


Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Чувашский государственный университет имени И. Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

«УТВЕРЖДАЮ»
Проректор по учебной работе


И.Е. Поверинов

«31» августа 2017 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Кросс-платформенные средства разработки программного обеспечения»

Направление подготовки (специальность) **09.03.01 «Информатика и вычислительная техника»**

Квалификация (степень) выпускника Бакалавр

Профиль (направленность) *Программное обеспечение средств вычислительной техники и автоматизированных систем*

Академический бакалавриат

Чебоксары – 2017

Рабочая программа основана на требованиях Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки 12.01.2016 г. №5.

СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):

старший преподаватель _____  С.В. Галибин

к.т.н., доцент _____  И.А. Обломов

ОБСУЖДЕНО:

на заседании кафедры вычислительной техники «30» августа 2017 г., протокол № 1


заведующий кафедрой _____  А.В. Щипцова


СОГЛАСОВАНО:

Методическая комиссия факультета информатики и вычислительной техники «30» августа 2017 г., протокол № 1

Декан факультета _____  А.В. Щипцова

Директор научной библиотеки _____  Н. Д. Никитина

Начальник управления информатизации _____  И. П. Пивоваров

Начальник учебно-методического управления _____  В. И. Маколов

Оглавление

1. Цель и задачи обучения по дисциплине.....	4
2. Место дисциплины в структуре основной образовательной программы (ООП).....	4
3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП	4
4. Структура и содержание дисциплины.....	5
5. Содержание разделов дисциплины.....	9
6. Образовательные технологии.....	13
7. Формы аттестации и оценочные материалы.....	14
8. Учебно-методическое и информационное обеспечение дисциплины	16
9. Материально-техническое обеспечение дисциплины	17
10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями	18
11. Методические рекомендации по освоению дисциплины	18

1. Цель и задачи обучения по дисциплине

Цель дисциплины – изучение синтаксических конструкций кросс-платформенного языка программирования JAVA, возможностей предоставляемых данным языком программирования в области создания кросс-платформенных программ.

Задачами преподавания дисциплины являются:

ознакомление с синтаксисом языка программирования JAVA и его основными возможностями;

развитие навыков применения основных принципов объектно-ориентированного программирования, таких как: инкапсуляция, наследование и полиморфизм, а так же навыков управлением объектом, его поведением и состоянием;

обеспечение практического опыта определения различных классов и их иерархий, объявления абстрактных классов и интерфейсов;

применение современных подходов к проектированию программ, в том числе использование шаблонов проектирования, ознакомление с основными шаблонами проектирования, наиболее используемых в практическом программировании;

ознакомление со стандартной библиотекой JAVA, включая систему классов для работы с коллекциями, потоками ввода-вывода, библиотекой SWING для создания пользовательских интерфейсов.

2. Место дисциплины в структуре основной образовательной программы (ООП)

Дисциплина реализуется в рамках дисциплин по выбору вариативной части образовательной программы бакалавра.

Изучение данной дисциплины базируется на следующих курсах:

«ЭВМ и периферийные устройства» – знать принципы функционирования ПЭВМ и их составных блоков: процессора, ОЗУ, устройств ввода-вывода. Знать состав программного обеспечения ПЭВМ, различать прикладное и системное программное обеспечение. Знать формат хранения данных, представления данных в различных системах счисления;

«Программирование на языке высокого уровня» – знать языки программирования высокого уровня и владеть методиками использования программных средств для решения прикладных практических задач. Иметь представление о назначении и функционировании компиляторов, способах отладки программ;

«Объектно-ориентированное программирование» – знать основные концепции объектно-ориентированного программирования, такие как: класс, объект, инкапсуляция, наследование и полиморфизм. Иметь практические навыки в создании классов и объектов, разработки системы и иерархии классов;

Дисциплина является необходимой для прохождения преддипломной практики и выполнения выпускной квалификационной работы бакалавра.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП

Процесс обучения по дисциплине направлен на формирование следующей профессиональной компетенции:

способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования (ПК-2).

В результате обучения по дисциплины обучающийся должен (ЗУН):

знать:

- принципы объектно-ориентированного программирования (31),
- способы создания иерархии классов основываясь на базовых классах и интерфейсах (32),
- основные шаблоны проектирования (33)

- основные интерфейсы, реализованные в языке программирования JAVA (34),
- основные коллекции, реализованные в JAVA (35),
- основные классы, предназначенные для решения типовых задач (36);
- основные классы для построения пользовательских интерфейсов с использованием пакета Swing (37);

уметь:

- создавать собственную иерархию классов (У1);
- использование стандартных интерфейсов (У2);
- использовать стандартные коллекции для решения прикладных задач (У3);
- использовать стандартные классы для решения типовых задач (У4);
- создавать пользовательские интерфейсы, используя Swing (У5);

владеть навыками:

- использования возможностей существующих интерфейсов и классов (Н1);
- использования одной из интегрированных средств разработки для написания, компилирования и отладки приложений (Н2);
- проектирования и реализации пользовательских интерфейсов (Н3);

4. Структура и содержание дисциплины

Образовательная деятельность по дисциплине проводится:

- в форме контактной работы обучающихся с педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях (далее – контактная работа);
- в форме самостоятельной работы.

Контактная работа включает в себя занятия лекционного типа, занятия семинарского типа (лабораторные работы), групповые и (или) индивидуальные консультации, в том числе в электронной информационно-образовательной среде.

