Федеральное агентство связи ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА» (СПбГУТ)

Факультет инфокоммуникационных сетей и систем Кафедра программной инженерии и вычислительной техники

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине «Разработка Java-приложений управления телекоммуникациями»

10 вариант

Выполнил: студент 3-го курса дневного отделения группы ИКПИ-05

Кузнецов Максим Сергеевич

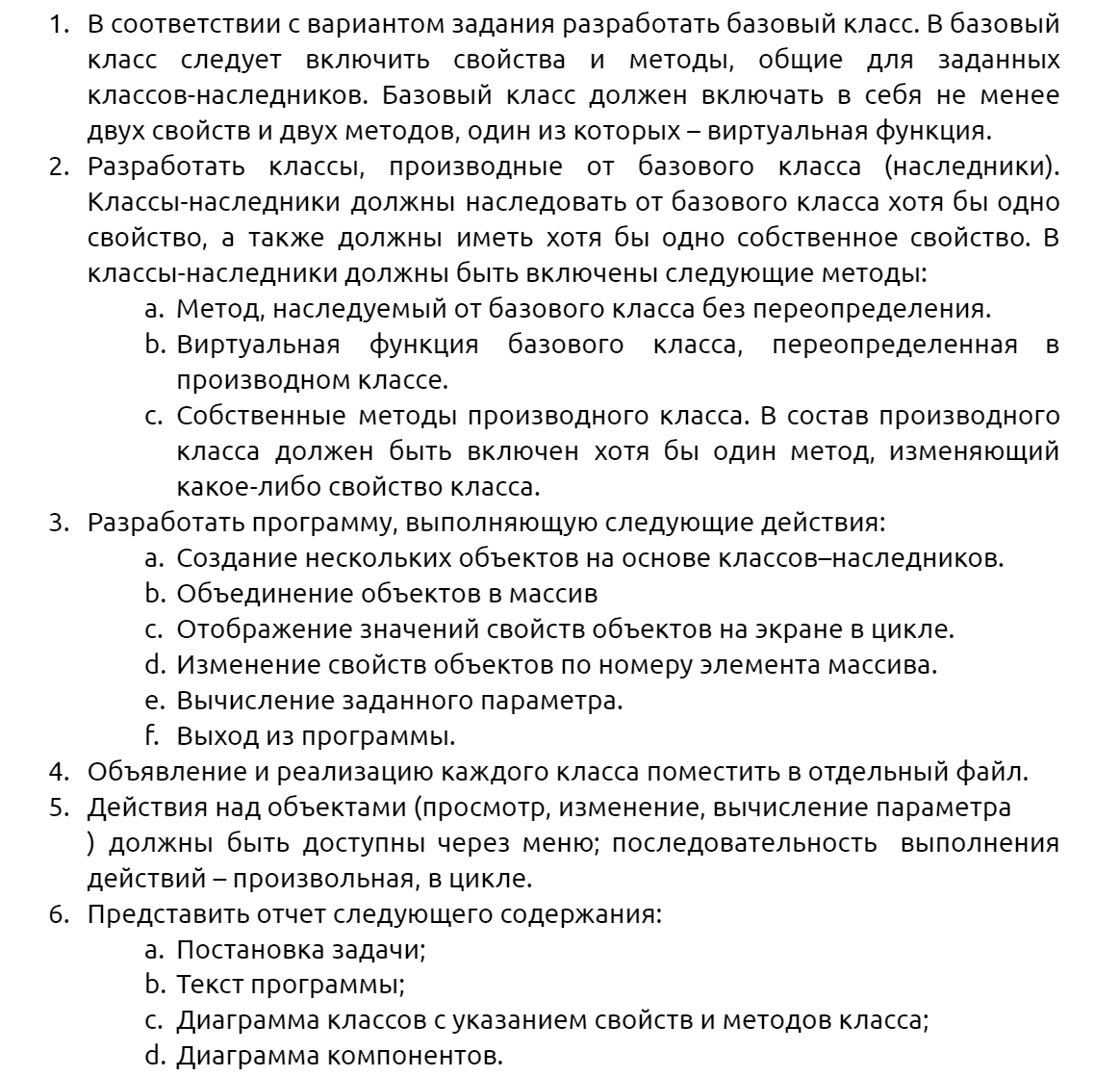
Преподаватель: Березин А. Ю.

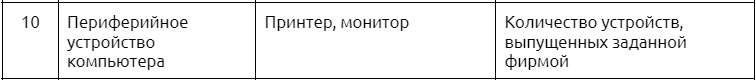
Санкт-Петербург 2022

**Цель работы**

Ознакомиться с принципами ООП на Java.

**Задание**



****

**Реализованные классы по заданию (базовый и наследники)**

|  |  |  |
| --- | --- | --- |
| **Имя класса** | **Свойства** | **Методы** |
| Peripheral  (базовый) | boolean turn  String firma | Peripheral()  [конструктор]  On()  Off()  get\_firma()  get\_OnOff()  abstract void power\_on\_action()  abstract void set(int n, String m)  abstract void get\_unique() |
| Monitor  (наследник) | int hertz | power\_on\_action()  set(int hertz, String m)  get\_unique()  Monitor() |
| Keyboard  (наследник) | String layout | power\_on\_action()  set(int n, String layout)  get\_unique()  Keyboard() |

