуть введенный массив (физически развернуть, а не "вывести в обратном порядке) 3. Заполнить массив размера n следующим образом: a[0]=1, a[1]=-3, a[2]=5, a[3]=-7, и т.д. Вывести полученный массив на экран 4. Вводится массив, затем число k, k < длина массива. Поменять местами блоки в массиве от 0 до k-1 и от k до n-1 Пример: Вход: 1 2 3 4 5 6 7 8 k = 4 Выход: 5 6 7 8 1 2 3 4 5. Реализовать с помощью двумерных массивов умножение матриц.

Тема 6. Символы и строки

домашнее задание , примерные вопросы:

1. Написать программу, которая сравнивает две строки и выводит, какая находится раньше лексикографически. Не использовать compareTo 2. Вводится строка, представляющая собой слова на английском языке, записанные через пробел. Подсчитать, сколько слов начинаются на заглавную букву. 3. Вводится строка, представляющая собой английский текст. Подсчитать и вывести, сколько раз встречается каждая буква английского алфавита в тексте. Не различать регистр ('A' и 'a' - одна и та же буква). Используйте по мере необходимости toUpperCase или toLowerCase, но заменяйте строки правильно. True должен превращаться в False, но true в false. Можете считать, что других вариантов (TRUE, tRuE и т.п.) не будет. 4. Вводится строка, представляющая собой английский текст. Каждое второе вхождение строки "true" заменить на "false". 5. Вводится строка, представляющая собой английский текст. Каждое вхождение слова "mom" заменить на "dad".

Тема 7. Объектно-ориентированный подход (ООП).

домашнее задание , примерные вопросы:

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

1. Создать класс "Преподаватель". Атрибуты - фио, предмет. Методы: конструктор (с параметрами), все set и get методы, а также метод "оценить студента", принимающего в параметры студента, и работающего следующим образом: генерируется случайное число от 2 до 5, выводится строка: "преподаватель ИМЯПРЕПОДАВАТЕЛЯ оценил студента с именем ИМЯСТУДЕНТА по предмету ИМЯПРЕДМЕТА на оценку ОЦЕНКА." Все слова, написанные капслоком, должны быть заменены соответствующими значениями. ОЦЕНКА должна принимать значения "отлично", "хорошо", "удовлетворительно" или "неудовлетворительно" (в зависимости от значения случайного числа). Протестировать методы написанного класса в методе main класса TestClass, используя при этом уже написанный на паре класс Student.

Тема 8. Реализация ООП в Java.

домашнее задание , примерные вопросы:

1. Спроектировать классы "Вектор на плоскости", "Комплексное число", "Рациональная дробь", "Комплексное число из рациональных дробей".

контрольная работа , примерные вопросы:

1. Написание класса по сущностям "Система линейных уравнений", "Трехмерная матрица", "n-мерный вектор". 10 методов, 2 конструктора.

Тема 9. Пакеты в Java

домашнее задание , примерные вопросы:

1. Поместить задания из предыдущей домашней в пакет ru.kpfu.itis, скомпилировать и запустить, учитывая нужный пакет.

Тема 10. Интерфейсы

домашнее задание , примерные вопросы:

1. У любой замкнутой трехмерной фигуры есть площадь поверхности и объем. - Реализуйте интерфейс Measurable3D с заголовками для методов, вычисляющих площадь поверности и объем трехмерной фигуры. - Напишите четыре класса - сфера, прямоугольный параллелепипед, цилиндр и конус. Продумайте для них атрибуты и конструкторы с параметрами. Конструкторы должны выбрасывать исключения IllegalArgumentException, если им на вход подаются отрицательные значения для атрибутов. Классы должны реализовывать методы из Measurable3D. Протестируйте все методы (включая конструкторы) для каждого класса в отдельном классе, корректно подвешивая обработку исключений с помощью try-catch в нужных местах

Тема 11. Исключения

домашнее задание , примерные вопросы:

1. Написать класс MyMath со следующими статическими методами. - int pow(int x, int n) - возведение числа x в степень n. Выбросить исключение IllegalArgumentException, если на вход подаются отрицательный x или отрицательный n (различить случаи для x и n с помощью текста сообщения об ошибке). Выбросить исключение ArithmeticException, если x и n одновременно равны 0. - int sum(int [] arr) - подсчет суммы целочисленного массива. Выбросить исключение ArithmenticException, если длина массива равна 0 (не путайте с массивом, равным null). В другом классе вызвать описанные методы, поместив их в блок try-catch (для каждого метода - отдельный блок). Для каждого метода обработать возникающие в нем исключения. Для второго метода также обработать NullPointerException (т.к. массив может быть не инициализирован. исключения "РЕАЛИЗУЙ МЕНЯ ПОЛНОСТЬЮ" нужно только для того, чтобы вы не забыли реализовать метод. Удалите эту строчку перед тем, как компилировать свое решение.