Обозначения:

Л – лекции, л/р – лабораторные работы, п/р – практические занятия, КСР – контроль самостоятельной работы, СРС – самостоятельная работа студента, ИФР – интерактивная форма работы, К – контроль.

4.1. Содержание дисциплины

Содержание	Формируемые компетенции	Формируемые ЗУН
Раздел 1. Основы языка программирования Java.	ПК-2	З1, У1, Н1, Н2
1.1. История развития языка Java.		
1.2. Основные особенности языка.		
1.3. Объектно-ориентированный подход в программировании на языке Java.		
1.4. Простейшая программа на Java.		
Раздел 2. Типы данных и операторы в Java	ПК-2	З1, У1, Н1, Н2
2.1. Примитивные типы данных в Java		
2.2. Создание и инициализация переменных в Java		
2.3. Область действия переменных		
2.4. Операторы в Java		
Раздел 3. Операторы управления в Java	ПК-2	З1, У1, Н1, Н2
3.1. Операторы ветвления		
3.2. Операторы цикла		
3.3. Операторы перехода		
Раздел 4. Введение в классы и методы в Java	ПК-2	З1, З2, У1, Н1, Н2
4.1. Базовый синтаксис описания классов		
4.2. Переменные ссылки и создание объектов		
4.3. Создание методов в классе, параметры и возвращаемое		

значение		
4.4. Создание конструкторов		
4.5. Сборщик мусора и метод finalize		
Раздел 5. Дополнительные типы данных и операторы	ПК-2	31, 32, У1, Н1, Н2
5.1. Создание массивов		
5.2. Работа со строками		
5.3. Дополнительные операторы		
Раздел 6. Дополнительные свойства классов и методов	ПК-2	31, 32, У1, Н1, Н2
6.1. Права доступа к членам класса		
6.2. Перегрузка методов		
6.3. Использование ключевого слова static		
Раздел 7. Наследование	ПК-2	31, 32, У1, Н1, Н2
7.1. Базовые сведения о наследовании		
7.2. Доступ к членам класса при наследовании		
7.3. Использование конструкторов при наследовании		
7.4. Переопределение методов базового класса		
7.5. Ссылка на базовый класс и полиморфизм		
7.6. Абстрактный класс		
7.7. Использование ключевого слова final		
Раздел 8. Использование пакетов и интерфейсов	ПК-2	31, 32, 33, 34, У1, У2, Н1, Н2
8.1. Использование пакетов		
8.2. Права доступа при использовании пакетов		
8.3. Импорт пакетов		
8.4. Понятие интерфейса		
8.5. Создание интерфейса, использование ссылки на интерфейс		
8.6. Реализация методов интерфейса по умолчанию.		
8.7. Основные шаблоны проектирования, базирующиеся на интерфейсах		
Раздел 9. Обработка исключений	ПК-2	31, 32, 33, 34, У1, У2, Н1, Н2
9.1. Базовое определение исключения		
9.2. Синтаксис обработки исключения		
9.3. Стандартные исключения, иерархия исключений		
9.4. Обработка нескольких исключений		
Раздел 10. Использование потоков ввода/вывода		31, 32, 33, 34, У1, У2, Н1, Н2
10.1. Понятие потока в Java		
10.2. Базовые интерфейсы для работы с потоками		
10.3. Предопределенные классы для работы с потоками		
10.4. Автоматическое закрытие ресурсов		
Зачет	ПК-2	31, 32, 33, 34, У1, У2, Н1, Н2
Раздел 11. Параллельное программирование	ПК-2	34, У2, У4, Н1, Н2
11.1. Основные сведения о потоках управления		
11.2. Создание нескольких потоков		
11.3. Управление потоками		
11.4. Синхронизация потоков		
Раздел 12. Перечисления, автоупаковка, аннотации	ПК-2	31, 32, 33, 34, У1, У2, Н1, Н2
12.1. Перечисления		
12.2. Автоупаковка, автораспаковка, классы-обертки		
12.3. Введение в аннотации		
Раздел 13. Использование generic в Java	ПК-2	34, 35, 36, У2, У3, У4, Н1, Н2
13.1. Определение дженериков (generics)		
13.2. Использование wildcards		
13.3. Рассмотрение стандартных коллекций в Java, основанных на дженериках		
Раздел 14. Лямбда-выражения и ссылки на методы классов	ПК-2	34, 35, 36, У2, У3, У4, Н1, Н2
14.1. Понятие лямбда-выражения и функционального интерфейса		
14.2. Стандартные функциональные интерфейсы		
14.3. Особенности лямбда-выражений		

14.4. Использование лямбда-выражений при обработке потоков		
14.5. Ссылка на метод класса		
14.6. Ссылка на конструктор		
Раздел 15. Апплеты и события	ПК-2	34, 35, 36, 37, У2, У3, У4, Н1, Н2, Н3
15.1. Апплеты		
15.2. Стадии работы апплета		
15.3. Обработка событий		
Раздел 16. Введение в Swing	ПК-2	34, 35, 36, 37, У2, У3, У4, У5, Н1, Н2, Н3
16.1. Компоненты и контейнеры		
16.2. Layout менеджер		
16.3. Пример интерфейса на Swing		
16.4. Основные компоненты		
16.5. Использование лямбда-выражений для обработки событий		
Раздел 17. Введение в JavaFX	ПК-2	34, 35, 36, 37, У2, У3, У4, У5, Н1, Н2, Н3
17.1. Основные понятия JavaFX		
17.2. Структура приложения с JavaFX		
17.3. Использование основных компонент в JavaFX		
Расчетно-графическая работа	ПК-2	34, 35, 36, 37, У2, У3, У4, У5, Н1, Н2, Н3
Экзамен	ПК-2	31, 32, 33, 34, У1, У2, У3, У4, Н1, Н2, Н3

