

5817

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени В.Ф.УТКИНА**

ВВЕДЕНИЕ В ПРОМЫШЕННУЮ РАЗРАБОТКУ ПО НА ПЛАТФОРМЕ JAVA

Методические указания
к лабораторным работам
и практическим занятиям

Рязань 2020

УДК 004.9+657

Введение в промышленную разработку ПО на платформе Java: методические указания к лабораторным работам и практическим занятиям / Рязан. гос. радиотехн. ун-т им. В.Ф.Уткина; Сост.: А.Н. Кортаев. Рязань, 2020. 16 с.

Излагаются методические рекомендации по выполнению лабораторных работ и практических занятий по дисциплине «Введение в промышленную разработку ПО на платформе Java», и «Промышленная разработка ПО на платформе Java».

Предназначены бакалаврам всех форм обучения направлений 09.03.03 «Прикладная информатика» и 09.03.04 «Программная инженерия».

Библиогр.: 3 назв.

Разработка программного обеспечения, объектно-ориентированное программирование, Java

Рецензент: кафедра вычислительной и прикладной математики Рязанского государственного радиотехнического университета (зав. кафедрой д-р техн. наук, проф. Г.В. Овечкин)

Введение в промышленную разработку
ПО на платформе Java

Составитель К о р т а е в Александр Николаевич

Подписано в печать 11.06.20. Усл. печ. л. 1.

Тираж 1 экз.

Рязанский государственный радиотехнический университет

390005, Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ.

ВВЕДЕНИЕ

Целью освоения дисциплины «Введение в промышленную разработку ПО на платформе Java» является формирование у студентов твердых теоретических знаний в области языка Java, а также получение практических навыков в разработке программного обеспечения.

Лабораторные работы и практические занятия по дисциплине «Введение в промышленную разработку ПО на платформе Java» являются обязательным элементом учебной работы, предусмотренной учебными планами подготовки, реализуемыми в РГРТУ.

Подготовка к практическому занятию состоит в теоретической подготовке (изучение конспекта лекций и дополнительной литературы). На практических занятиях студенты реализовывают задание. При подготовке к практическому занятию необходимо ознакомиться с заданием по изучаемой теме.

Подготовка к лабораторной работе состоит в следующем. Перед началом проведения лабораторной работы необходимо ознакомиться с методическими указаниями к данной лабораторной работе, внимательно ознакомиться с заданием. Желательно заранее выполнить задание в используемой инструментальной среде, чтобы время лабораторного занятия использовать для исправления ошибок, модификации и защиты данной работы. После выполнения лабораторной работы необходимо согласовать полученные результаты с преподавателем. По результатам выполнения лабораторных работ студентам необходимо составить отчет, отражающий основные полученные результаты.

Лабораторная работа по теме № 1. Основы Java

Цель работы: ознакомление с основными конструкциями языка Java, операциями, литералами, различными типами операторов.

Задание

1. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести четные и нечетные числа.
2. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести числа, которые делятся на 3 или на 9.

3. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести числа, которые делятся на 5 или на 10.
4. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести наибольший общий делитель и наименьшее общее кратное этих чисел.
5. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести простые числа.
6. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести "счастливые" числа.
7. Для произвольной цифры от 0 до 9 вывести на консоль ее значение прописью. Например, для цифры 9 на консоли должна быть напечатана строка «Девять».
8. Создайте приложение, которое покажет, что для выражения $an+bn=cn$ (теорема Ферма) нет натуральных решений от 1 до 100 и $n>2$. Убедитесь, что есть решения для $n=2$, и выведите их в консоль.
9. Вычислить выражение $1 - 1/2 + 1/3 - 1/4 + +1/ 9999 - 1/10000$, используя оператор условия.
10. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести числа Фибоначчи: $f_0 = f_1 = 1, f(n) = f(n-1) + f(n-2)$.
11. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести числа-палиндромы, значения которых в прямом и обратном порядке совпадают.
12. Ввести с консоли n целых чисел и поместить их в массив. На консоль вывести период десятичной дроби $p = m/n$ для первых двух целых положительных чисел n и m , расположенных подряд.
13. Ввести с консоли n целых чисел и поместить их в массив. Построить треугольник Паскаля для первого положительного числа.
14. Создайте приложение, которое осуществит перевод чисел из десятичной системы счисления в двоичную и шестнадцатеричную.
15. Написать приложение, выводящее n строк с переходом и без перехода на новую строку.
16. Написать приложение для ввода пароля из командной строки и сравнения его со строкой-образцом.
17. Написать программу ввода целых чисел как аргументов командной строки, подсчета их суммы (произведения) и вывода результата на консоль.
18. Написать приложение, выводящее фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания. Для получения последней даты и времени использовать класс *Date* из пакета *Java.util* (Объявление объекта *Date d = new Date();*) или статический метод класса *System.currentTimeMillis()*.

