

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета информационных
технологий и компьютерной безопасности

П.Ю. Гусев/
31.08.2021 г.

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Сетевое программирование»

**Направление подготовки (специальность) 09.03.02 Информационные
системы и технологии**

**Профиль (специализация) Системы автоматизации проектирования и
разработки информационных систем**

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения Очная

Год начала подготовки 2019 г.

Автор(ы) программы _____

подпись

Е.Н. Королев

**Заведующий кафедрой Системы
автоматизированного проектирования
и информационные системы** _____

Я.Е. Львович

Руководитель ОПОП _____

подпись

О.Г. Яскевич

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

изучение основных принципов и технологий сетевого программирования. Изучение принципов и особенностей построения серверных и клиентских приложений, двух и трехзвенных архитектур построения сетевых приложений процессами. Кроме того, задачей курса является изучение идеологии и архитектуры современных распределенных систем, а также получения навыков написания клиент-серверных приложений с различной архитектурой и с использованием различных современных протоколов.

Изучение дисциплины должно способствовать формированию у студентов основ научного мышления, в том числе: пониманию принципов построения распределенных информационных систем; умению оценивать эффективность применения различных информационных технологий для сетевого программирования.

1.2. Задачи освоения дисциплины

- ознакомление студентов с разными технологиями и архитектурами построения сетевых приложений;
- изучение пакета java.net и технологии Сервлетов и JSP;
- изучение принципов сетевого программирования с использованием протоколов TCP и UDP;
- приобретение навыков работы в современных средах разработки сетевых приложений;
- приобретение навыков разработки сетевых приложений, как клиентской, так и серверной части;

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Сетевое программирование» относится к дисциплинам вариативной части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Сетевое программирование» направлен на формирование следующих компетенций:

ПК-1 - Способен выполнять синтез требований к программному продукту и декомпозицию программного средства на компоненты

ПК-5 - Способен определять внешние и внутренние интерфейсы каждого из компонент информационной системы и осуществлять их разработку

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	знать синтаксис и основные возможности языка java и основных пакетов для разработки сетевых приложений

	<p>уметь выполнять работы по созданию (модификации) и сопровождению сетевых приложений, выполнять синтез требований к программному продукту и декомпозицию программного средства на компоненты</p> <p>владеть инструментальными средствами, методами и навыками разработки сетевого программного обеспечения для разработки компонент программного обеспечения с использованием возможностей пакета java.net, технологий Servlet и JSP</p>
ПК-5	<p>знать возможности пакета java.net для разработки сетевых приложений; технологии сетевого доступа к базам данных при проектировании и разработке автоматизированных систем среднего и большого уровня сложности</p>
	<p>уметь разрабатывать компоненты информационной системы, используя технологии сетевого программирования и определять внешние и внутренние интерфейсы каждого из компонент</p>
	<p>владеть инструментальными средствами и технологиями для разработки компонент информационных систем среднего и большого уровня сложности</p>

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Сетевое программирование» составляет 5 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		6
Аудиторные занятия (всего)	90	90
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР), в том числе в форме практической подготовки	54	54
Самостоятельная работа	54	54
Курсовой проект	+	+
Часы на контроль	36	36
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость:		
академические часы	180	180
зач.ед.	5	5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Основы языка программирования Java	Особенности языка Java. Принципы ООП. Реализация принципов. Основы языка java. Основные конструкции, выражения, циклы, операции. Понятия класса, объекта, поле, метод, области видимости. Простейшая программа. Типы данных, ключевые слова, классы. Работа с файлами.	6	10	8	24
2	Многопоточные сетевые приложения на Java	Реализация многопоточности. Создание и запуск потока. Синхронизация потоков. Апплет с многопоточностью. Интерфейсы. Реализация интерфейсов.	6	10	8	24
3	Сетевое программирование. Пакет java.net	Работа с сетью. Протоколы. Работа с сетью. Классы InetAddress, URL. Работа с сетью. Классы Socket, ServerSocket. Классы DatagramSocket, DatagramPacket.	6	6	8	24
		практическая подготовка обучающихся		4		
4	Работа с удаленными базами данных	Библиотека JDBC, подключение к базе данных. Объекты Connection, Statement, PreparedStatement. Библиотека JDBC, обработка SQL запросов. Объект ResultSet, ResultSetMetaData.	6	4	10	24
		практическая подготовка обучающихся		4		
5	Технология написания серверных компонент Servlet	Основы сервлетов, преимущества, жизненный цикл. Простой сервлет, формы, обработка данных формы. Объекты HttpServletRequest, HttpServletResponse. Понятие сессии и куки. Аутентификация, обработка запросов	6	4	10	24
		практическая подготовка обучающихся		4		