4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения

Содержание	Всего, час	Контактная работа, час				СРС, час	ИФР, час	К, час
		Л	л/р	п/р	КСР			
Раздел 1. Основы языка программирования Java.								
1.1. История развития языка Java.	8	2	4			2		
1.2. Основные особенности языка.								
1.3. Объектно-ориентированный подход в программировании на языке Java.								
1.4. Простейшая программа на Java.								
Раздел 2. Типы данных и операторы в Java								
2.1. Примитивные типы данных в Java	3	2				1		
2.2. Создание и инициализация переменных в Java								
2.3. Область действия переменных								
2.4. Операторы в Java	3	2				1		
Раздел 3. Операторы управления в Java								
3.1. Операторы ветвления	4	2				2		
3.2. Операторы цикла								
3.3. Операторы перехода								
Раздел 4. Введение в классы и методы								
4.1. Базовый синтаксис классов	3	2				1		
4.2. Переменные ссылки и создание объектов								
4.3. Создание методов в классе, параметры и возвращаемое значение	3	2				1		
4.4. Создание конструкторов								
4.5. Сборщик мусора и метод finalize								
Раздел 5. Дополнительные типы данных и операторы								
5.1. Создание массивов	8	2	4			2		

5.2. Работа со строками								
5.3. Дополнительные операторы								
Раздел 6. Дополнительные свойства классов и методов								
6.1. Права доступа к членам класса	4	2				2		
6.2. Перегрузка методов								
6.3. Использование ключевого слова static								
Раздел 7. Наследование								
7.1. Базовые сведения о наследовании	4	2				2		
7.2. Доступ к членам класса при наследовании								
7.3. Использование конструкторов при наследовании								
7.4. Переопределение методов	4	2				2		
7.5. Ссылка на базовый класс и полиморфизм								
7.6. Абстрактный класс								
7.7. Ключевое слово final								
Раздел 8. Использование пакетов и интерфейсов								
8.1. Использование пакетов	3	2				1		
8.2. Права доступа при использовании пакетов								
8.3. Импорт пакетов								
8.4. Понятие интерфейса	3	2				1		
8.5. Создание интерфейса, использование ссылки на интерфейс								
8.6. Реализация методов интерфейса по умолчанию.								
8.7. Основные шаблоны проектирования, базирующиеся на интерфейсах	9	4	4			1		
Раздел 9. Обработка исключений								
9.1. Базовое определения исключения	3	2				1		
9.2. Синтаксис обработки исключения								
9.3. Стандартные исключения, иерархия исключений								
9.4. Обработка нескольких исключений								
Раздел 10. Использование потоков ввода/вывода								
10.1. Понятие потока в Java	7	2	4			1		
10.2. Базовые интерфейсы для работы с потоками								
10.3. Предопределенные классы для работы с потоками								
10.4. Автоматическое закрытие ресурсов								
Зачет	3					3		
За семестр	72	32	16			24		
Раздел 11. Параллельное программирование								
11.1. Основные сведения о потоках управления	7	2	4			1		
11.2. Создание нескольких потоков								
11.3. Управление потоками								
11.4. Синхронизация потоков	8	2	2			4		
Раздел 12. Перечисления, автоупаковка, аннотации								
12.1. Перечисления	3	2				1		
12.2. Автоупаковка, автораспаковка,								

классы-обертки								
12.3. Введение в аннотации	3	2				1		
Раздел 13. Использование generic в Java								
13.1. Определение дженериков (generics)	5	2	2			1		
13.2. Использование wildcards								
13.3. Рассмотрение стандартных коллекций в Java, основанных на дженериках	19	2	4			3		
Раздел 14. Лямбда-выражения и ссылки на методы классов								
14.1. Понятие лямбда-выражения и функционального интерфейса	8	2	4			2		
14.2. Стандартные функциональные интерфейсы								
14.3. Особенности лямбда-выражений								
14.4. Использование лямбда-выражений при обработке потоков	3	2				1		
14.5. Ссылка на метод класса								
14.6. Ссылка на конструктор								
Раздел 15. Апплеты и события								
15.1. Апплеты	3	2				1		
15.2. Стадии работы апплета								
15.3. Обработка событий								
Раздел 16. Введение в Swing								
16.1. Компоненты и контейнеры	3	2				1		
16.2. Layout менеджер								
16.3. Пример интерфейса на Swing								
16.4. Основные компоненты	13	2	8			3		
16.5. Использование лямбда-выражений для обработки событий								
Раздел 17. Введение в JavaFX								
17.1. Основные понятия JavaFX	3	2				1		
17.2. Структура приложения с JavaFX								
17.3. Использование основных компонент в JavaFX								
Расчетно-графическая работа	4				2	2		
Экзамен	36							36
За семестр	108	24	24		2	22		36
Итого	180, 5 з.е.	56	40		2	46		36

5. Содержание разделов дисциплины

5.1. Лекции

Тема 1. Основы языка программирования Java

Лекция 1. Предмет дисциплины, ее объем, задачи и цели, содержание и связь с другими дисциплинами учебного плана. Характеристика учебной литературы. Происхождение языка Java, его основные отличия и предназначение. Преимущества и недостатки Java-программ. Понятие байт-кода. Пример простейшей программы.

