

**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/
подпись
«31» августа 2021 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

«Программирование на Java»

Направление подготовки 09.03.01 Информатика и вычислительная техника

Профиль Программное обеспечение автоматизированных систем

Квалификация выпускника бакалавр

Нормативный период обучения 4 года / 4 года и 11 м.

Форма обучения очная / заочная

Год начала подготовки 2021

Автор программы



/В.В. Сокольников /

**Заведующий кафедрой
Компьютерных
интеллектуальных
технологий проектирования**



Чижов М.И.

Руководитель ОПОП



/В.В. Ветохин/

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины «Программирование на Java» изучение основ программирования на кроссплатформенном языке высокого уровня Java. Повышение интереса к разработке программного обеспечения, формирование творческого подхода к программированию. Приобретение навыков, позволяющих будущим специалистам вести успешную разработку специализированного программного обеспечения в тех областях и сферах деятельности, в которых они будут трудиться.

Изучение дисциплины должно способствовать формированию у студентов основ научного мышления, в том числе: владение основными методами, способами и инструментами создания программного обеспечения, использования для решения практических задач

1.2. Задачи освоения дисциплины

- ознакомление с тенденцией развития программного обеспечения;
- изучения основных структур языка Java;
- изучение операций над структурами в языке Java;
- изучение лексики языка Java;
- обучение работе с современными средствами разработки.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование на Java» относится к дисциплинам блока ФТД.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программирование на Java» направлен на формирование следующих компетенций:

ПК-5 - Способен обеспечивать производственный процесс предприятия программным обеспечением в соответствии с предъявляемыми требованиями

ПК-8 - Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-5	Знать основы программирования на кроссплатформенном языке высокого уровня Java
	Уметь писать программный код в соответствии с особенностями языка Java
	Владеть навыками разработки ПО в соответствии с предъявляемыми требованиями
ПК-8	Знать принципы проектирования систем
	Уметь проектировать системы
	Владеть навыками проектирования систем

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программирование на Java» составляет 2 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		6
Аудиторные занятия (всего)	36	36
В том числе:		
Лекции	18	18
Лабораторные работы (ЛР)	18	18
Самостоятельная работа	36	36
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	72	72
зач.ед.	2	2

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		10
Аудиторные занятия (всего)	8	8
В том числе:		
Лекции	4	4
Лабораторные работы (ЛР)	4	4
Самостоятельная работа	60	60
Часы на контроль	4	4
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	72	72
зач.ед.	2	2

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	Лексика языка Java	Описание деталей внутреннего устройства языка. Лексический анализ программы. Лексемы. Работа с операторами.	1	1	2	4
2	Типы данных языка Java	Переменные. Прimitives и ссылочные типы данных.	1	1	2	4
3	Имена. Пакеты.	Элементы пакета. Модуль компиляции. Область видимости имен. Соглашения по именованию.	1	1	2	4
4	Объявления классов	Модификаторы доступа. Объявления классов. Дополнительные свойства	1	1	3	5

		классов.				
5	Преобразования типов	Виды приведений. Применение приведений. Типы переменной и тип ее значения.	1	1	3	5
6	Объектная модель в Java.	Статические элементы. Ключевые слова this и super. Ключевое слово abstract. Интерфейсы. Объявление интерфейсов. Полиморфизм.	2	2	3	5
7	Массивы	Массивы, как тип данных в Java. Преобразование типов для массивов. Клонирование.	1	1	3	5
8	Операторы и структура кода. Исключения	Управление ходом программы. Нормальное и прерванное выполнение операторов. Блоки и локальные переменные. Пустой оператор. Метки. Оператор if. Оператор switch. Управление циклами. Операторы break и continue. Именованные блоки. Оператор return. Оператор synchronized. Ошибки при работе программы. Исключения (Exceptions).	2	2	3	5
9	Потоки выполнения. Синхронизация	Многопоточная архитектура. Базовые классы для работы с потоками. Синхронизация. Методы wait(), notify(), notifyAll() класса Object	2	2	3	5
10	Пакет java.lang	Object. Class. Wrapper Classes. Math. Системные классы. Потоки исполнения. Исключения	1	1	3	5
11	Пакет java.util	Работа с датами и. Интерфейс Observer и класс Observable. Коллекции. Класс Properties. Интерфейс Comparator. Класс Arrays. Класс StringTokenizer. Класс BitSet. Класс Random. Локализация	1	1	3	5
12	Пакет java.io	Система ввода/вывода. Потоки данных (stream). Serialization. Классы Reader и Writer. Их наследники. Класс StreamTokenizer. Работа с файловой системой.	2	2	3	5
13	Введение в сетевые протоколы	. Основы модели OSI. Physical layer (layer 1). Data layer (layer 2). Network layer (layer 3). Transport layer (layer 4). Session layer (layer 5). Presentation layer (layer 6). Application layer (layer 7). Утилиты для работы с сетью. Пакет java.net.	2	2	3	5
Итого			18	18	36	72