Лабораторная работа по теме № 2 Абстрактные классы и интерфейсы

Цель работы: реализовать иерархию классов, а также наследование и полиморфизм.

Задание

Создайте программу на языке программирования Java. В программе должны присутствовать: иерархия классов с указанием ключевых полей, свойств и методов для предметной области, указанной в соответствующем варианте. Имена классов придумываются самостоятельно, исходя из названия предметной области. Иерархия классов должна быть такой, как на рис. 1.

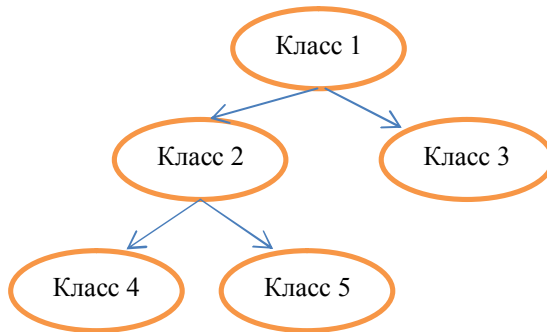


Рис. 1. Иерархия классов.

1. В каждом классе должно быть как минимум 3 поля.
2. Изменять значения поля можно только через методы.
3. Реализовать, как минимум 2 метода, в которых бы использовался полиморфизм.
4. В программе должна быть предусмотрена возможность создания экземпляров всех классов (кроме абстрактных) и просмотра соответствующих полей и результатов выполнения методов.

Варианты предметной области.

1. Магазин электротоваров.
2. Кинотеатр.
3. Поликлиника.
4. Автосервис.
5. Завод металлических изделий.

6. Библиотека.
7. Лекарственные травы.
8. Аптека.
9. Увлечения.
10. Животные.
11. Природные ресурсы.
12. Магазин сантехники.
13. Служба такси.
14. Магазин стройматериалов.
15. Агентство недвижимости.
16. Учебное учреждение (ВУЗ).
17. Учебное учреждение (школа).
18. Супермаркет.
19. Фотостудия.
20. Больница.
21. Салон продажи автомобилей.
22. Кадровое агентство.
23. Книжный магазин.
24. Спортивная секция (футбол, танцы или др.).
25. Ресторан.
26. Железнодорожные перевозки.
27. Магазин молочной продукции.
28. Рыбалка.
29. Магазин кондитерских изделий.

Лабораторная работа по теме № 3 **Интерфейсы**

Цель работы: научиться использовать интерфейсы, а также применять модель исключений.

Задание

Доработайте программу из лабораторной работы №2. Создайте интерфейс и используйте его в одном или нескольких классах своего приложения. Реализуйте для методов, присваивающих значения переменным (например, сеттерам), модель исключений.

Лабораторная работа по теме № 4 Многопоточное программирование

Цель работы: изучение основ многопоточного программирования, знакомство с механизмами регулирования и приостановки потоков.

Задание

Доработайте программу из лабораторной работы №3. Создайте 2 потока: один для создания товаров или ресурсов, а другой для их продажи или расхода. Необходимо отрегулировать работу потоков так, чтобы количество потребляемых ресурсов или продаваемых товаров не превышало количество создаваемых. Для этих целей используйте механизм приостановки потоков.

Лабораторная работа по теме № 5 Аннотации

Цель работы: изучение аннотаций и механизма их работы.