Тема 2. Типы данных и операторы в Java.

Лекция 2. Примитивные типы данных в Java, их характеристики. Отличие примитивных типов данных от объектов. Создание и инициализация примитивных типов данных в Java. Литералы в Java. Область действия и время жизни локальных переменных.

Лекция 3. Операторы в Java. Арифметические операторы, логические операторы, сокращенные логические операторы, операторы присваивания. Преобразование типов данных при присваивании, расширяющее и сужающее приведение типов данных.

Тема 3. Операторы управления в Java

Лекция 4. Оператор ветвления if, полная и сокращенная версия оператора ветвления. Оператор switch, выражения допустимые в операторе switch. Операторы цикла – for, while и do. Операторы перехода break, continue и return. Использование операторов break и return с метками. Вложенные циклы.

Тема 4. Введение в классы и методы

Лекция 5. Базовое понятие о классах и объектах. Синтаксис описания нового класса. Понятие переменной-ссылки на объект. Создание нового объекта с помощью ключевого слова new. Основные сведения о создании и удалении объектов, понятие сборщик мусора. Работа с ссылками в Java.

Лекция 6. Задание методов в классе. Особенности передачи параметров в метод в языке программирования Java. Указание возвращаемого значения в методе. Создание конструкторов. конструкторы с параметром. Перегруженные методы. Метод finalize. Ключевое слово this.

Тема 5. Дополнительные типы данных и операторы

Лекция 7. Создание массивов в Java. Массив как класс, методы массива. Создание одномерных массивов. Создание многомерных массивов. Создание разреженных массивов. Оператор for-each для работы с массивами. Работа со строками в Java. Побитовые операторы и тернарный оператор.

Тема 6. Дополнительные свойства классов и методов.

Лекция 8. Указание прав доступа к полям и методам класса. Передача объектов в качестве параметров метода. Перегрузка методов и конструкторов. Вариант определения параметров по умолчанию. Использование ключевого слова static. Вложенные и внутренние классы.

Тема 7. Наследование.

Лекция 9. Базовое представление о наследовании. Синтаксис построение иерархии классов. Права доступа к полям и методам базового и дочернего классов при наследовании. Ключевое слово super. Порядок вызова конструкторов дочернего и базового классов.

Лекция 10. Переопределение методов базового класса. Понятие полиморфизма. Особенности присваивания значения ссылке на базовый класс. Понятие абстрактного метода и класса. Использование абстрактных классов. Использование ключевого слова final. Базовый класс Object как общий предок всех классов.

Тема 8. Использование пакетов и интерфейсов.

Лекция 11. Основное определение пакета, принципы именования пакетов, синтаксис задания пакетов и импорта пакетов в приложении. Место пакетов в механизме ограничения прав доступа. Использование права доступа protected. Использование пакетов в базовой библиотеке Java.

Лекция 12. Основное понятие интерфейса. Отличие интерфейса от класса. Синтаксис определения интерфейса и его реализации в классе. Использование ссылки на интерфейс. Переменные в интерфейсах. Реализация методов интерфейса по умолчанию. Использование static в интерфейсе. Место интерфейсов в языке Java.

Лекция 13. Основное понятие о шаблонах проектирования. Шаблон одиночка (Singleton). Шаблон наблюдатель (Listener). Шаблон декоратор.

Лекция 14. Прочие шаблоны проектирования – адаптер, итератор, состояние, основные сведения о шаблоне Model-View-Controller.

Тема 9. Обработка исключений .

Лекция 15. Основные понятие об исключениях. Иерархия исключений. Основы обработки исключений (блоки try-catch). Последствия необработанных исключений. Обработка нескольких исключений. Обработка исключения находящегося выше в иерархии исключений. Генерация исключения (ключевое слово throw). Использование блока finally. Указание генерируемых исключений в методе (throws). Стандартные исключения в Java.

Тема 10. Использование потоков ввода/вывода.

Лекция 16. Основные сведения и понятие о потоках ввода/вывода в Java, введение понятия stream. Потоки для работы с байтами и символами. Базовые классы для работы с потоками и их основные методы. Конкретные классы для работы с потоками на примере работы с файлами. Использование шаблона декоратор для эффективной работы с потоками. Автоматическое закрытие потоков (ресурсов). Использование синтаксиса try-with-resources. Генерация и обработка исключений при работе с потоками.

Тема 11. Параллельное программирование

Лекция 17. Основные сведения о многопоточном подходе к программированию, преимуществах и недостатках, сложностях возникающих в процессе написания программ подобного рода. Класс Thread и Runnable в Java для создания дополнительных потоков. Создание потока. Ожидание завершения потока. Приоритеты потоков.

Лекция 18. Проблемы синхронизации при работе многопоточного приложения. Использование ключевого слова synchronized. Вопросы производительности при использовании синхронизации. Использование методов notify, notifyAll и wait для синхронизации потоков. Приостановка, возобновление и полная остановка потоков управления. Использование главного потока.

Тема 12. Перечисления, автоупаковка, аннотации.

Лекция 19. Основные сведения о перечислениях. Перечисления как класс в Java. Методы values() и valueOf() в перечислениях. Конструкторы и ссылки на перечисления. Автоупаковка и автораспаковка для представления элементарных типов данных в качестве объектов. Классы обертки (wrappers). Автоупаковка и автораспаковка в выражениях.