6	Технология написания серверных компонент JSP	Основы технологии JSP. Выражения, директивы. Использование JSP с применением JavaBeans.	6	4	10	24
		практическая подготовка обучающихся		4		
Итого			36	54	54	144

Практическая подготовка при освоении дисциплины (модуля) проводится путем непосредственного выполнения обучающимися отдельных элементов работ, связанных с будущей профессиональной деятельностью, способствующих формированию, закреплению и развитию практических навыков и компетенций по профилю соответствующей образовательной программы на практических занятиях и (или) лабораторных работах.

№ п/п	Перечень выполняемых обучающимися отдельных элементов работ, связанных с будущей профессиональной деятельностью	Формируемые профессиональные компетенции
1	Разработка сетевых приложений с помощью Socket и DatagramSocket	ПК-5
2	Построение приложений с использованием Swing и JDBC	ПК-1
3	Разработка серверных компонент с помощью технологии Servlet	ПК-1
4	Разработка серверных компонент с помощью технологии JSP и JavaBeans	ПК-1

5.2 Перечень лабораторных работ

1. Лабораторная работа №1. Объектно-ориентированное программирование на языке Java.
2. Лабораторная работа №2. Написание многопоточного приложения на языке Java.
3. Лабораторная работа №3. Разработка сетевых приложений с помощью Socket и DatagramSocket
4. Лабораторная работа №4. Построение приложений с использованием Swing и JDBC
5. Лабораторная работа №5. Разработка серверных компонент с помощью технологии Servlet
6. Лабораторная работа №6. Разработка серверных компонент с помощью технологии JSP и JavaBeans

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 6 семестре для очной формы обучения.

Примерная тематика курсового проекта: «Разработка многопоточного сетевого приложения на языке Java с использованием пакета java.net, сервлетов и JSP»

Задачи, решаемые при выполнении курсового проекта:

- Изучение пакета java.net, сервлетов и JSP.
- Проектирование сетевого приложения.
- Разработка компонент сетевого многопоточного приложения

Курсовой проект включают в себя графическую часть и расчетно-пояснительную записку.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-1	знать синтаксис и основные возможности языка java и основных пакетов для разработки сетевых приложений	Знание синтаксиса и основных конструкций языка. Знание пакета java.net	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь выполнять работы по созданию (модификации) и сопровождению сетевых приложений, выполнять синтез требований к программному продукту и декомпозицию программного средства на компоненты	Умение создавать проекты, реализующие сетевые приложения, автоматизирующие задачи организационного управления и бизнес-процессы	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть инструментальными средствами, методами и навыками разработки сетевого программного обеспечения для разработки компонент программного обеспечения с использованием	Владение инструментальными средствами для создания разработки сетевого программного обеспечения с использованием возможностей пакета java.net, технологий Servlet и JSP	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	возможностей пакета java.net, технологий Servlet и JSP			
ПК-5	знать возможности пакета java.net для разработки сетевых приложений; технологии сетевого доступа к базам данных при проектировании и разработке автоматизированных систем среднего и большого уровня сложности	Знание пакетов java.net, javax.servlet, javax.servlet.http, java.sql	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разрабатывать компоненты информационной системы, используя технологии сетевого программирования и определять внешние и внутренние интерфейсы каждого из компонент	умение разрабатывать компоненты программных комплексов с использованием пакетов java.net, javax.servlet, javax.servlet.http, java.sql	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть инструментальными средствами и технологиями для разработки компонент информационных систем среднего и большого уровня сложности	владение инструментальными средствами и технологиями для проектирования автоматизированных систем среднего и большого уровня сложности с использованием пакетов java.net, javax.servlet, javax.servlet.http, java.sql	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 6 семестре для очной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ПК-1	знать синтаксис и основные возможности языка java и основных пакетов для	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов

	разработки сетевых приложений					
	уметь выполнять работы по созданию (модификации) и сопровождению сетевых приложений, выполнять синтез требований к программному продукту и декомпозицию программного средства на компоненты	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть инструментальными и средствами, методами и навыками разработки сетевого программного обеспечения для разработки компонент программного обеспечения с использованием возможностей пакета <code>java.net</code> , технологий <code>Servlet</code> и <code>JSP</code>	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-5	знать возможности пакета <code>java.net</code> для разработки сетевых приложений; технологии сетевого доступа к базам данных при проектировании и разработке автоматизированных систем среднего и большого уровня сложности	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь разрабатывать компоненты информационной системы, используя технологии сетевого программирования и определять внешние и	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	внутренние интерфейсы каждого компонента из					
	владеть инструментальными и средствами и технологиями для разработки компонента информационных систем среднего и большого уровня сложности	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типичные контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Наследует ли класс конструкторы своего суперкласса?

- Нет
- Да

2. У каких операторов всегда вычисляются все операнды (выберите все подходящие варианты)?

- %
- ||
- && (2 амперсанда)
- ?: (тернарный оператор)

3. Какие модификаторы позволяют обращаться к полю/методу публичного класса верхнего уровня из других классов верхнего уровня, находящихся в том же пакете?

- по умолчанию (**package-private**)
- **public**
- **protected**
- private

4. Какой результат выполнения программы:

```
class Main {
public static void main(String[] args) {
    Integer i = new Integer("10");
    if (i.toString() == i.toString()) {
        System.out.println("Равный");
    } else {
        System.out.println("Неравный");
    }
}
```

```
}  
}
```

- **Выведется "Неравный"**
- Ошибка компилятора
- Выведется "Равный"
- Ни один из вариантов

5. Какой результат компиляции и запуска следующего кода с использованием параметров $a = 0$, $b = 3$?

```
public void divide(int a, int b) {  
    try {  
        int c = a / b;  
    } catch (Exception e) {  
        System.out.print("Exception ");  
    } finally {  
        System.out.println("Finally");  
    }  
}
```

- **Будет выведено Finally**
- Будет выведено Exception Finally
- Будет выведено Exception
- Ошибка компиляции

6. Можно ли при объявлении класса использовать модификаторы `abstract` и `final` одновременно?

- **Нет**
- Да

7. Может ли класс быть объявлен с модификатором `transient`?

- **Нет**
- Да

8. Имеем:

```
interface Animal {  
    void eat();  
}
```

// Сюда вставить код

```
public class HouseCat implements Feline {  
    public void eat() { }  
}
```

Какие из следующих строк можно вставить вместо комментария для успешной компиляции?

- **interface Feline extends Animal { }**
- **interface Feline extends Animal { void eat(); }**
- **interface Feline extends Animal { void eat() { } }**

9. Что будет напечатано после попытки компиляции и выполнения кода:

```
int[] mass = {1, 2};
List<String> list = new ArrayList(10);
list.add("03");
list.add("04");
System.out.println(mass.length + list.size() + ".");
```

- 4
- 2
- 0
- 6
- 12

10. Будет ли разница при исполнении этих строк кода?

```
1: int v1=1; long v2=2; v1=v1+v2;
2: int v1=1; long v2=2; v1+=v2;
```

- Да
- Нет

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Что будет выведено на экран?

```
class ClassA {
    int x = 1;

    public void printX() {
        System.out.println(getX());
    }

    public int getX() {
        return x;
    }
}
```

```
class ClassB extends ClassA {
    int x = 2;

    public int getX() {
        return x + 1;
    }
}
```

```

}

public class Test {
    public static void main(String[] args) {
        ClassA a = new ClassB();
        System.out.println(a.x);
    }
}