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Лексика языка Java	Описание деталей внутреннего устройства языка. Лексический анализ программы. Лексемы. Работа с операторами.	0.5	0.5	5	6
2	Типы данных языка Java	Переменные. Примитивные и ссылочные типы данных.	0.5	0.5	5	6
3	Имена. Пакеты.	Элементы пакета. Модуль компиляции. Область видимости имен. Соглашения по именованию.	0.5	0.5	5	6

4	Объявления классов	Модификаторы доступа. Объявления классов. Дополнительные свойства классов.	0.5	0.5	5	6
5	Преобразования типов	Виды приведений. Применение приведений. Типы переменной и тип ее значения.	0.5	0.5	10	11
6	Объектная модель в Java.	Статические элементы. Ключевые слова this и super. Ключевое слово abstract. Интерфейсы. Объявление интерфейсов. Полиморфизм.	0.5	0.5	10	11
7	Массивы	Массивы, как тип данных в Java. Преобразование типов для массивов. Клонирование.	0.5	0.5	10	11
8	Операторы и структура кода. Исключения	Управление ходом программы. Нормальное и прерванное выполнение операторов. Блоки и локальные переменные. Пустой оператор. Метки. Оператор if. Оператор switch. Управление циклами. Операторы break и continue. Именованные блоки. Оператор return. Оператор synchronized. Ошибки при работе программы. Исключения (Exceptions).	0.5	0.5	10	11
Итого			4	4	60	68

5.2 Перечень лабораторных работ

Укажите перечень лабораторных работ

1. Консольное приложение
2. Разработка приложения – “Калькулятор”
3. Разработка простейшего
4. Ошибки при работе программы. Исключения
5. Работа с потоками
6. Работа с пакетом java.net

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

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

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».

Компе-	Результаты обучения,	Критерии	Аттестован	Не аттестован
--------	----------------------	----------	------------	---------------

компетенция	характеризующие сформированность компетенции	критерии оценивания	Зачтено	Не зачтено
ПК-5	Знать основы программирования на кроссплатформенном языке высокого уровня Java	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь писать программный код в соответствии с особенностями языка Java	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть навыками разработки ПО в соответствии с предъявляемыми требованиями	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-8	Знать принципы проектирования систем	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь проектировать системы	Количество выполненных лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть навыками проектирования систем	Количество выполненных лабораторных работ	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, 10 семестре для заочной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-5	Знать основы программирования на кроссплатформенном языке высокого уровня Java	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь писать программный код в соответствии с особенностями языка Java	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками разработки ПО в соответствии с	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в	Задачи не решены

	предъявляемыми требованиями		большинстве задач	
ПК-8	Знать принципы проектирования систем	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь проектировать системы	Решение стандартных практических задач	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены
	Владеть навыками проектирования систем	Решение прикладных задач в конкретной предметной области	Продемонстрировать верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые 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. Как записать в Java-программе символ с кодом 514?

- a.) '514'
- b.) \u0546
- c.) \u222
- ✓d.) \u0222

2. Сколько пробелов в следующем примере кода:

```
int x = 3; int y=1;
int z = x+y;
```

- a.) 7
- b.) 8
- ✓c.) 9
- d.) 11

3. Сколько комментариев в следующем примере кода:

```
int x = 0; /* text // text */
int y=1; // text */ // text */
```

- a.) 1 блочный, 0 строчных
- ✓b.) 1 блочный, 1 строчный
- c.) 1 блочный, 2 строчных
- d.) 2 блочных, 3 строчных

4. Какие из перечисленных идентификаторов являются корректными?