Лекция 20. Введение в аннотации. Предназначение аннотаций. Основные аннотации.

Тема 13. Использование generic в Java.

Лекция 21. Определение дженериков в Java. Синтаксис определения дженериков, простейший пример создания дженериков с подстановочным типом данных. Важность дженериков в Java. Ограниченный тип данных в дженериках. Использование wildcard. Дженерик методы, конструкторы и интерфейсы. Diamond оператор.

Лекция 22. Рассмотрение стандартных коллекций в Java основанных на джереиках. Коллекции ArrayList, LinkedList, HashMap. Основные сведения о методах hashCode и equals. Использование интерфейса Comparator для сортировки коллекций.

Тема 14. Лямбда-выражения и ссылки на методы классов

Лекция 23. Введение в лямбда-выражения. Понятие функционального интерфейса и лямбда-выражения. Пример простейшего использования лямбда-выражения. Использование лямбда-выражения для сортировки коллекций, создания потоков. Стандартные функциональные интерфейсы. Особенности лямбда-выражений.

Лекция 24. Введения понятия потока (stream) как последовательности операция применяемых к набору входных данных. Использование лямбда-выражений при обработке потоков. Основные методы и функциональные интерфейсы для обработки потоков. Ссылки на методы классов. Ссылки на конструкторы.

Тема 15. Апплеты и события

Лекция 25. Базовые сведения об апплетах. Жизненный цикл апплетов. Инициализация и завершение апплетов. Перерисовка графического представления апплета. Класс Applet. Основные сведения о событиях. Источник события. Слушатель события. Основные классы для представления стандартных событий.

Тема 16. Введение в Swing

Лекция 26. Основные понятия, введенные в пакете Swing – компонент, контейнер, панель, layout менеджер. Верхний уровень иерархии контейнеров. Основные классы и методы для работы с контейнерами. Пример простейшей программы демонстрирующей графический пользовательский интерфейс, созданный с использованием пакета swing.

Лекция 27. Основные компоненты и принципы работы с ними. Использование анонимных классов или лямбда-выражений для обработки событий. Виды событий и их обработчики. Использование компонентов JButton, JTextField, JCheckBox и JList.

Тема 17. Введение в JavaFX.

Лекция 28. Основные понятия JavaFX. Структура приложения с JavaFX. Использование основных компонент в JavaFX

5.2. Лабораторные работы

Лабораторная работа 1. Консольное приложение на Java (задание в электронном виде на сервере кафедры).

Лабораторная работа 2. Работа с массивами в Java (задание в электронном виде на сервере кафедры).

Лабораторная работа 3. Работа с интерфейсами и использование шаблонов проектирования (задание в электронном виде на сервере кафедры).

Лабораторная работа 4. Использование потоков ввода/вывода в Java (задание в электронном виде на сервере кафедры).

Лабораторная работа 5. Создание многопоточного приложения в Java (задание в электронном виде на сервере кафедры).

Лабораторная работа 6. Использование стандартных коллекций в Java (задание в электронном виде на сервере кафедры).

Лабораторная работа 7. Использование лямбда-выражений в Java (задание в электронном виде на сервере кафедры).

Лабораторная работа 8. Создание пользовательского интерфейса с использованием пакета swing (задание в электронном виде на сервере кафедры).

№	Тема	Количество ауд. часов
1	Лабораторная работа 1. Консольное приложение на Java	
1.1	Ознакомление с инструментальными средствами компилирования и запуска Java программ с использованием Java Developer Kit (JDK 8).	4
1.2	Ознакомление со средой разработки IntelliJ Idea.	4
1.3	Написание и отладка простейшей консольной программы на Java с использованием интегрированной среды разработки IntelliJ Idea.	4
2	Лабораторная работа 2. Работа с массивами в Java	
2.1	Ознакомление с синтаксисом написания классов, указания ссылок, создания объектов.	4
2.2	Ознакомление с принципами работы с массивами, стандартными классами и методами для работы с массивами в Java.	4
2.3	Написание программы для демонстрации принципов работы с массивами и классами в соответствии с индивидуальным заданием.	4
3	Лабораторная работа 3. Работа с интерфейсами и использование шаблонов проектирования	
3.1	Ознакомление с синтаксисом создания интерфейсов.	2
3.2	Ознакомление с шаблоном проектирования в соответствии с индивидуальным заданием и реализации программы для демонстрации принципов работы выбранного шаблона.	2
4	Лабораторная работа 4. Использование потоков ввода/вывода в Java	
4.1	Ознакомление с основными классами для работы с потоками, их методами, преимуществами и недостатками использования тех или иных классов.	2