```

- 1
- 2
- 3
- Ошибка компиляции

2. Каким будет результат выполнения следующей программы?

```

import java.util.*;
public class TestFormatter {
    public static void main(String... args){
        Integer I1 = 0;
        Integer I2 = -1;
        Integer I3 = 1;
        Formatter f = new Formatter();
        f.format("%1$b ", I1.toString())
        .format("%1$b ", I2.toString())
        .format("%1$b ", I3.toString());
        System.out.println(f.toString());
    }
}

```

- **true true true**
- true false true
- false false true
- false true true
- 0 -1 1

3. Что выведется в консоль в результате компиляции и выполнения следующего кода:

```

public class A {
    {
        System.out.println("one");
    }

    public static void main(String[] args) {
        System.out.println("two");
    }
}

```

```

}

static {
    System.out.println("three");
}
}

```

- **three**
- **two**
- three
 - one
 - two
- one
 - two
 - three
- Ошибка компиляции

4. Каким будет результат компиляции и запуска следующей программы?

```

public class Test {
    public static void main(String[] args) {
        int i = 1;
        long j = 1;
        if (i == 1) {
            int k = 1;
            if (i == k) {
                System.out.println("i equals k");
            }
        } else if (j == k) {
            System.out.println("j equals k");
        }
    }
}

```

- **Ошибка компиляции**
- i equals k
- j equals k

5. Что произойдет при попытке откомпилировать/запустить следующий код:

```

abstract class Animal {
    static {
        System.out.println("Inside Animal");
    }
}

```

```
class Cat extends Animal {
    static {
        System.out.println("Inside Cat");
    }
}
```

```
class Dog extends Animal {
    static {
        System.out.println("Inside Dog");
    }
}
```

```
public class Main {
    public static void main(String[] args){
        Animal cat = new Cat();
        Animal dog = new Dog();
    }
}
```

- **В консоли будет выведено: Inside Animal Inside Cat Inside Dog**
- Произойдет ошибка компиляции
- В консоли будет выведено: Inside Animal Inside Cat Inside Animal Inside Dog
- В консоли будет выведено: Inside Cat Inside Dog

6. Выберите единственный верный из приведенных результатов компиляции и выполнения кода.

```
public class IntegerTest{
    public static void main(String[] argv){
        Integer i1 = new Integer("1");
        Integer i2 = new Integer("2");
        Integer i3 = Integer.valueOf("3");
        int i4 = i1 + i2 + i3;
        System.out.println(i4);
    }
}
```

- **Программа выведет "6" без кавычек**
- Ошибка компиляции: Integer class has no constructors which take a String argument
- Ошибка компиляции: Integer objects cannot work with + operator
- Программа выведет "123" без кавычек

7. Какой метод интерфейса `HttpServletRequest` вернет путь текущего запроса к веб-серверу:
- **`getServletPath`**
 - `getPath`
 - `getRoute`
 - `getServletRoute`
8. Какие из следующих методов есть у интерфейса `HttpServletResponse`:
- **`sendError`**
 - **`sendRedirect`**
 - `setStatus`
 - `sendHeader`
9. Каким образом в методе `init()` сервлета получить параметр инициализации сервлета с именем "PARAM"? (выберите два)
- **`getServletConfig().getInitParameter("PARAM");`**
 - **`this.getInitParameter("PARAM");`**
 - `HttpServletRequest.getInitParameter("PARAM");`
 - `ServletContext.getInitParameter("PARAM");`
 - `ServletConfig.getInitParameter("PARAM");`
10. Как нужно модифицировать сервлет, чтобы он поддерживал SSL шифрование данных?
- **Никаких изменений не требуется, это не задача сервлета**
 - Унаследовать `HttpSecureServlet`
 - Имплементировать интерфейс `HttpSSL`
 - Кодировать `HttpServletResponse` с помощью `encodeURL()` и декодировать `HttpServletRequest` с помощью `decodeURL()`

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Что случится после вызова метода `getParameter("param1")` у `HttpServletRequest` при сабмите следующей формы на сервер:
- ```
<form action="http://localhost:8080/mycontext/myservlet?param1=value1" method="POST">
 <input type="text" name="param1" value="value2" />
 <input type="submit" value="value3" />
</form>
```
- **Ничего из вышеперечисленного**
  - Произойдет ошибка времени выполнения
  - Метод будет возвращать одно из значений случайным образом
  - Метод вернет массив строк

2. С помощью какого класса/интерфейса можно получить cookie?