Тема 13. Generics

домашнее задание , примерные вопросы:

1. Реализовать параметризованный класс AverageComp, T - параметр типа. Параметром должен быть любой класс-наследник Number (отразить при объявлении). Класс содержит массив из объектов класса T, конструктор с параметром и метод double average() - подсчет среднего арифметичекого среди элементов массива. При подсчете воспользоваться тем, что у любого из объектов подклассов Number есть методы intValue, doubleValue, floatValue и другие, с помощью которых можно выполнить нужную конвертацию типов. В методе main класса Main протестировать полученный класс на примере AverageComp<Float> и AverageComp<Integer>.

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

Тема 14. Классы-коллекции

домашнее задание , примерные вопросы:

1. Реализовать класс MyArrayCollection, реализовав его от интерфейса java.util.Collection<Integer>. Не реализовывать методы iterator() и toArray. В качестве реализации внутри взять массив. 2. Реализовать класс MyArrayList, реализовав его от интерфейса java.util.List<Integer> и унаследовав от MyArrayCollection. Не реализовывать методы iterator(), listIterator() и toArray. В качестве реализации внутри взять массив. 3. Реализовать класс MyCollection<T>, реализовав его от интерфейса java.util.Collection<T>. Не реализовывать методы iterator() и toArray. Реализацией внутри берите односвязный линейный список. 4. Реализовать класс MyLinkedCollection, унаследовав его от java.util.Collection<Integer>. Не реализовывать методы iterator() и toArray. В качестве реализации внутри взять односвязный линейный список. 5. Реализовать класс MyLinkedList, реализовав его от интерфейса java.util.List<Integer> и унаследовав от MyLinkedCollection. Не реализовывать методы iterator(), listIterator() и toArray. В качестве реализации внутри взять односвязный линейный список. 6. Реализовать класс MyList<T>, реализовав его от интерфейса java.util.List<T> и унаследовав от реализованного в 11а MyCollection<T>. Не реализовывать методы iterator(), listIterator() и toArray. Реализацией внутри берите односвязный линейный список.

контрольная работа , примерные вопросы:

1. На вход программы подается файл с английским текстом. Составить статистику с помощью Map<Character, Integer> о том, сколько раз каждая буква английского алфавита упоминается в тексте. 2. Из входного файла in.txt вводятся целые числа. Они записываются в ArrayList<Integer>. Упорядочить полученную коллекцию по возрастанию количества цифр в числах (реализовать Comparator<Integer>). Результат сортировки вывести в выходной файл out.txt. Запрещается конвертировать числа в строки для применения к последним length. Все сделать через арифметику цифр числа. Для прохождение по элементам коллекции использовать итератор!

Тема 15. Ввод и вывод в Java

домашнее задание , примерные вопросы:

1. Скопировать один текстовый файл в другой посимвольно с помощью FileReader и FileWriter, а потом построчно с помощью BufferedReader и PrintWriter. Сравнить с помощью nanoTime() время работы обоих способов. 2. Напишите программу, которая создает около 10 объектов класса Graph и сериализует их в каком-нибудь файле. А затем напишите другую программу, которая осуществляет десериализацию, проверяя при этом, четно ли количество вершин или нечетно. Выведите количество графов с четным количеством вершин на экран.

Тема 16. Многопоточность

домашнее задание , примерные вопросы:

1. Вводится массив из n чисел. Вычислить его сумму с помощью двух потоков, каждый из которых считает сумму своей половины массива. 2. Вводится массив из n чисел. Вычислить его сумму с помощью k потоков (число k вводится), каждый из которых считает сумму своего кусочка массива.

контрольная работа , примерные вопросы:

1. Реализовать модель "Producer Consumer" для вычисления корней квадратного уравнения. Producer вычисляет дискриминант, а Consumer основываясь на полученном дискриминанте, принимает решает уравнение. 2. Отсортировать массив с помощью алгоритма слияния, при этом использовав N потоков.