4.2	Разработка и реализация программы в соответствии с индивидуальным заданием с использованием классов для работы с потоками ввода/вывода. Определение времени выполнения программы для демонстрации преимущества использования буферизированных потоков ввода/вывода.	2
5	Лабораторная работа 5. Создание многопоточного приложения в Java	
	Ознакомление с синтаксисом создания и управления потоками в Java, их синхронизацией, а так же основными преимуществами и недостатками многопоточных приложений в сравнении с однопоточными. Реализация многопоточного приложения в соответствии с индивидуальным заданием.	3
	Лабораторная работа 6. Использование стандартных коллекций в Java.	
	Ознакомление с основными коллекциями в Java, а так же их общими методами и способами работы с ними.	3
	Разработка способа представления данных для реализации индивидуального задания, выбор наиболее подходящей коллекции для представления данных, а так же реализации алгоритма обработки данных с опорой на методы, предоставляемые стандартными коллекциями Java.	6
	Лабораторная работа 7. Использование лямбда-выражений в Java	
	Ознакомление с синтаксисом использования лямбда-выражений, стандартными функциональными интерфейсами, методами работы с потоками. Использование лямбда-выражений для реализации индивидуального задания в соответствии с методическими указаниями.	3
	Лабораторная работа 8. Создание пользовательского интерфейса с использованием пакета Swing.	
	Ознакомление с архитектурой компонентов и контейнеров пакета Swing. Ознакомление с основными свойствами и методами необходимых компонент.	3
	Разработка пользовательского интерфейса для отображение структуры данных в соответствии с индивидуальным заданием. Написание процедур обработки событий, генерируемых при взаимодействии с пользователем для модификации, отображения или удаления записей таблиц данных.	6
	Всего	56

6. Образовательные технологии

В соответствии со структурой образовательного процесса по дисциплине применяются следующие технологии:

- диагностики;
- целеполагания;
- управления процессом освоения учебной информации;
- применения знаний на практике, поиска новой учебной информации;
- организации совместной и самостоятельной деятельности обучающихся (учебно-познавательной, научно-исследовательской, частично-поисковой, репродуктивной, творческой и пр.);
- контроля качества и оценивания результатов образовательной деятельности (технология оценивания качества знаний, рейтинговая технология оценки знаний и др.)

В соответствии с требованиями ФГОС ВО для реализации компетентного подхода при обучении дисциплине предусмотрено широкое использование в учебном процессе активных и интерактивных методов проведения занятий:

При обучении дисциплине применяются следующие формы занятий:

- лекции, направленные на получение новых и углубление научно-теоретических знаний, в том числе вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция, лекции-дискуссии, лекции-беседы и др.;
- лабораторные занятия, проводимые под руководством преподавателя в учебной лаборатории с использованием компьютеров и учебного оборудования, направленные на закрепление и получение новых умений и навыков, применение знаний и умений, полученных на теоретических занятиях, при решении практических задач и др.

Все занятия обеспечены мультимедийными средствами (проекторы, экраны) для повышения качества восприятия изучаемого материала. В образовательном процессе широко используются информационно-коммуникационные технологии.

Самостоятельная работа студентов – это планируемая работа студентов, выполняемая по заданию при методическом руководстве преподавателя, но без его непосредственного участия. Формы самостоятельной работы студентов определяются содержанием учебной дисциплины, степенью подготовленности студентов. Они могут иметь учебный или учебно-исследовательский характер: подготовка к лабораторным работам, выполнение расчетно-графической работы и др.

Формами контроля самостоятельной работы выступают проверка письменных отчетов по результатам выполненных заданий и лабораторных работ, расчетно-графической работы. Результаты самостоятельной работы учитываются при оценке знаний - зачёте.

7. Формы аттестации и оценочные материалы

Текущий контроль предназначен для проверки хода и качества усвоения учебного материала, стимулирования учебной работы обучающихся и совершенствования методики проведения занятий. Он проводится в ходе всех видов занятий в форме, избранной преподавателем.

Промежуточная аттестация имеет целью определить степень достижения учебных целей по учебной дисциплине и проводится в форме зачета или допуска к экзамену. Принимается зачет (допуск) преподавателями, читающими лекции по данной учебной дисциплине в соответствии с перечнем основных вопросов, выносимых для контроля знаний обучающихся. К экзамену допускаются студенты, выполнившие и защитившие все лабораторные работы. Экзамен проводится по билетам, включающим в себя два теоретических вопроса.

7.1. Вопросы и задачи к зачету и к экзамену

1. Классы языка Java и их синтаксис. Члены класса.
2. Поля и методы класса. Главный метод приложения.
3. Конструкторы класса. Цепочки конструкторов. Блоки инициализации.
4. Статические поля и методы классов. Инициализация статических полей класса.
5. Пакеты классов и интерфейсов. Импорт классов и интерфейсов из других пакетов.
6. Разновидности видимости классов и интерфейсов. Соглашения об именовании пакетов, классов, директорий и файлов при программировании на Java.
7. Наследование полей и методов классов в языке Java. Перекрытие наследуемых методов.
8. Использование конструкторов наследуемых классов. Разновидности видимости членов классов.
9. Полиморфизм. Абстрактные классы. Конечные (final) классы.
10. Вложенность классов. Статические вложенные и внутренние классы. Доступ к статическим вложенным классам. Внутренние классы.
11. Внутренние анонимные классы.