- **HttpServletRequest**
- HttpServletResponse
- SessionContext
- SessionConfig

3. Выберите варианты, которые, вставленные независимо в строке 5, приведут к созданию записи в лог-файле сервлета.

```
1. public class ServletTest extends HttpServlet {
2. public void doGet(HttpServletRequest request,
3. HttpServletResponse response) throws ServletException, IOException {
4. String message = "In doGet";
5. // код
6. }
7. }
```

- **log(message);**
- **getServletContext().log(message);**
- request.getSession().log(message);
- getServletConfig().log(message);
- request.log(message);
- getServletInfo().log(message);

3. Какие интерфейсы находятся в пакете javax.servlet?

- **Servlet;**
- **ServletRequest;**
- ServletOutputStream;
- PageContext;
- ServletContextEvent;

4. Какой метод сервлета FirstServlet будет вызван при активизации ссылки следующего HTML-документа?

```
<html>
<body>
OK!
</body>
```

<html> Соответствующий сервлету тег <url-pattern> в файле web.xml имеет вид: <url-pattern>/FirstServlettest</url-pattern>

- **doGet();**
- performTask();
- doPOST().

5. При выполнении следующих двух строк кода результат будет одинаковым:

- **Да**
- Нет



6. Какой из методов `HttpServletResponse` используется для перенаправления запроса на другой URL.
- **`sendRedirect()`**
  - `redirectHttp()`
  - `redirectURL()`
  - `sendURL()`
7. Выберите корректный способ указания вывода контента как приложенного файла
- **`response.setHeader("Content-Disposition", "attachment; filename=\"1.xlsx\")`**;
  - `response.header("Content-Disposition", "inline; filename=\"1.xlsx\")`;
  - `response.header("Content-Disposition: attachment; filename=\"1.xlsx\")`;
  - `request.setHeader("Content-Disposition", "attachment; filename=\"1.xlsx\")`;
8. Жизненный цикл сервлета управляется:
- **контейнером сервлетов (Servlet container)**
  - контекстом приложения (Application context)
  - дескриптором развертывания (Deployment descriptor)
  - контекстом сервлетов (Servlet context)
9. Какой из следующих интерфейсов предоставляет метод `getWriter` для получения экземпляра класса `PrintWriter`?
- **`ServletResponse`**
  - `HttpServletResponse`
  - `ServletContext`
  - `HttpServletRequest`

10. Как можно получить всю информацию из запроса, посланного следующей формой? (выберите два варианта ответа)

```
<HTML><BODY>
<FORM action="/com/MyServlet">
<INPUT type="file" name="filename">
<INPUT type="submit" value="Submit">
</FORM></BODY></HTML>
```

- **`request.getReader();`**
- **`request.getInputStream();`**
- `request.getAttribute("filename");`
- `request.getParameterValues("filename");`
- `request.getFileInputStream();`

#### 7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

### **7.2.5 Примерный перечень заданий для решения прикладных задач**

1. Особенности языка Java. Принципы ООП. Реализация принципов.
2. Понятия класса, объекта, поле, метод, области видимости. Простейшая программа.
3. Типы данных, простые, классы, преобразование типов.
4. Комментарии и встроенная документация.
5. Контроль над выполнением, конструкции языка if-else, return, while, do-while, for, break, continue, switch. Примеры.
6. Ключевые слова static, this, super, final. Примеры использования.
7. Спецификаторы доступа к классам, полям и методам класса в Java. Примеры.
8. Система ввода-вывода в Java. Работа с файлами. Примеры.
9. Апплеты, Структура и методы жизненного цикла апплета. Вызов апплета.
10. Программирование отношений типа «Является», «Имеет», «Использует», «Создает».
11. Интерфейсы. Реализация интерфейсов.
12. Исключительные ситуации. Обработка исключений.
13. Реализация многопоточности. Создание и запуск потока. Синхронизация потоков.
14. Апплет с многопоточностью.
15. Построение пользовательского интерфейса. Библиотека Swing. Визуальные компоненты.
16. Работа с сетью. Классы InetAddress, Socket, ServerSocket.
17. Работа с сетью. Классы InetAddress, DatagramSocket, DatagramPacket.
18. Библиотека JDBC, подключение к базе данных. Объекты Connection, Statement, PreparedStatement.
19. Реализация запроса SQL и обработка результатов. Объекты Statement, PreparedStatement, ResultSet.
20. Сервлеты, структура и организация, методы жизненного цикла.
21. JSP, структура и организация, работа с бинами. Класс Session.