Задание

Доработайте программу из лабораторной работы №4. Для своей предметной области реализуйте аннотацию `@TestCase`, сформировав исходный файл, имя которого состоит из имени класса, где эта аннотация встречается, а также из имени `Test`.

Так, если исходный файл `MyMath.java` содержит следующие строки:

```
@TestCase(params="4", expected="24")
```

```
@TestCase(params="0", expected="1")
```

```
public static long factorial(int n) { ... }
```

то сформируйте исходный файл `MyMathTest.java` со следующими операторами:

```
assert(MyMath.factorial(4) == 24); assert(MyMath.factorial(0) == 1);
```

Лабораторная работа по теме № 6 Организация ввода-вывода

Цель работы: изучение основ ввода-вывода информации при разработке программного обеспечения.

Задание

Доработайте программу из лабораторной работы №5. Для своей предметной области реализуйте ввод исходных данных для своей программы (имя экземпляра класса, значения его полей и т.д.) из файла, например, in.txt. Результаты работы программы, т.е. получившиеся экземпляры классов с их параметрами необходимо сохранить в другой файл, например, out.txt.

Лабораторная работа по теме № 7

Обобщения

Цель работы: изучение основ обобщенного программирования, знакомство со спецификой обобщений.

Задание

На основе предыдущих лабораторных работ для своей предметной области на базе обычного массива (готовые массивы типа ArrayList не использовать) создать обобщённый класс DynamicArray для работы с однотипными полями экземпляров класса, для которых в лабораторной работе №6 создавался ввод и вывод в файл.

Необходимо у всех экземпляров класса считать в DynamicArray значения одного поля (например, ID), вывести содержимое на экран, провести поиск дубликатов при наличии и вывести их количество.

Те же самые операции необходимо произвести поочерёдно со всеми остальными полями класса.

Добавлять в массив данные можно из файлов.

Лабораторная работа по теме № 8

Шаблон проектирования «Фабрика»

Цель работы: изучение реализации шаблона «Фабрика».

Задание

Реализовать шаблон «Фабрика», взяв за основу свой проект, созданный в предыдущих лабораторных работах.

Пояснение

Шаблон «Фабрика» используется, когда у нас есть суперкласс с несколькими подклассами и на основе ввода, нам нужно вернуть один из подкласса. Класс не знает, какого типа объект он должен создать. Объекты создаются в зависимости от входных данных.

Пример реализации:


```

class Factory {
    public OS getCurrentOS(String inputos) {
        OS os = null;
        if (inputos.equals("windows")) {
            os = new windowsOS();
        } else if (inputos.equals("linux")) {
            os = new linuxOS();
        } else if (inputos.equals("mac")) {
            os = new macOS();
        }
        return os;
    }
}
interface OS {
    void getOS();
}
class windowsOS implements OS {
    public void getOS () {
        System.out.println("применить для виндовс");
    }
}
class linuxOS implements OS {
    public void getOS () {
        System.out.println("применить для линукс");
    }
}
class macOS implements OS {
    public void getOS () {
        System.out.println("применить для мак");
    }
}

public class FactoryTest { //тест
    public static void main(String[] args){
        String win = "linux";
        Factory factory = new Factory();
        OS os = factory.getCurrentOS(win);
        os.getOS();
    }
}

```

Лабораторная работа по теме № 9 Лямбда-выражения

Цель работы: изучение лямбда-выражений и способов их использования.

Задание

На основе предыдущих лабораторных работ для своей предметной области составить лямбда-выражение, которое будет считать количество элементов массива, равных/больших/меньших

заданного значения. Например, вывести количество покупателей, купивших товар на сумму, больше 1 000 руб.

Лабораторная работа по теме № 10

Шаблон проектирования «Абстрактная фабрика»

Цель работы: изучение реализации шаблона «Абстрактная фабрика».

Задание

Реализовать шаблон «Абстрактная фабрика», взяв за основу свой проект, созданный в предыдущих лабораторных работах.

Пояснение

Шаблон «Абстрактная фабрика» позволяет выбрать конкретную реализацию фабрики из семейства возможных фабрик. Создает семейство связанных объектов. Легко расширяется.