12. Интерфейсы языка Java. Члены интерфейсов языка Java. Реализация интерфейсов классами.
13. Наследование интерфейсов. Назначение и использование интерфейсов при программировании на языке Java.
14. Основные шаблоны проектирования.
15. Типы-перечисления. Поля и методы типов-перечислений.
16. Аннотации. Объявление типа аннотации и его использование. Аннотации для компилятора.
17. Автоупаковка и автораспаковка. Классы обертки для элементарных типов данных.
18. Исключения (exceptions) в языке Java.
19. Порождение и перехват исключений. Список исключений в объявлении метода.
20. Источники исключений. Иерархия исключений.
21. Приоритеты при обработке исключений.
22. Стандартная библиотека коллекций языка Java. Интерфейсы, реализации и алгоритмы коллекций.
23. Структура библиотеки коллекций. Коллекции, множества и списки.
24. Карты (maps) в библиотеке коллекций. Использование различных реализаций карт.
25. Функции hashCode и equals.
26. Итераторы карт и коллекций.
27. Стандартные алгоритмы библиотеки для работы с коллекциями и массивами.
28. Использование дженерик классов и интерфейсов в языке Java.
29. Ограничение возможных типов данных в дженериках. Использование wildcard.
30. Потоки ввода/вывода языка Java.
31. Назначение и возможности классов OutputStream, FileOutputStream, BufferedOutputStream и DataOutputStream.
32. Назначение и возможности классов InputStream, FileInputStream, BufferedInputStream, DataInputStream.
33. Автоматическое закрытие ресурсов. Использование блока try-with-resources для автоматического закрытия ресурсов.
34. Лямбда выражения в языке Java8. Функциональные интерфейсы. Синтаксис Lambda-выражений.
35. Замена анонимных классов лямбда выражениями.
36. Примеры функциональных интерфейсов из пакета java.util.function.
37. Потоки в языке Java8. Определение потоков. Обработка потока в конвейере.
38. Методы для порождения потоков. Методы преобразования потоков в конвейере.
39. Финальные операции над потоками.
40. Многопоточное программирование в Java. Использование классов или интерфейсов для создания потока и управление им.
41. Синхронизация доступа к объекту из разных потоков.
42. Иерархия классов пакета Swing для представления пользовательского интерфейса.
43. Компоненты и контейнеры Swing.
44. Layout-менеджер для размещения компонентов в контейнерах. Основные типы менеджера размещения компонентов.

7.2. Выполнение и примерные задания расчетно-графической работы

Расчетно-графическая работа выполняется в процессе изучения дисциплины. Общее руководство и контроль за ходом выполнения расчетно-графической работы осуществляет преподаватель соответствующей дисциплины. Расчетно-графическая работа выполняется в соответствии с методическими указаниями доступными в электронной форме на сервере кафедры.

Основными функциями руководителя расчетно-графической работы являются:

- определение и формулирование задания расчетно-графической работы;

- консультирование по вопросам содержания и последовательности выполнения расчетно-графической работы;
- оказание помощи студенту в подборе необходимой литературы;
- контроль хода выполнения расчетно-графической работы.

Примерные задания для выполнения расчетно-графической работы:

1. Реализация модели БД автомастерской

Реализовать структуру данных для представления нескольких таблиц для моделирования базы данных автомастерской. База данных должна состоять как минимум из трех таблиц: информация о мастере, информация о машине и таблица, реализующая связь между мастером, занимающимся ремонтом машины, и самой машиной. Каждая таблица должна иметь как минимум три поля. Например, машина может содержать гос.номер, цвет и год выпуска.

Необходимо выбрать контейнеры для хранения таблиц, реализовать методы для их графического ввода и вывода, используя библиотеку swing. При вводе данных необходимо контролировать их корректность.

2. Реализация модели БД продуктового магазина

Реализовать структуру данных для представления нескольких таблиц для моделирования базы данных продуктового магазина. База данных должна состоять как минимум из трех таблиц: информация о товаре, информация о продавцах и таблица, реализующая связь между проданным товаром и продавцом, осуществившим продажу. Каждая таблица должна иметь как минимум три поля. Например, товар может содержать артикул, цену и остаток на складе.

Необходимо выбрать контейнеры для хранения таблиц, реализовать методы для их графического ввода и вывода, используя библиотеку swing. При вводе данных необходимо контролировать их корректность.

3. Реализация модели БД ГИБДД

Реализовать структуру данных для представления нескольких таблиц для моделирования базы данных ГИБДД. База данных должна состоять как минимум из трех таблиц: информация о машине, информация о инспекторе и таблица, реализующая связь между машиной нарушавшей ПДД и инспектором, выписавшем протокол. Каждая таблица должна иметь как минимум три поля. Например, машина может содержать гос.номер, цвет, год выпуска и ФИО владельца.

Необходимо выбрать контейнеры для хранения таблиц, реализовать методы для их графического ввода и вывода, используя библиотеку swing. При вводе данных необходимо контролировать их корректность.

8. Учебно-методическое и информационное обеспечение дисциплины

Электронный каталог и электронные информационные ресурсы, предоставляемые научной библиотекой ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://library.chuvsu.ru/>

8.1. Рекомендуемая основная литература

№ п/п	Наименование
1.	Картузов А. В. Программирование на языке Java: учебное пособие / Картузов А. В., Калмыков Б. М., [отв. ред. Б. М. Калмыков] ; Чуваш. гос. ун-т им. И. Н. Ульянова - Чебоксары: Изд-во Чуваш. ун-та, 2009. - 326с.: ил. - ISBN 978-5-7677-1270-0.
2.	Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М. : Издательство Юрайт, 2017. — 206 с. — (Серия : Университеты России). — ISBN 978-5-534-00849-4. — Режим доступа: www.biblio-online.ru/book/BDEEFB2D-532D-4306-829E-5869F6BDA5F9 .