### **7.2.6. Методика выставления оценки при проведении промежуточной аттестации**

*Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.*

*1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.*

*2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов*

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.

### 7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Основы языка программирования Java	ПК-1, ПК-5	Тест, защита лабораторных работ, требования к курсовому проекту.
2	Многопоточные сетевые приложения на Java	ПК-1, ПК-5	Тест, защита лабораторных работ, требования к курсовому проекту.
3	Сетевое программирование. Пакет java.net	ПК-1, ПК-5	Тест, защита лабораторных работ, требования к курсовому проекту.
4	Работа с удаленными базами данных	ПК-1, ПК-5	Тест, защита лабораторных работ, требования к курсовому проекту.
5	Технология написания серверных компонент Servlet	ПК-1, ПК-5	Тест, защита лабораторных работ, требования к курсовому проекту.
6	Технология написания серверных компонент JSP	ПК-1, ПК-5	Тест, защита лабораторных работ, требования к курсовому проекту.

### 7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем

осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

## **8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)**

### **8.1 Перечень учебной литературы, необходимой для освоения дисциплины**

1. Мухамедзянов Р.Р. JAVA. Серверные приложения [Электронный ресурс]/ Мухамедзянов Р.Р.— Электрон. текстовые данные.— М.: СОЛОН-ПРЕСС, 2010.— 336 с.— Режим доступа: <http://www.iprbookshop.ru/65089.html>.— ЭБС «IPRbooks»
2. Королев Е.Н. Методические указания к лабораторным работам по теме «Программирование на языке JAVA» для студентов очной формы обучения. ВГТУ, 2003.
3. Королев Е.Н. Методические указания к лабораторным работам по теме: «Программирование серверных компонент на языке JAVA» для студентов очной формы обучения № 33-2004. ВГТУ, 2004.
4. Королев Е.Н. Программирование под Internet на языке Java: учебное пособие. ВГТУ, 2003.
5. Королев Е.Н. Технологии сетевого программирования: учебное пособие. ВГТУ, 2015.
6. Дубаков А.А. Сетевое программирование [Электронный ресурс]: учебное пособие/ Дубаков А.А.— Электрон. текстовые данные.— СПб.: Университет ИТМО, 2013.— 249 с.— Режим доступа: <http://www.iprbookshop.ru/68118.html>.— ЭБС «IPRbooks»

**8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:**

Программное обеспечение  
MySQL  
NetBeans 8.1  
jdk 1.8

Ресурс информационно-телекоммуникационной сети «Интернет»

<http://www.edu.ru/>  
Образовательный портал ВГТУ

Информационная справочная система

<http://window.edu.ru>

<https://wiki.cchgeu.ru/>

Сайт, посвященный SQL, программированию, базам данных, разработке информационных систем <https://www.sql.ru/>

<https://habr.com/ru/>

## **9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

Специализированная лекционная аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой
--------------------------------------------------------------------------------------------------------------------------

Дисплейный класс, оснащенный компьютерными программами для проведения лабораторного практикума
------------------------------------------------------------------------------------------------

## **10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

По дисциплине «Сетевое программирование» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать

	вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

### 11 ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
1	Актуализирован раздел 8.1 Перечень учебной литературы, необходимой для освоения дисциплины	31.08.2020	
2	Актуализирован раздел 8.2 в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных	31.08.2020	

	и справочных информационных систем		
3	Актуализирован раздел 8.1 Перечень учебной литературы, необходимой для освоения дисциплины	31.08.2021	
4	Внесены изменения в связи с вступлением в силу приказа № 403-ФЗ от 2.12.2019 «О внесении изменений в Федеральный закон об образовании в Российской Федерации» и отдельные законодательные акты Российской Федерации	31.08.2021	