Пример реализации

```
interface Lada {
    long getLadaPrice();
}
interface Ferrari {
    long getFerrariPrice();
}
interface Porshe {
    long getPorshePrice();
}
interface InteAbsFactory {
    Lada getLada();
    Ferrari getFerrari();
    Porshe getPorshe();
}
class UaLadaImpl implements Lada { // первая
    public long getLadaPrice() {
        return 1000;
    }
}
class UaFerrariImpl implements Ferrari {
    public long getFerrariPrice() {
        return 3000;
    }
}
class UaPorsheImpl implements Porshe {
    public long getPorshePrice() {
        return 2000;
    }
}
class UaCarPriceAbsFactory implements InteAbsFactory {
    public Lada getLada() {
        return new UaLadaImpl();
    }
}
```

```

    }
    public Ferrari getFerrari() {
        return new UaFerrariImpl();
    }
    public Porshe getPorshe() {
        return new UaPorsheImpl();
    }
} // первая
class RuLadaImpl implements Lada { // вторая
    public long getLadaPrice() {
        return 10000;
    }
}
class RuFerrariImpl implements Ferrari {
    public long getFerrariPrice() {
        return 30000;
    }
}
class RuPorsheImpl implements Porshe {
    public long getPorshePrice() {
        return 20000;
    }
}
class RuCarPriceAbsFactory implements InteAbsFactory {
    public Lada getLada() {
        return new RuLadaImpl();
    }
    public Ferrari getFerrari() {
        return new RuFerrariImpl();
    }
    public Porshe getPorshe() {
        return new RuPorsheImpl();
    }
} // вторая

public class AbstractFactoryTest { //тест
    public static void main(String[] args) {
        String country = "UA";
        InteAbsFactory ifactory = null;
        if(country.equals("UA")) {
            ifactory = new UaCarPriceAbsFactory();
        } else if(country.equals("RU")) {
            ifactory = new RuCarPriceAbsFactory();
        }

        Lada lada = ifactory.getLada();
        System.out.println(lada.getLadaPrice());
    }
}

```

Лабораторная работа по теме № 11

Модули

Цель работы: изучение основ разработки модульных приложений.

Задание

На основе предыдущих лабораторных работ для своей предметной области написать службу на основе модулей, которая будет сортировать данные в выходном файле. Алгоритм сортировки выберите самостоятельно, исходя из данных, содержащихся в файле.

Лабораторная работа по теме № 12 Шаблон проектирования «Фасад»

Цель работы: изучение реализации шаблона «Фасад».

Задание

Реализовать шаблон «Фасад», взяв за основу свой проект, созданный в предыдущих лабораторных работах.

Пояснение

Шаблон «Фасад» скрывает сложную систему классов, приводя все вызовы к одному объекту. Помещает вызов нескольких сложных объектов в один объект.

Пример реализации

```
interface Car {
    void start();
    void stop();
}

class Key implements Car {
    public void start() {
        System.out.println("Вставить ключи");
    }
    public void stop() {
        System.out.println("Вытянуть ключи");
    }
}

class Engine implements Car {
    public void start() {
        System.out.println("Запустить двигатель");
    }
    public void stop() {
        System.out.println("Остановить двигатель");
    }
}

class Facade {
    private Key key;
    private Engine engine;
    public Facade() {
        key = new Key();
        engine = new Engine();
    }
}
```

```

    }
    public void startCar() {
        key.start();
        engine.start();
    }
    public void stopCar() {
        key.stop();
        engine.stop();
    }
}

public class FacadeTest { //тест
    public static void main(String[] args) {
        Facade facade = new Facade();
        facade.startCar();
        System.out.println();
        facade.stopCar();
    }
}

```

Лабораторная работа по теме № 13

Основы Spring

Цель работы: изучение основ промышленной разработки ПО с помощью фреймворка Spring.

Задание

На основе своей предметной области из предыдущих лабораторных работ разработайте приложение с помощью фреймворка Spring. В приложении должны присутствовать: контроллер, домашняя страница и тестовый класс для домашней страницы.