Тема 17. Регулярные выражения

домашнее задание , примерные вопросы:

1. Напишите регулярное выражения для даты и времени в промежутке с 6 марта 1237 12:00 по 27 февраля 1978 21:35 в формате MM/DD/YYYY HH:MM 2. Напишите регулярное выражение для вещественного числа с периодом. Подходят: 0, -6, -0.5, +2, 0,0(64), Не подходят: -0, 001, 0,(35)00, -3,750

Тема 17. Основы работы веб-приложений.

домашнее задание , примерные вопросы:

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

1. Сверстать html-страницу, которая отправляет поисковые запросы к поисковым системам baidu.com, bing.com, yahoo.com, aol.com. Одна страница, четыре формы. Желающие (кто знает js) могут сделать одну форму с radio-кнопками (выбором нужного поискового запроса). 2. На вход java программы поступают строки. Среди них есть правильные запросы и неправильные. Правильные: /getdate Генерирует html-страницу, содержащую текущую дату и время /строка/целоечисло/целоечисло Где строка - либо add, либо mult. Возвращает html-страницу, содержащую результат операции, примененной к целым числам, которые пришли в заголовке запроса. /адрес/search Где адресом может быть один из адресов предыдущей задачи. Возвращает html страницу, возвращающую поисковую форму к заданному ресурсу. html-страницы создавать в той же папке, где находится программа, называть каждую с помощью java.util.Date, чтобы имя у каждой страницы было уникальным. При неправильном запросе должна создаваться страница html с содержимым, информирующем о произошедшей 404й ошибке.

Тема 18. Разработка серверных приложений на Java

домашнее задание , примерные вопросы:

1. Спроектировать интерфейс своего приложения. 2. Спроектировать архитектуру запросов своего приложения. 3. Сверстать все HTML-страницы приложения. 4. Реализовать сервлеты, обеспечивающие работу приложения. 5. Реализовать механизм авторизации с использованием сессий и cookie. 6. Реализовать шаблоны с помощью Free Marker

контрольная работа , примерные вопросы:

1. Реализовать следующее веб-приложение: На странице GET /process есть - текстовое поле ввода (textarea), в который вводится английский текст, - список (любой) для выбора операции - кнопка Process Данное приложение делает анализ текста, возвращая его результат клиенту по result. Операции: - Количество символов - Количество слов - Количество предложений - Количество абзацев (считать примерно) Форма отправляет POST запрос на /process, где происходит обработка текста. Далее происходит редирект на страницу /result, где отображается результат. Способ передачи результата (сессией или параметрами запроса) выбрать самостоятельно.

Тема 19. Событийно-ориентированное программирование.

домашнее задание , примерные вопросы:

1. Реализовать стандартный калькулятор с кнопками. Должны поддерживаться операции +, -, *, / для вещественных чисел.

Тема 20. Работа с базами данных

домашнее задание , примерные вопросы:

1. Спроектируйте архитектуру базы данных вашего приложения. Какие таблицы будут в базе, какие в этих таблицах будут поля (название, тип, первичные и внешние ключи). - Напишите SQL-скрипт, создающий структуру этой таблицы (CREATE). Добавьте его в репозиторий под названием create.sql. - Напишите SQL-скрипт, заполняющий созданную БД тестовыми данными (INSERT). В каждой таблице должно быть не менее 3 строк + не должно быть фиктивных строк (например, студент в students есть, а оценок в results для этого студента нет). Добавьте его в репозиторий под названием data.sql. 2. Подключите базу в написанное клиент-серверное приложение. Обработку БД делайте в концепции паттерна "Репозиторий". Т.е. есть выделенный класс для каждой сущности, скрывающий обращение к базе и выполняющий самостоятельно отображение результата запроса (ResultSet) во то, что можно быстро передать в шаблонизатор.

контрольная работа , примерные вопросы:

1. Вывести имена студентов и названия предметов, которые посещали эти студенты с 1992 по 1998 годы. 2. Вывести, какие студенты посещали занятия преподавателя Horace Slughorn и в какие годы Выполнить запросы в рамках приложения на сервлетах, выводя результат запроса в виде таблицы. Для разных запросов пожалуйста разные сервлеты. Либо используете только JSP, либо только севелеты + шаблонизаторы (Freemarker или Velocity). Работу с БД спрятать в отдельные классы, являющиеся реализацией репозитория для каждой таблицы (которые нужны для запроса).