- ✓a.) abc
- b.) 1ab
- ✓c.) _bc
- ✓d.) _1c \$ac
- ✓e.) \$ac
- ✓f.) for_
- ✓g.) Int
- h.) byte

5. Являются ли следующие слова ключевыми:

- a.) true

- ✓b.) goto
- c.) null
- ✓d.) const
- e.) false

6. Равны ли следующие числа:

- ✓a.) 5 и 05
- b.) 9 и 09
- c.) 10 и 010
- ✓d.) 0x5A и 90L

7. Чему будет равно следующее выражение и значение переменной x после вычислений?

```
int x=0;  
print(++x==x++);
```

- a.) Выражение истинно. ✓
- b.) Выражение ложно

8. Чему будет равно следующее выражение и значения переменных x и y после вычислений?

```
int x=0, y=0;  
print((++x==1) || (y++==1));
```

- a.) Выражение истинно ✓
- b.) Выражение ложно

9. Эквиваленты ли две следующие операции над ссылочными переменными x1 и x2 (SomeClass2 – тип переменной x2)?

```
x1 instanceof SomeClass2  
x1.getClass().getName().equals(x2.getClass().getName())
```

- ✓a.) Ответ «нет».
- б.) Ответ «да».

10. Ниже приведено несколько вариантов записи модуля компиляции. Какие из них корректны, если предполагается описать класс Point из пакета test.demo, причем класс активно использует классы java.awt.Point и несколько классов из пакета java.net?

a)
package test.demo;
import java.awt.Point;
import java.net.*;

b)
import java.awt.*;
import java.net.*;
package test.demo;

✓c)
package test.demo;
import java.net.*;
import java.awt.*;

d)

```
package test.demo.*  
import java.net.*;  
import java.awt.*;
```

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какие модификаторы позволяют обращаться к элементу из классов того же пакета?

а.) Модификаторы `public` и `protected`, а также доступ по умолчанию (без указания модификатора).

2. Если в классе заводится новый элемент, и пока нет никаких факторов, позволяющих выбрать тот или иной модификатор доступа. Какой модификатор использовать в таком случае?

а.) Лучше использовать `private`. Раз не требуется внести этот элемент в открытый интерфейс класса, значит, стоит его отнести к реализации. В дальнейшем при необходимости всегда можно расширить уровень доступа. Обратное действие – сужение – запрещено.

3. Пусть класс `User` описывает пользователя системы. В качестве имени используется его e-mail адрес, который всем известен, а пароль, конечно, не должен быть никому доступен, кроме самого пользователя. Корректна ли следующая реализация?

```
public class User {  
    public String login; // открытый e-mail  
    private String password; // закрытый пароль  
}
```

а.) Хотя код корректен с точки зрения компилятора, он не верен с точки зрения ООП дизайна. В предложенном варианте любой класс может изменить значение `login` у пользователя, что вряд ли соответствует задуманному алгоритму работы системы. Рекомендуется закрыть доступ к полю `login` извне и добавить метод для чтения, который бы возвращал значение `login`.

Например:

```
public class User {  
    private String login; // открытый e-mail  
    private String password; // закрытый пароль  
    public String getLogin(){  
        return login;  
    }  
}
```

4. Корректен ли следующий код?

```
public class Test {  
    private int id;  
    public Test(int i) {  
        id=i;  
    }  
    public static boolean test(Test t, int id) {  
        return t.id==id;  
    }  
}
```

}

а.) Да. Метод test является методом класса Test, а значит, имеет доступ к полю id любого объекта этого класса.

5. Из каких частей состоит заголовок объявления класса? Тело класса?

а.) Заголовок класса (именно в таком порядке):

1. опциональные модификаторы (public, abstract, final)
2. ключевое слово class и имя класса
3. опционально ключевое слово extends и имя суперкласса
4. опционально ключевое слово implements и список имен

реализуемых

интерфейсов

5. тело класса в фигурных скобках

Тело класса (в произвольном порядке):

- поля
- методы
- внутренние типы (классы и интерфейсы)
- конструкторы
- инициализаторы

6. Если метод использует переменную класса, должна ли она быть объявлена выше объявления метода?

а.) Нет. Областью видимости переменной класса является все объявление тела класса.

7. Из каких частей состоит заголовок объявления метода? Какие части обязательные?

а.) Заголовок метода (именно в таком порядке):

1. опциональные модификаторы (доступа public|private|protected и прочие static, final, native, synchronized)
2. тип возвращаемого значения или void, если его нет
3. имя метода
4. список типов и имен аргументов в круглых скобках
5. опционально throws выражение

Обязательно должны присутствовать имя метода, тип возвращаемого значения (или void), перечисление аргументов в круглых скобках (или пустые скобки, если их нет).

