

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.2.30.1 CASE ТЕХНОЛОГИИ В JAVA

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Вычислительные машины, комплексы, системы и сети»

Квалификация (степень) выпускника бакалавр

Форма обучения очная

Пенза, 2016

1. Цели освоения дисциплины

Целью изучения дисциплины является изучение современных технологий, используемых при разработке программного обеспечения с использованием языка программирования Java.

2. Место дисциплины в структуре ООП

2.1. Дисциплина относится к дисциплинам по выбору вариативной части блока Б1 программы бакалавриата по направлению 09.03.01 «Информатика и ВТ» и базируется на следующих курсах: «Информатика», «Программирование», «Операционные системы».

Дисциплина является предшествующей для выполнения квалификационной работы бакалавра.

2.2. Минимальные требования к «входным» знаниям, необходимым для успешного усвоения данной дисциплины - удовлетворительное усвоение программ по следующим разделам указанных выше дисциплин:

- «Информатика» в полном объеме;
- «Программирование» в полном объеме.
- «Операционные системы» в полном объеме.

2.3. На основе знаний, полученных в ходе практических занятий формируются навыки владения языком объектно-ориентированного программирования Java, а также технологиями разработки сложных программных систем.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Коды компетенции	Наименование компетенции	Структурные элементы компетенции (в результате освоения дисциплины обучающийся должен знать, уметь, владеть)
1	2	3
ПК-1	способностью разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек - электронно-вычислительная машина»	Знать основные классы языка Java для описания интерфейсов «человек-машина».
		Уметь разрабатывать приложения на языке программирования Java.
		Владеть основными методами проектирования программных компонентов с использованием языка Java.

4. Структура и содержание дисциплины (модуля)

4.1. Структура дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов.

№ п/п	Наименование разделов и тем дисциплины (модуля)	Семестр	Недели семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)								Формы текущего контроля успеваемости (по неделям семестра)						
				Аудиторная работа				Самостоятельная работа				Собеседование	Коллоквиум	Проверка тестов	Проверка контрольн. работ	Проверка реферата	Проверка эссе и иных творческих работ	курсовая работа (проект)
				Всего	Лекция	Практические занятия	Лабораторные занятия	Всего	Подготовка к аудиторным занятиям	Реферат, эссе и др.	Курсовая работа (проект)							
1.	Раздел 1. Введение			1	1			2				2	2					
1.1.	Тема 1.1. Вводная лекция			1	1			2				2	2					
2.	Раздел 2. Программирование с использованием языка Java			34	16		18	68	36			32	3-8					
2.1.	Тема 2.1. Операции и операторы в языке Java.			2	2			10	6			4	3					
2.2.	Тема 2.2. Реализация принципов объектно-ориентированного подхода в Java			4	2		2	10	6			4	4					
2.3	Тема 2.3. Элементы хранения наборов данных			7	3		4	12	6			6	4					
2.4	Тема 2.4. Обработка исключительных ситуаций			7	3		4	12	6			6	6					
2.5	Тема 2.5. Ввод/вывод в Java			7	3		4	12	6			6	7					

2.6	Тема 2.6. Многопоточность в Java			7	3		4	12	6			6	8						
3	Раздел 3. Заключение			1	1			2				2	8						
3.1	Тема 3.1. Перспективы развития языка Java.			1	1			2				2	8						
	Общая трудоемкость, в часах			36	18		18	72	36			36	Промежуточная аттестация						
													Форма		Семестр				
													Зачет		8				
													Экзамен		8				

4.2. Содержание дисциплины (модуля)

4.2.1. Содержание лекционного курса

Раздел 1. Введение

Тема 1.1. Вводная лекция

Цели и задачи курса и его место в подготовке бакалавра. Этапы и перспективы изучения дисциплины. Обзор литературы. Сравнение языка C++ с языком Java. Простейшая программа на Java.

Раздел 2. Программирование с использованием языка Java

Тема 2.1. Операции и операторы в языке Java.

Операции сравнения. Операции инкремента, декремента. Расширенные операции присваивания. Логические операции. Операции сдвига. Операция приведения типов. Оператор — выражение. Условный оператор. Операторы цикла. Оператор цикла "foreach". Операторы break и continue.