Тема 21. Работа с сетью в Java

домашнее задание , примерные вопросы:

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

1. Реализовать игру в города между клиентом и сервером. Должно проверяться, что вводимый город начинается на букву, на которую заканчивается предыдущий + то, что город не был до этого назван. Все проверки делаются до отправки, пользователь вводит данные, пока не введет правильные. 2. Реализовать текстовый чат в консоли и в swing-приложении. 3. Найдите в сети страницу, на которой лежат прямые ссылки на pdf или mp3 файлы. Например, страницы википедии о каких-нибудь городах странах содержат много pdf-статистики в приложении. Надо с помощью класса URL считать html-код страницы, в нем с помощью регулярных выражений найти ссылки на все указанные на странице файлы, составить на их основе составить новые экземляры класса URL, с помощью которых требуется скачать все указанные файлы. Имя файла должно совпадать с именем файла в url.

Тема 22. Рефлексия в Java

домашнее задание , примерные вопросы:

1. Создайте аннотацию author с параметром name, аннотируйте ей каждый метод в классах RationalFraction, Vector2D, CompexNumber, указывая в name имя какого-нибудь вашего одногруппника (на все классы не более 5-7 имен). Напишите приложение, использующее рефлексию для сбора статистики: распределите по местам (с 1го до количества имен) ваших одногруппников по следующим критериям: - Средняя длина названия метода (по всем методам одного автора) - Средняя длина имени параметра в методе (по всем параметрам всех методов одного автора) - Количество void-функций - Общее количество параметров в написанной функции

Тема 23. Паттерны проектирования

устный опрос , примерные вопросы:

Рассказать о следующих паттернах - как и для чего они применяются, привести пример кода: Abstract Factory, Adapter, Bridge, Builder, Chain of Responsibility, Command, Composite, Decorator, Facade, Factory Method, Flyweight, Iterator, Memento, Observer, Prototype, Proxy, Singleton, State, Strategy, Template Method

Тема 24. Фреймворк Spring. Инъекция зависимостей

домашнее задание , примерные вопросы:

1. Реализовать приложение с использованием Spring IoC, содержащее минимум 3 интерфейса, 10 классов. Вам надо описать одну из указанных предметных областей. - Университет - Общежитие - ИТИС - Соц.сети - Команда разработчиков Обязательный функционал: - beans - constructor-args - properties - collections (либо <list> либо <util:list>) - init-method - scope - autowiring - annotations (@Component, @Autowired и др.) Для того, чтобы поддерживался и xml-конфигурация, и аннотации, вы можете сделать 2 spring-config

Тема 25. Фреймворк Spring. Аспектно-ориентированное программирование.

домашнее задание , примерные вопросы:

1. Взять приложение с количеством классов не меньшим пяти. а. Добавить логирование (любым способом). В выводе лога писать, какой метод вызван, на каких аргументах и во сколько. б. Перенести по возможности логику обработки исключений, связанных с вызывающим метод классов, в соответствующие Pointcut и Advise.

Тема 26. Фреймворк Spring. MVC-приложения.

домашнее задание , примерные вопросы:

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

1. Реализовать MVC-приложение с использованием Spring, автоматизирующее обращение граждан страны в ее министерство (выберите самостоятельно, какое). Функционал ПОЛЬЗОВАТЕЛЯ: - Страница входа: логин и пароль, после которого происходит переход на домашнюю страницу. Также ссылки "Регистрация" и "Запомнить меня". - Страница регистрации: email, пароль, имя, фамилия - Домашняя страница: дэшборд, содержащий меню с тем, что можно сделать. - Личные сообщения / вопросы граждан - Страница запроса документа - Форма должна содержать не только текстовые поля, но и выпадающие списки, чекбоксы или радио-группы. - Валидация данных требуется на: телефон, email (самые легкий регулярные выражения), серию и номер паспорта, ИНН, СНИЛС и т.п. данные (по количеству цифр), ФИО (только буквы, пробелы и дефис, но не два подряд) + все данные должны быть заполнены! Все отправленные данные сохраняются в БД (о ней дальше). - Генерация документов возможна в форматах html, docx, rtf или pdf - Профиль пользователя - страница отображения информации о пользователе, возможность изменить email и пароль. ФУНКЦИОНАЛ СОТРУДНИКА: - Личные сообщения - по аналогии с пользователем. - Просмотр всех сгенерированных документов. - Профиль сотрудника - страница отображения информации о сотруднике, возможность изменить email и пароль. Требования к технологиям: - Spring AOP (логирование, исключения) - Spring MVC (ну это понятно) - Spring DAO + JDBC / Hibernate (выберите сами) - Unit-тесты для методов из DAO и для генерации документов. При тестировании также использовать Spring IoC (данные для вставки в assert подгружаются из spring-config.xml) - Для авторизации и аутентификации использовать Spring Security. - СУБД - любая (postgres, mysql и т.п.) - Использование CSS и JS опционально (за исключением случаев, если вы выбираете генерацию документов в html, где вам очевидно пригодится CSS)