Лабораторная работа по теме № 14

Шаблон проектирования «Наблюдатель»

Цель работы: изучение реализации шаблона проектирования «Наблюдатель».

Задание

Реализовать шаблон «Наблюдатель», взяв за основу свой проект, созданный в предыдущих лабораторных работах.

Пояснение

Позволяет одним объектам наблюдать за действиями, происходящими в других объектах.

Пример реализации

```
import java.util.ArrayList;
```

```

import java.util.List;
interface Observer {
    void event(List<String> strings);
}
class University {
    private List<Observer> observers = new
ArrayList<Observer>();
    private List<String> students = new ArrayList<String>();
    public void addStudent(String name) {
        students.add(name);
        notifyObservers();
    }
    public void removeStudent(String name) {
        students.remove(name);
        notifyObservers();
    }
    public void addObserver(Observer observer){
        observers.add(observer);
    }
    public void removeObserver(Observer observer) {
        observers.remove(observer);
    }
    public void notifyObservers(){
        for (Observer observer : observers) {
            observer.event(students);
        }
    }
}
class Director implements Observer {
    public void event(List<String> strings) {
        System.out.println("The list of students has changed: "
+ strings);
    }
}

public class ObserverTest { //тест
    public static void main(String[] args) {
        University university = new University();
        Director director = new Director();
        university.addStudent("Vaska");
        university.addObserver(director);
        university.addStudent("Anna");
        university.removeStudent("Vaska");
    }
}

```

Практическое занятие по теме № 1

Основы Git

1. Установите на свой ПК систему контроля версий Git.

2. Добавьте для отслеживания в неё лабораторную работу №7. Храните её в ветке с именем master.
3. Создайте ветку develop и выполните в ней лабораторную работу №8.
4. Произведите коммит изменений.
5. С помощью команды merge произведите слияние ветки develop с master.
6. Для проверки правильности выполнения упражнения посмотрите логи и постройте ASCII граф.

Практическое занятие по теме № 2

Ветвление в Git

1. Создайте ветку lab10 и выполните в ней лабораторную работу №10.
2. Поставьте аннотированную метку с версией проекта.
3. Произведите коммит изменений.
4. Переключитесь на ветку master и внесите в неё изменение (например, добавьте строку вывода сообщения) с коммитом.
5. Переключитесь на ветку lab10 и убедитесь в том, что изменения в ветку не произошло.
6. Для проверки правильности выполнения упражнения посмотрите логи и постройте ASCII граф.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Программирование на языке Java [Электронный ресурс]: конспект лекций/ А.В. Гаврилов [и др.].— Электрон. текстовые данные.— СПб.: Университет ИТМО, 2015.— 123 с.— Режим доступа: <http://www.iprbookshop.ru/68692.html>.— ЭБС «IPRbooks».
1. Сеттер Р.В. Изучаем Java на примерах и задачах [Электронный ресурс]/ Сеттер Р.В.— Электрон. текстовые данные.— Санкт-Петербург: Наука и Техника, 2016.— 240 с.— Режим доступа: <http://www.iprbookshop.ru/44025.html>.— ЭБС «IPRbooks»
2. Ермаков А.В. Технологии обработки информации на Java [Электронный ресурс]: учебное пособие/ Ермаков А.В.— Электрон. текстовые данные.— Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2015.— 47 с.— Режим доступа: <http://www.iprbookshop.ru/76522.html>.— ЭБС «IPRbooks».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Лабораторная работа по теме № 1	3
Лабораторная работа по теме № 2	5
Лабораторная работа по теме № 3	6
Лабораторная работа по теме № 4	7
Лабораторная работа по теме № 5	7
Лабораторная работа по теме № 6	7
Лабораторная работа по теме № 7	8
Лабораторная работа по теме № 8	8
Лабораторная работа по теме № 9	9
Лабораторная работа по теме № 10	10
Лабораторная работа по теме № 11	11
Лабораторная работа по теме № 12	12
Лабораторная работа по теме № 13	13
Лабораторная работа по теме № 14	13
Практическое занятие по теме № 1	14
Практическое занятие по теме № 2	15
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	15