Тема 2.2. Реализация принципов объектно-ориентированного подхода в Java.

Ссылки на объекты. Элементарные типы данных. Область видимости и время жизни переменных. Область видимости и время жизни объектов. Описание методов класса. Передача параметров. Наследование классов. Класс Object. Инициализация полей при наследовании классов. Модификаторы доступа при наследовании. Преобразования типов (классов) при наследовании. Полиморфизм. Ключевое слово final. Абстрактные классы. Интерфейсы. Вложенные классы. Анонимные вложенные классы. Статические вложенные классы.

Тема 2.3. Элементы хранения наборов данных.

Массивы в Java. Массивы объектов. Многомерные массивы. Присваивание и копирование массивов. Работа со строками. Коллекции объектов Java. Коллекции как наборы произвольных объектов. Итераторы. Коллекции-списки. Коллекции-множества. Задание порядка элементов в коллекциях. Интерфейс Comparator. Коллекции-ассоциативные массивы.

Тема 2.4. Обработка исключительных ситуаций.

Поведение программы при возникновении исключения. Структура и использование блока перехвата исключений. Классы исключительных ситуаций. Механизм контроля перехвата исключений. Генерация исключительных ситуаций. Создание собственных классов исключительных ситуаций. Исключительные ситуации и наследованием.

Тема 2.5. Ввод/вывод в Java.

Иерархия InputStream. Иерархия Reader. Иерархия OutputStream. Иерархия Writer. Класс RandomAccessFile. Класс File. Сериализация объектов. Интерфейс Serializable. Классы ObjectOutputStream и ObjectInputStream.

Тема 2.6. Многопоточность в Java.

Интерфейс Runnable. Класса Thread. Завершение процесса и демоны. Завершение потоков. Механизм Interruption. Метод sleep. Метод yield. Метод join. Приоритеты потоков. Блокировки. Методы wait, notify, notifyAll.

Раздел 3. Заключение

Тема 3.1. Перспективы развития языка Java. Обзор курса.

4.2.2. Перечень и содержание лабораторных занятий.

№ п/п	№ темы	Наименование лабораторных работ	Кол. ч
1	2.2	Графические интерфейсы	2
2	2.3	Работа с коллекциями объектов	4
3	3.4	Обработка исключительных ситуаций	4
4	2.5	Работа с файлами	4
5	2.6	Многопоточность в Java	4

5. Образовательные технологии

5.1 Чтение лекций по дисциплине проводится с использованием мультимедийного компьютерного проектора с раздачей демонстрируемых слайдов комментариев.

5.2 При изучении материалов лабораторного практикума использовать Интернет ресурсы с сайта кафедры ВТ (alice.pnzgu.ru и titan.vt).

5.3 При самостоятельной работе используются материалы сайта «Интернет-Университет Информационных Технологий (www.intuit.ru).

5.4. Все лабораторные занятия носят проектный характер.

5.5 В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по собственной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины базируется на следующих возможностях: обеспечение внеаудиторной работы со студентами, в том числе в электронной образовательной среде с использованием соответствующего программного оборудования, дистанционных форм обучения, возможностей интернет-ресурсов, индивидуальных консультаций и т.д.

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

6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы	Задание	Рекомендуемая литература	Количество
2	Тема 2.1. Операции и операторы в языке Java.	Подготовка к аудиторным занятиям	Изучить операции в языке Java	Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет	6

				Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: http://www.bibliocomplectator.ru/book/?id=39552	
4	Тема 2.2. Реализация принципов объектно-ориентированного подхода в Java	Подготовка к аудиторным занятиям	Изучить принципы объектно-ориентированного подхода в Java	Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: http://www.bibliocomplectator.ru/book/?id=39552	6
6	Тема 2.3. Элементы хранения наборов данных	Подготовка к аудиторным занятиям	Изучить особенности хранения наборов данных в Java	Мейер Б. Объектно-ориентированное программирование и	6