контрольная работа , примерные вопросы:

Реализовать на Spring MVC: 1. Реализовать форму регистрации на мероприятие какой-либо конференции. Обязательные поля - имя, университет/компания, интересы, почта, телефон Требования: - все поля должны быть заполнены, - интересы должны быть сделаны через textare - email и телефон валидируются. Данные должны обрабатываться и сохраняться в базу данных, фреймворк для работы с базой взять произвольный.

Тема 27. ORM-модель баз данных. DAO. Фреймворк Hibernate

домашнее задание , примерные вопросы:

1. Реализовать для своего веб-приложения работу с БД. Добавляемая информация из документов приложения несет определенные сущности внутри себя - люди (паспорт, номер, серия), адреса, документы, учреждения, устройства, здания, машины и прочее. Поэтому структуру такую, чтобы была продуманная структура с выделенными конкретными сущностями (Человек, Дом, Машина, Учреждение и т.п). Таблицы для 3-х документов должны содержать foreign key на выделенные сущности. Требование: выделить не менее 3 сущностей (помимо трех таблиц для трех документов).

контрольная работа , примерные вопросы:

Реализовать работу с таблицей virtual-school через Spring Data JPA и через Hibernate выполнить следующие запросы (обоими способами): 1. Вывести имена студентов и названия предметов, которые посещали эти студенты с 1992 по 1998 годы. 2. Вывести, какие студенты посещали занятия преподавателя Horace Slughorn и в какие годы

Тема 28. Тестирование приложений.

домашнее задание , примерные вопросы:

1. Протестировать методы классов "Комплексное число", "Вектор на плоскости" 2. Протестировать семестровую работу (веб-приложение) прошлого семестра.

Тема . Итоговая форма контроля

Тема . Итоговая форма контроля

Тема . Итоговая форма контроля

Тема . Итоговая форма контроля

Примерные вопросы к зачету и экзамену:

Билеты генерируются случайным образом по 2 вопроса из следующего списка:

1 СЕМЕСТР

1. Структура компьютера по фон Нейману. Понятие алгоритма. JVM. Этапы компилирования и исполнения программ на Java. Комплирование и выполнение Java-программ с помощью командной строки.

2. Примитивные типы данных Java (int, short, long, byte, double, float, boolean, char), различия между примитивными и ссылочными типами данных. Приведение типов. Арифметические и логические операции (+,-,*,/,%, &&, ||, !). Преобразование типов в при арифметических операциях. Сужение при преобразовании типов

3. Условные и циклические операторы языка. Условные операторы if, switch case, (P ? v1 : v2). Циклические операторы while, do while, циклы for и for each, операторы break и continue.

4. Массивы, действия с ними. Одномерные массивы объявление ссылки и выделение памяти. Обращение к элементу массива. Как пройти по всем элементам массива. Многомерные массивы - объявление, способ пройти по всем элементам. Ступенчатые массивы.

5. Символы и строки. Тип char, номер символа в ASCII. Способ узнать номер. Класс String. Объявление, использование (ввод-вывод, конкатенация, характерные методы). Особенности хранения строковых констант в Java.

6. Методы в java. Объявление и вызов. Правила вызова метода. Возвращаемые значения, return. Параметры, передача по ссылке, по значению. Перегрузка.

7. Классы и объекты в Java. Классы. Структура класса, члены класса. Объявление объекта класса. Объявление ссылки и выделение памяти. Конструктор класса, оператор this.

8. Инкапсуляция. Модификаторы доступа - private, protected, public, по умолчанию. Модификатор static.

9. Наследование и полиморфизм в Java. Правила наследования методов и атрибутов. Оператор super. Переопределение. Модификаторы abstract и final. Позднее связывание. Восходящее преобразование.