8. Пусть класс должен обладать методом со следующими вариациями. Метод должен принимать в качестве аргумента дробное значение типа double или float и возвращать результат округления. В случае float нужно возвращать либо «короткое» значение (byte), либо «полное» (int). Аналогично для double – int или long соответственно. Сколько методов должно быть объявлено в таком классе, и каковы их сигнатуры?

а.) Если формально следовать вопросу, то должно быть объявлено 4 метода:

```
byte round(float x) { ... }  
int roundToInt(float x) { ... }  
int round(double x) { ... }
```

```
long roundToLong(double x) { ... }
```

9. Может ли класс не иметь ни одного конструктора?

а.) Нет, такой класс объявить невозможно. Даже если не указать ни одного конструктора, компилятор добавит конструктор по умолчанию. Если возникнет противоречие с родительским классом (ведь конструктор по умолчанию требует наличие доступного конструктора без параметров у родительского класса), то возникнет ошибка, и класс не будет скомпилирован.

10. Что появится на консоли при вызове конструктора следующего класса?

```
class Test {  
    private long id=getId();  
    private String name=getName();  
    private String login;  
    public Test(int domain) {  
        login=domain+" "+name;  
        System.out.println(login);  
    }  
    private static long getId() {  
        int id = 3;  
        System.out.println(id);  
        return id;  
    }  
    private String getName() {  
        String name="name"+getId();  
        System.out.println(name);  
        return name; } }  
    а.) Результатом будет:
```

```
3  
3  
name3  
5 name3
```

11. Как записывается заголовок метода main?

а.) public static void main(String[] args)

12. Может ли измениться содержимое переменной типа String, если передать ее в качестве аргумента при вызове метода?

а.) Нет, так как класс String, а стало быть, и его объекты, не изменяемы.

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Корректен ли следующий пример кода, и если да, то сколько преобразований и каких видов будет произведено при его исполнении?

```
byte b=1;  
long m=-b;  
Object o="=";  
int("m"+o+m);
```

а.) Пример корректен. Преобразования:

byte

- первая строка: преобразование при присвоении, сужение от int к byte
- вторая: числовое расширение при вычислении оператора унарный минус от byte к int, расширение при присвоении значения от int к long
- третья: расширение при присвоении String к Object
- четвертая: приведения к строке: от Object к String и от long к String (через Long).

2. Произойдет ли потеря точности при следующем преобразовании?

```
float f = -16777217;
```

a.) Да. Число 16777217 записывается как 1000000000000000000000001 в двоичной системе, всего 25 бит. А тип float отводит 24 бита для хранения мантиссы (значащих цифр), причем один из них знаковый. После отбрасывания не уместившихся битов получаем 1000000000000000000000000, или -16777216 в десятичной системе. Таким образом, теряется точность в последнем разряде. (Для получения правильного ответа достаточно запустить подобный пример и увидеть результат.)

3. Корректен ли следующий пример? Если нет, то в каких строках какие ошибки будут сгенерированы?

```
byte b=100-100;
```

```
byte b=100+100;
```

```
byte b=100*100;
```

a.) Результат вычитания в первой строке равен 0, что укладывается в тип byte, ошибки не будет. Результат вычислений в строках 2 и 3 выходит за рамки byte (максимально допустимое значение 127), поэтому неявное сужение не допустимо.

4. Для каких значений числовой переменной x будет верно следующее выражение?

```
x== -x
```

a.) Для целочисленных типов:

- 0

- самое наименьшее значение для типов int и long. Например, для типа int это -2147483648

Для дробных типов -

- значения 0.0 и -0.0

5. Верны ли следующие выражения для переменной d дробного типа?

```
(short)d==(short)(int)d
```

```
(int)d==(int)(long)d
```

a.) Первое выражение верно, так как при приведении к short, дробное значение всегда сначала приводится к int.

Второе выражение не верно, в чем можно легко убедиться, вычислив его для любого d, большего, чем максимально допустимое значение типа int.

Левая часть будет равна 231-1, в то время как правая будет вычислена путем отбрасывания битов, старше 32:

```
double d=3e9;
System.out.println((int)d);
System.out.println((int)(long)d);
```

Результатом будет:
2147483647
-1294967296

6. Корректны ли следующие преобразования?

```
Object o = (String)null;
String s = o;
```

а.) Нет. Хотя значение null можно привести к любому ссылочному типу, во 2 строке происходит сужение типа ссылки (Object) до класса String. Такое действие должно происходить явно:

```
String s = (String)o;
```

7. Пусть классы Wolf и Rabbit являются наследниками класса Animal. Корректен ли следующий пример?

```
Wolf w = new Wolf();
Animal a = (Animal)w;
Rabbit r = (Rabbit)a;
```

а.) Хотя этот пример будет корректно скомпилирован, при выполнении он всегда будет порождать ошибку. Не смотря на приведение, переменная «a» ссылается на объект типа Wolf, созданный в первой строке.