				<p>программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: http://www.bibliocomplectator.ru/book/?id=39552</p>	
7	<p>Тема 2.4. Обработка исключительных ситуаций</p>	<p>Подготовка к аудиторным занятиям</p>	<p>Изучить принципы организации обработки исключительных ситуаций в Java</p>	<p>Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа:</p>	6

				http://www.bibliocomplector.ru/book/?id=39552	
8	Тема 2.5. Ввод/вывод в Java	Подготовка к аудиторным занятиям	Изучить способы организации ввода-вывода в Java.	Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: http://www.bibliocomplector.ru/book/?id=39552	6
9	Тема 2.6. Многопоточность в Java	Подготовка к аудиторным занятиям	Изучить особенности реализации параллельных алгоритмов на языке Java.	Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые	6

				данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: http://www.bibliocomplectator.ru/book/?id=39552	
--	--	--	--	---	--

6.2. Методические указания по организации самостоятельной работы студентов

Планируются следующие виды самостоятельной работы (внеаудиторной) относятся:

- подготовка к лабораторным работам занятиям,
- оформление отчётов по лабораторным работам,
- подготовка к зачету,
- работа с конспектом лекций и изучение рекомендованной литературы при подготовке к экзаменам.

6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

Контроль освоения компетенций

№ п\п	Вид контроля	Контролируемые разделы	Компетенции, компоненты которых контролируются
1	Текущий: собеседование при защите лаб. заданий Промежуточный: зачет, экзамен	Раздел 2 Программирование с использованием языка Java	ПК-1

Контроль освоения компетенции выполняется:

– для компетенции ПК-1 – путем оценки качества выполненных лабораторных работ и в ходе экзамена.

Примерный перечень вопросов для собеседования

Вопрос 1. Дан следующий код:

```
abstract class A{
    public abstract void print();
}
class B extends A{
    public void print(){
        System.out.println("B");
    }
}
```

Можно ли унаследовать от класса В абстрактный класс? Если да, то что нужно сделать с методом print(): его обязательно надо переопределить, его можно не переопределять, можно ли его определить как abstract?

Вопрос 2. Дан следующий код:

```
abstract class A {
    int p1;
    A() {
        p1 = 1;
    }
    ...
}
class B extends A {
}
```

- 1) Можно ли описать конструкторы в абстрактном классе?
- 2) Может ли быть класс абстрактным без единого абстрактного метода? А наоборот?
- 3) При определении метода в интерфейсе подразумевается модификатор public и abstract. Можно ли при определении интерфейса явно указать модификатор abstract?

Вопрос 3. Дан следующий метод:

```
public void method() throws IllegalArgumentException,IOException,IOException {
    //...
}
```

Возникнет ли ошибка компиляции? Можно ли указывать `unchecked` исключения в секции `throws`? Что будет, если в секции `throws` одно и то же исключение указано несколько раз? Можно ли смешивать `checked` и `unchecked` исключения?

Вопрос 4. Дан следующий код:

```
1: class Formatter {
2:   public String format(String value) {
3:     return "["+value+>";
4:   }
5: }
6:
7: public class TestNPE {
8:   public static String handle(Formatter f, String s) {
9:     if(s.isEmpty()) {
10:      return "(none)";
11:    }
12:    return f.format(s.trim());
13:  }
14: }
```

Откуда-то был вызван метод `handle` с какими-то параметрами, и было получено:

```
Exception in thread "main" java.lang.NullPointerException
    at TestNPE.handle(TestNPE.java:12)
```

В чём причина исключения — в `f` или `s`? Чтобы поменяться, если бы в девятой строчке было бы написано так:

```
if("".equals(s))
```

Вопрос 5. Какой из предложенных ниже методов не вызовет ошибку компиляции?

```
1) public static double sqr(double arg) {
    }
```

```

2) public static double sqr(double arg) {
    while (true);
}
3) public static double sqr(double arg) {
    if (System.currentTimeMillis() % 2 == 0) {
        return arg * arg;
    }
}
4) public double sqr(double arg) {
    int k = 1;
    return k;
}
5) public static double sqr(double arg) {
    throw new RuntimeException();
}

```

Вопрос 6. Как сделать Singleton потоко безопасным?

Вопрос 7. Дан такой код:

```

class Test{
    public static void main(String[] args){
        Map map = new TestMap();
        for(Object o: map.keySet()){
            System.out.println(o);
        }
    }
}

class TestMap extends HashMap{
    @Override
    public Set keySet(){
        return null;
    }
}

```

Какой результат выполнения данного кода?