10. Пакеты в Java. Создание пакета. Иерархия. Директивы import и package. Classpath. Компиляция и выполнение классов в командной строке с учетом пакетов.

2 СЕМЕСТР

1. Интерфейсы. Объявление, структура. Что могут содержать интерфейсы? Наследование интерфейсов. Правила реализации интерфейса классом. Восходящее преобразование в случае интерфейса.

2. Исключения. Throwable, Exception, Error, RuntimeException. Оператор throw. Использование throws. Примеры известных исключений - ArithmeticException, NullPointerException.Блок try - catch, правила работы. Использование finally. Корректная последовательность обработки исключений

3. Оболочки примитивных типов. Integer, Real, Float, Double, Character, Boolean, Long, Short. Зачем нужны. Автоматическая упаковка. Абстрактный класс Number. Назвать по 3-4 характерных метода у каждого класса.

4. Generics. Цель использования. Параметризация класса. Пример. Параметризация метода. Объявление объекта параметризованного класса. Diamond-оператор. Механизм реализации параметризованных классов. Стирание типов. Последствия для программиста. Неизвестный параметр (wildcard-type). Зачем он нужен? использование extends и super в параметрах.

5. Классы-коллекции. Интерфейсы Collection, Set, List, Queue, Map, их основные особенности и классы коллекции, реализующие их.

6. Java io. InputStream, OutputStream, Reader, Writer и их наследники. Надстройки. Мосты.

7. Threads. Виртуальная параллельность. Реализация потоков с помощью Runnable и Thread. Жизненный цикл потока. Методы start, stop, sleep, yield, join.

8. Передача объектов в поток. Синхронизация и ее проблемы. Модификатор volatile. Способы синхронизации.

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

9. Модель Producer - Consumer. Методы wait, notify, notifyAll.

10. Concurrent API.

3 СЕМЕСТР

1. Регулярные выражения. Синтаксис регулярных выражений. Классы Pattern, Matcher, методы compile, match, find, group. Жадные, ленивые регулярные выражения.

2. Примитивная модель сетевого взаимодействия. Клиент. Сервер. OSI-модель. Сетевой протокол. Протоколы уровней OSI-модули.

3. HTTP-протокол. URL, URI, URN. Запрос (структура), заголовки запроса, стартовая строка, http-методы. POST и GET запросы. Параметры запросов. Response (структура). Заголовки. Код ответа. MIME-типы

HTML, HTML-формы, XML (DTD, XML Schema), JSON.

4. Статические и динамические сайты. Динамическая генерация. MVC-модель.

5. Порты. Веб-сервер. Примеры веб-серверов. CGI. Tomcat и Catalina.

Сервлет (структура, его методы). Жизненный цикл сервлета. HTTP-сервлет. HttpServletReques и его методы. HttpServletResponse (методы, принцип работы). URL-mapping (синтаксис шаблонов). Web.xml.

6. Сборка. Минусы ручной сборки. Декларативный и императивный подход к сборке. Ant. Target, task. Maven. Pom.xml. Зависимость. Деплой. Артефакт. Механизм плагинов. Жизненный цикл сборки. Архетипы. Необходимая структура приложения для Maven.

7. Понятие Stateless в HTTP. Сессия. HttpSession его принцип работы и методы. Авторизация пользователя через сессию. Редирект. Cookie (устройство, атрибуты, как работают механизмы cookie). Реализация сессии через cookie. ServletContext.

8. Шаблонизаторы. JSP. Принцип работы. Jasper. Выражения, объявления, скриптлеты, директивы (синтаксис, назначение). Другие шаблонизаторы - JSTL, Freemarker, Velocity.

9. UI. История развития UI и GUI. Графические оболочки ОС.

Окно (форма). Виджеты (компоненты), Layout. События. Listener. Событийно-ориентированный подход в программировании.

Примеры графических элементов в библиотеке Swing. Принцип разработки приложений на swing. Анонимный класс для Listener.

10. Браузерные приложения. Их преимущество над Desktop-ными. События в браузере. ECMAScript. DOM. BOM. JS. Как происходит работа с компонентами страницы с помощью DOM.

Прототипирование в JS. Использование функций как объекта в JS.

AJAX. Необходимость и преимущества. Примеры использования AJAX (без кода).

11. Реляционные БД. Таблица, строка, столбец. Primary Key, Foreign Key. Избыточность и целостность.

Технология jdbc, работа с базами данных. Драйвер базы данных. Классы Connection, Statement, ResultSet.