8.2. Рекомендуемая дополнительная литература

№ п/п	Наименование
-------	--------------

3.	Монахов В.В. Язык программирования Java и среда NetBeans [Электронный ресурс] / В.В. Монахов. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 450 с. — Режим доступа: http://www.iprbookshop.ru/73739.html
4.	Мухаметзянов Р.Р. Основы программирования на Java [Электронный ресурс] : учебное пособие / Р.Р. Мухаметзянов. — Электрон. текстовые данные. — Набережные Челны: Набережночелнинский государственный педагогический университет, 2017. — 114 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/66812.html
5.	Николаев Е.И. Объектно-ориентированное программирование. Часть I [Электронный ресурс] : лабораторный практикум / Е.И. Николаев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 183 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/62966.html
6.	Николаев Е.И. Объектно-ориентированное программирование [Электронный ресурс] : учебное пособие / Е.И. Николаев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 225 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/62967.html

8.3. Программное обеспечение, профессиональные базы данных, информационно-справочные системы.

Программное обеспечение, профессиональные базы данных, информационно-справочные системы, предоставляемые управлением информатизации ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://ui.chuvsu.ru/>

№ п/п	Наименование	Условия доступа/скачивания
	Комплект разработчика Java Development Kit	http://www.oracle.com/technetwork/java/javase/downloads/index.html дата обращения 27.08.2017
	Интегрированная среда разработки IntelliJ IDEA	https://www.jetbrains.com/idea/download/#section=windows дата обращения 27.08.2017

8.3.1. Базы данных, информационно-справочные системы

№ п/п	Наименование программного обеспечения	Условия доступа/скачивания
1.	Гарант	из внутренней сети университета (договор)*
2.	Консультант +	

8.4. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы

№ п/п	Наименование интернет ресурса	Режим доступа
1.	Официальная документация Java	https://docs.oracle.com/javase/8/docs/api/overview-summary.html дата обращения 27.08.2017

9. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных занятий по дисциплине оснащены автоматизированным рабочим местом (АРМ) преподавателя, обеспечивающим тематические иллюстрации и демонстрации, соответствующие программе дисциплины в составе:

- ПЭВМ с доступом в интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением;
- настенный экран.

Учебные аудитории для лабораторных и самостоятельных занятий по дисциплине оснащены АРМ преподавателя и пользовательскими АРМ по числу обучающихся, объединенных локальной сетью («компьютерный» класс), с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова».

10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

11. Методические рекомендации по освоению дисциплины

В ходе лекционных занятий студенту рекомендуется вести конспектирование учебного материала. Следует обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. При составлении конспекта желательно оставлять в рабочих конспектах поля, на которых в дальнейшем можно делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. В ходе лекционных занятий рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к лабораторным работам рекомендуется изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях. Основой для выполнения лабораторной работы являются разработанные кафедрой методические указания. Рекомендуется дорабатывать свой конспект лекций, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной рабочей программой дисциплины. В процессе подготовки студент может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при написании расчетно-графической работы.

Форма организации студентов на лабораторных работах: фронтально-индивидуальная. Все студенты выполняют одновременно одну и ту же работу по индивидуальному заданию в соответствии с порядковым номером студента в списке группы.

В результате выполнения лабораторной работы запланирована подготовка письменного отчета в соответствии с требованиями методических указаний. Качество выполнения лабораторных работ является важной составляющей оценки текущей успеваемости обучающегося.

Изменения и (или) дополнения от 01.09.2018 г (протокол №1 МК факультета ИВТ) к рабочей программе дисциплины (программе практики) «Кросс-платформенные средства разработки программного обеспечения» (направление подготовки 09.03.01 «Информатика и вычислительная техника», профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»):

к перечню учебной литературы и ресурсов сети «Интернет»

№ п/п	Рекомендуемая основная литература
1	Картузов А. В. Программирование на языке Java: учебное пособие / Картузов А. В., Калмыков Б. М., [отв. ред. Б. М. Калмыков] ; Чуваш. гос. ун-т им. И. Н. Ульянова - Чебоксары: Изд-во Чуваш. ун-та, 2009. - 326с.: ил.. - ISBN 978-5-7677-1270-0.
2	Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М. : Издательство Юрайт, 2017. — 206 с. — (Серия : Университеты России). — ISBN 978-5-534-00849-4. — Режим доступа: www.biblio-online.ru/book/BDEEFB2D-532D-4306-829E-5869F6BDA5F9 .
Рекомендуемая дополнительная литература	
1	Монахов В.В. Язык программирования Java и среда NetBeans [Электронный ресурс] / В.В. Монахов. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 450 с. — Режим доступа: http://www.iprbookshop.ru/73739.html
2	Мухаметзянов Р.Р. Основы программирования на Java [Электронный ресурс] : учебное пособие / Р.Р. Мухаметзянов. — Электрон. текстовые данные. — Набережные Челны: Набережночелнинский государственный педагогический университет, 2017. — 114 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/66812.html
3	Николаев Е.И. Объектно-ориентированное программирование. Часть 1 [Электронный ресурс] : лабораторный практикум / Е.И. Николаев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2015. — 183 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/62966.html

к перечню информационных технологий, включая перечень программного обеспечения и информационных справочных систем

№ п/п	Наименование Рекомендуемого ПО	Условия доступа/скачивания
1.	Microsoft Visual Studio	свободное лицензионное соглашение: https://visualstudio.microsoft.com/ru/free-developer-offers/
2.	Microsoft Windows	из внутренней сети университета (договор)
3.	Microsoft Office	
4.	Комплект разработчика Java Development Kit	свободное лицензионное соглашение: http://www.oracle.com/technetwork/java/javase/downloads/index.html
5.	Интегрированная среда разработки IntelliJ IDEA	свободное лицензионное соглашение: https://www.jetbrains.com/idea/download/#section=windows

Декан факультета

 А.В. Щипцова