Вопрос 8. Что получим после выполнения кода:

```

int a = 0xFF0;
int b = 0xF0F;
int c = 0x0FF;

```

```
c = a = (a ^ b | a & b) ^ c;  
System.out.println((c | a) ^ (c & a));
```

Вопрос 9. Например вы не знаете как отсортировать лист чисел и не хотите писать сортировку для листа, то очень удобно произвести конвертирование с листа в массив, вызвать сортировку для массива, и осталось вернуться обратного к листу. самый очевидный способ использовать циклы, но допустим вам сказали обойтись без циклов и ваших подпрограмм.

Существует ли метод для преобразования такого типа: `int[] -> ArrayList<Integer>`. Если да, назовите его.

Вопрос 10. 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() + ".");
```

Вопрос 11. Дан код:

```
public interface A {  
    String text = "a";  
}  
  
public interface B {  
    String text = "b";  
}  
  
public class Implementor implements A, B {  
    public static void main(String[] args) {  
        System.out.println(Implementor.text);  
    }  
}
```

Какое значение примет переменная `text` в классе `Implementor`? Что выведется на экран?

Вопрос 12. Дан следующий код. Как вы думаете, что произойдет в результате выполнения данного кода?

```
public class Test {  
    class A {
```



```

String str = "ab";

A() {
    printLength();
}

void printLength() {
    System.out.println(str.length());
}

class B extends A {
    String str = "abc";

    void printLength() {
        System.out.println(str.length());
    }
}

public static void main(String[] args) {
    new Test().new B();
}
}

```

Вопрос 13. Измените код так, чтобы он вернул hello world на консоль.

```

public class NullReferenceTest {
    //.....
    public static void main(String[] args) {
        NullReferenceTest nullReferenceTest = null;
        System.out.println(nullReferenceTest./*<some descriptor>*/);
    }
}

```

Ограничения - нельзя создавать новые объекты. Нельзя изменять текущий код, можно только добавить немного своего.

Вопрос 14. Дан код:

```

public class HelloWorld {

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

```


Вопрос 20. Почему String в Java является immutable?

Вопрос 21. Что будет выведено в консоли?

```
int i = 5;  
i = i++;  
System.out.println(i);
```

Вопрос 22. Что будет выведено в консоли?

```
i = 5;  
i = i++ + i++;  
System.out.println(i);
```

Вопрос 23. Что будет выведено в консоли?

```
i = 5;  
i = i++ + ++i;  
System.out.println(i);
```

Вопрос 25. Что будет выведено в консоли?

```
i = 5;  
i = ++i + ++i;  
System.out.println(i);
```

Вопрос 26. Сравните 2 фрагмента кода:

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

Будет ли разница при исполнении этих строк кода?

Вопрос 27. Опишите механизм добавления элемента в HashMap

Вопрос 28. Что такое " **Deadlock**" (взаимная блокировка)? Причины возникновения? Методы предотвращения?

Вопрос 29. Дан класс точка, предложите реализацию метода equals, чтобы обеспечить корректность сравнения, в том числе и математического.

```
public class Point {  
    private double x;  
    private double y;  
  