12. java.net. URL, Socket, ServerSocket, Datagram. Работа с сетью с помощью NIO.

13. Хардкод и его минусы. Перечисления. Чем фактически являются. Методы перечислений. Конструкторы.

Аннотации. Зачем нужны. Создание собственных аннотаций. Аннотации Retention и Target.

14. Рефлексия. Понятие рефлексии. Классы Class, Method, Annotation, Type и их методы. Как получить class по объекту, классу и строке. Доступ к приватным полям через рефлексию. Вызов метода через рефлексию. Сигнатура метода (общее определение и способ задания через рефлексию).

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ

15. Поведенческие паттерны проектирования: Интерпретатор (Interpreter), Итератор (Iterator), Команда (Command), Наблюдатель (Observer), Посетитель (Visitor), Посредник (Mediator), Состояние (State), Стратегия (Strategy), Хранитель (Memento), Цепочка обязанностей (Chain of Responsibility), Шаблонный метод (Template Method).

16. Порождающие паттерны проектирования: Абстрактная фабрика (Abstract Factory), Одиночка (Singleton), Прототип (Prototype), Строитель (Builder), Фабричный метод (Factory Method).

17. Структурные паттерны проектирования: Адаптер (Adapter), Декоратор (Decorator), Заместитель (Proxy), Компоновщик (Composite), Мост (Bridge), Приспособленец (Flyweight), Фасад (Facade).

4 СЕМЕСТР

1. Spring. Что такое Spring. Понятие сильной связанности. Инъекция зависимостей и Inversion of Control.

2. Конфигурация Spring, основные xml-теги и аннотации.

3. AOP. Зачем нужен. Spring AOP. Основные понятия: Advice, JoinPoint, PointCut, Proxy-объект.

4. Spring и AspectJ. Реализация через аннотации и через xml.

5. ORM - основная концепции. JPA.

6. Основные особенности Spring Data JPA.

7. Основные особенности Hibernate.

8. MVC-архитектура. Spring MVC. Структура простого приложения.

9. Основные особенности и возможности Spring MVC.

10. Основные особенности библиотеки Spring Security.

11. Основы тестирования. Виды тестирования (с примерами).

7.1. Основная литература:

1. Технология разработки объектно-ориентированных программ на JAVA / Васюткина И.А. - Новосиб.:НГТУ, 2012. - 152 с.: ISBN 978-5-7782-1973-1
http://znanium.com/bookread2.php?book=557111

2. Информатика: Учебное пособие / Под ред. Б.Е. Одинцова, А.Н. Романова. - 2-е изд., перераб. и доп. - М.: Вузовский учебник: НИЦ Инфра-М, 2012. - 410 с.: 70x100 1/16. (переплет) ISBN 978-5-9558-0230-5
http://znanium.com/bookread2.php?book=263735

3. Машнин Т. С. Современные Java-технологии на практике. - СПб.: БХВ-Петербург, 2010. - 560 с. - (Профессиональное программирование). - ISBN 978-5-9775-0561-1.
http://znanium.com/bookread2.php?book=351236

7.2. Дополнительная литература:

1. Информатика: Курс лекций. Учебное пособие / Е.Л. Федотова, А.А. Федотов. - М.: ИД ФОРУМ: ИНФРА-М, 2011. - 480 с.: ил.; 60x90 1/16. - (Высшее образование). (переплет) ISBN 978-5-8199-0448-0, 1500 экз.
http://znanium.com/bookread.php?book=204273

2. Численные методы и программирование: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 336 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0333-9, 300 экз.
http://znanium.com/bookread2.php?book=452274

7.3. Интернет-ресурсы:

Online-площадка для изучения языков программирования - https://www.codecademy.com/

Образовательный справочник технических вопросов по программированию и информатике - http://www.quizful.net/test

Руководства по изучению Java и связанных технологий и фреймворков - http://tutorials.jenkov.com/

Сайт фреймворка Spring - статьи и уроки - http://spring-projects.ru/

Статьи и уроки по разработке на языке Java - http://javaxblog.ru/

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Информатика" 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 оборудованием 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.

Проектор в аудитории.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 09.03.03 "Прикладная информатика" и профилю подготовки не предусмотрено .

Автор(ы):
Абрамский М.М. _____
"__" _____ 201 __ г.

Рецензент(ы):
Таланов М.О. _____
"__" _____ 201 __ г.

ЭЛЕКТРОННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА КФУ