В третьей строке при приведении ссылки на объект класса Wolf к типу Rabbit будет возникать ошибка преобразования типов, поскольку эти классы не находятся на одной ветви дерева наследования.

8. Какие значения не могут участвовать в преобразовании к строке?

а.) Все могут.

9. Какие преобразования будут производиться при работе следующего метода?

```
public int add(byte a, byte b) {
    short x=(short)a;
    char y=(char)b;
    return x+y;
}
```

а.) Преобразования:

- первая строка: явное приведение от byte к short
- вторая: явное приведение от byte к char
- третья: числовое расширение при вычислении бинарного оператора сложения: от short и char к int

10. Какое значение появится на консоли после выполнения следующего кода?

```
char c=65;
print(c);
print(+c);
print(“=”+c);
```

а.) Ответ:

A
65
=A

11. Значение какого типа будет хранить переменная после инициализации?

byte b=1+2;

а.) Примитивные переменные всегда хранят значения точного того же типа, что и они сами, то есть, в данном примере – byte. Это значение будет получено после неявного сужения при присвоении от типа int.

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Типы данных в языке программирования Java
2. Правила объявления классов и интерфейсов
3. Массивы в языке программирования Java
4. Управление ходом выполнения программы
5. Работа с ошибками в языке программирования Java
6. Возможности в языке программирования Java создания многопоточных приложений
7. Контейнеры и коллекции в языке программирования Java
8. Возможности в языке программирования Java для обмена и передачи информации
9. Работа с файлами в языке программирования Java
10. Работа с сетью в языке программирования Java

7.2.5 Примерный перечень заданий для решения прикладных задач

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет 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 «Не зачтено» ставится в случае, если студент набрал менее 11 баллов.

2. Оценка «Зачтено» ставится в случае, если студент набрал от 11 до 20 баллов.)

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Лексика языка Java	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
2	Типы данных языка Java	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
3	Имена. Пакеты.	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ

4	Объявления классов	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
5	Преобразования типов	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
6	Объектная модель в Java.	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
7	Массивы	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
8	Операторы и структура кода. Исключения	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
9	Потоки выполнения. Синхронизация	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
10	Пакет java.lang	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
11	Пакет java.util	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
12	Пакет java.io	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ
13	Введение в сетевые протоколы	ПК-5, ПК-8	Тест, контрольная работа, защита лабораторных работ

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 методике 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 методике 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 методике выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Блох Дж. Java. Эффективное программирование / Блох Дж.. — Саратов : Профобразование, 2019. — 310 с. — ISBN 978-5-4488-0127-3. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89870.html>

2. Мухамедзянов Р.Р. JAVA. Серверные приложения / Мухамедзянов Р.Р.. — Москва : СОЛОН-Р, 2016. — 336 с. — ISBN 5-93455-134-5. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/90352.html>

3. Вязовик Н.А. Программирование на Java : учебное пособие / Вязовик Н.А.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 601 с. — ISBN 978-5-4497-0852-6. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102048.html>

4. Окулов С.М. Программирование в алгоритмах / Окулов С.М.. — Москва : Лаборатория знаний, 2021. — 384 с. — ISBN 978-5-93208-521-9. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/105770.html>

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:

Лицензионное ПО:

- Microsoft Word
- IntelliJ IDEA Ultimate

Свободное программное обеспечение:

- LibreOffice

Отечественное ПО:

- СУБД Линтер

Ресурсы информационно-телекоммуникационной сети «Интернет»:

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

Информационные справочные системы:

- <http://window.edu.ru>
- <https://wiki.cchgeu.ru/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

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

Учебные лаборатории (г. Воронеж, ул. Плехановская, д. 11):

- “Компьютерное моделирование и дизайн”.
- “Интеллектуальные системы проектирования”.

Дисплейный класс, оснащенный компьютерными программами для проведения лабораторного практикума.

Кабинеты, оборудованные проекторами и интерактивными досками.

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

По дисциплине «Программирование на Java» читаются лекции, проводятся лабораторные работы.

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.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать 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 усвоения 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 зачетом три дня эффективнее всего использовать для повторения и систематизации материала.