    public Point(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
  
    @Override  
    public boolean equals(Object obj) {  
  
    }  
  
    @Override  
    public int hashCode() {  
  
    }  
}
```

Вопрос 30. Объясните чем отличается &| и &&, ||. Какой рекомендуется использовать и почему?

Примерный перечень вопросов и заданий к экзамену

1. Сравнение языка C++ с языком Java. Простейшая программа на Java.
2. Операции и операторы в языке Java. Операции сравнения. Операции инкремента, декремента. Расширенные операции присваивания.
3. Операции и операторы в языке Java. Логические операции. Операции сдвига. Операция приведения типов. Оператор — выражение.
4. Операции и операторы в языке Java. Условный оператор. Операторы цикла. Оператор цикла "foreach". Операторы break и continue.
5. Реализация принципов объектно-ориентированного подхода в Java. Ссылки на объекты. Элементарные типы данных.

6. Область видимости и время жизни переменных. Область видимости и время жизни объектов. Описание методов класса. Передача параметров.
7. Наследование классов. Класс Object. Инициализация полей при наследовании классов. Модификаторы доступа при наследовании. Преобразования типов (классов) при наследовании.
8. Полиморфизм. Ключевое слово final. Абстрактные классы. Интерфейсы.
9. Вложенные классы. Анонимные вложенные классы. Статические вложенные классы.
10. Массивы в Java. Массивы объектов. Многомерные массивы. Присваивание и копирование массивов.
11. Работа со строками.
12. Коллекции объектов Java. Итераторы.
13. Коллекции-списки.
14. Коллекции-множества.
15. Задание порядка элементов в коллекциях. Интерфейс Comparator.
16. Коллекции-ассоциативные массивы.
17. Обработка исключительных ситуаций. Поведение программы при возникновении исключения.
18. Структура и использование блока перехвата исключений. Классы исключительных ситуаций.
19. Механизм контроля перехвата исключений. Генерация исключительных ситуаций.
20. Создание собственных классов исключительных ситуаций. Исключительные ситуации и наследование.
21. Ввод/вывод в Java. Иерархия InputStream. Иерархия Reader.
22. Ввод/вывод в Java. Иерархия OutputStream. Иерархия Writer.
23. Ввод/вывод в Java. Класс RandomAccessFile. Класс File.
24. Сериализация объектов. Интерфейс Serializable. Классы ObjectOutputStream и ObjectInputStream.
25. Многопоточность в Java. Класс Thread. Интерфейс Runnable.

26. Завершение процесса и демоны. Завершение потоков. Механизм Interruption.
27. Метод sleep. Метод yield. Метод join. Приоритеты потоков.
28. Блокировки. Методы wait, notify, notifyAll.

7. Учебно-методическое и информационное обеспечение дисциплины (модуля)

7.1. Основная литература:

1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.— Режим доступа: <http://www.bibliocomplectator.ru/book/?id=39552>.

7.2. Дополнительная литература:

1. Кознов Д.В. Введение в программную инженерию [Электронный ресурс]/ Кознов Д.В.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 306 с.— Режим доступа: <http://www.bibliocomplectator.ru/book/?id=52146>.
2. Костюкова Н.И. Графы и их применение [Электронный ресурс]/ Костюкова Н.И.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 147 с.— Режим доступа: <http://www.bibliocomplectator.ru/book/?id=52185>.

7.3. Интернет-ресурсы

1. Сайт «Национальный Открытый Университет ИНТУИТ» Раздел «Углубленное программирование на Java», <http://www.intuit.ru/studies/courses/3711/953/info>
2. Сайт «Study JAVA» <http://study-java.ru>

7.4 Программное обеспечение:

1. Среда разработки ПО для выполнения обязательных лабораторных работ: NetBeans 7.0;

2. Среда разработки отчетов по выполненным лабораторным работам:
пакет Open Office;
3. Терминальный клиент, функционирующий в среде ОС Windows.

8. Материально-техническое обеспечение дисциплины (модуля)

Лабораторные занятия проводятся в классе, оснащённом ПЭВМ, с операционной системой Windows XP и средой разработки Microsoft Visual Studio 2005 или более новой версии.

Для лиц с ограниченными возможностями здоровья по ходатайству заведующего кафедрой устанавливается специальный индивидуальный набор программного обеспечения (Scype, Viber и т.д.) на вычислительную технику, выделенную для освоения дисциплины для лица с ограниченными возможностями здоровья.

Рабочая программа дисциплины «Case технологии в java» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки «Информатика и вычислительная техника».

Программу составил:

Доцент кафедры ВТ



Д.А. Трокоз

Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.

Программа одобрена на заседании кафедры ВТ

Протокол № 7 от «15» 02 2016 года

Зав. кафедрой ВТ



Д.В. Пащенко

Программа одобрена методической комиссией ФВТ

Протокол № 4 от «15» 02 2016 года

Председатель методической комиссии ФВТ



Н.Н. Коннов

Сведения о переутверждении программы на очередной учебный год и регистрации изменений

Учебный год	Решение кафедры (№ протокола, дата, подпись зав. кафедрой)	Внесенные изменения	Номера листов (страниц)		
			заменен- ных	новых	аннулиро- ванных
2016/17	№1, 05.09.16г.	Без изменений	1	0	0
2017/18	№1, 06.09.17г.	Без изменений	1	0	0
2018/19	№14, 06.07.18	Без измен.	1	—	—