Peripheral ---> Monitor

|

Keyboard laba2

**Исходный код**

Class: laba2

public static void new\_keyboard(List<Peripheral> masiv){  
 Peripheral klava = new Keyboard();  
 masiv.add(klava);  
}  
public static void new\_monitor(List<Peripheral> masiv){  
 Peripheral monik = new Monitor();  
 masiv.add(monik);  
}  
  
public static void new\_obj(List<Peripheral> masiv){  
 System.out.println("Choose class: 0-monitor, 1-keyboard");  
 while(true) {  
 Scanner input = new Scanner(System.in);  
 int k = 2;  
 try {  
 k = input.nextInt();  
 } catch (Exception e) {  
 System.out.println("Написано же ввести 0 или 1");  
 continue;  
 }  
 switch (k) {  
 case 0 -> new\_monitor(masiv);  
 case 1 -> new\_keyboard(masiv);  
 default -> {  
 System.out.println("Написано же ввести 0 или 1");  
 continue;  
 }  
 }  
 break;  
 }  
 //for(Peripheral i:masiv){out\_obj(i);}//Вывод всех obj для проверки  
}  
  
public static boolean out\_masiv(List<Peripheral> masiv){  
 if (masiv.size()!=0){  
 System.out.println("Вывод устройств");  
 for (int i=0; i<masiv.size(); i++){  
 System.out.println(i + ": " + masiv.get(i));  
 }  
 return true;  
 }else{  
 System.out.println("Нет доступных устройств");  
 return false;  
 }  
}  
public static void choose\_obj(List<Peripheral> masiv){  
 boolean m = out\_masiv(masiv);  
 if(!m){  
 System.out.println("Сначала нужно создать объект");  
 new\_obj(masiv);  
 out\_masiv(masiv);  
 }  
 int k = -1;  
 System.out.println("Введите номер объекта");  
 while(true) {  
 Scanner input = new Scanner(System.in);  
 k = -1;  
 try {  
 k = input.nextInt();  
 } catch (Exception e) {  
 System.out.println("Это не номер существующего объекта");  
 continue;  
 }  
 if(0<=k & k<=masiv.size()){  
 break;  
 }else{  
 System.out.println("Это не номер существующего объекта");  
 }  
 }  
 change\_obj(masiv.get(k));  
}  
  
private static void change\_obj(Peripheral obj) {  
 out\_obj(obj);  
  
 System.out.println("Какое свойство вы хотите поменять, введите его порядковый номер от 0 до 2. Введите 3, чтобы оставить без изменений");  
 while(true) {  
 Scanner input = new Scanner(System.in);  
 int k = 2;  
 try {  
 k = input.nextInt();  
 } catch (Exception e) {  
 System.out.println("Написано же ввести номер свойства от 0 до 2");  
 continue;  
 }  
 switch (k) {  
 case 0:  
 obj.power\_on\_action();  
 break;  
 case 1:  
 System.out.println("Какая фирма у этого устройства?");  
 obj.firma = input.next();  
 break;  
 case 2:  
 if(obj.getClass().getSimpleName().equals("Monitor")){  
 while (true){  
 System.out.println("Какая частота в герцах у этого устройства?");  
 Scanner input2 = new Scanner(System.in);  
 k = -1;  
 try {  
 k = input2.nextInt();  
 } catch (Exception e) {  
 System.out.println("Цифрами надо!!!!!");  
 continue;  
 }  
 obj.set(k, "");  
 break;  
 }  
 }else{  
 System.out.println("Какая раскладка у этого устройства?");  
 obj.set(0, input.next());  
 }  
 break;  
 case 3:  
 break;  
 default:  
 System.out.println("Написано же ввести 0 или 1");  
 continue;  
 }  
 out\_obj(obj);  
 System.out.println("Хотите еще что-нибудь поменять? Введите 1. Иначе любой символ.");  
 Scanner input1 = new Scanner(System.in);  
 if (input1.next().equals("1")){  
 System.out.println("Какое свойство вы хотите поменять, введите его номер. Введите 3, чтобы ничего не изменять");  
 out\_obj(obj);  
 }else{  
 break;  
 }  
 }  
  
}  
static void out\_obj(Peripheral obj){  
 System.out.println(obj.getClass().getSimpleName());  
 obj.get\_OnOff();  
 obj.get\_firma();  
 obj.get\_unique();  
}  
  
static void count\_company\_children(List<Peripheral> masiv){  
 Map<String,Integer> dictionary = new HashMap<String,Integer>();  
 for (Peripheral i: masiv){  
 if (dictionary.containsKey(i.firma)){  
 dictionary.put(i.firma, dictionary.get(i.firma)+1);  
 }else{  
 dictionary.put(i.firma, 1);  
 }  
 }  
 for (String i: dictionary.keySet()) {  
 System.out.println(i+": "+dictionary.get(i));  
 }  
}  
  
public static void main(String[] args) {  
 /\*Peripheral monik = new Monitor();  
 Peripheral klava = new Keyboard();  
 masiv.add(monik);  
 masiv.add(klava);\*/  
 List masiv = new ArrayList<>();  
 System.out.println("Добро пожаловать в меню. Вы можете по разному взаимодействовать с объектами класса.");  
 while(true) {  
 System.out.println("Введите нужный символ, чтобы начать.");  
 System.out.println("0-Просмотреть доступные устройства");  
 System.out.println("1-Создать устройство");  
 System.out.println("2-Изменить одно из устройств");  
 System.out.println("3-Вывести колличество устройств от разных компаний");  
 System.out.println("q-Выход из программы");  
 Scanner input = new Scanner(System.in);  
 switch (input.next()) {  
 case "0":  
 out\_masiv(masiv);  
 break;  
 case "1":  
 new\_obj(masiv);  
 break;  
 case "2":  
 choose\_obj(masiv);  
 break;  
 case "3":  
 count\_company\_children(masiv);  
 break;  
 case "q":  
 return;  
 default:  
 System.out.println("Я не знаю такую команду");  
 }  
 }  
}

Class: Peripheral

protected boolean turn = false;  
protected String firma = "Canon";  
  
protected Peripheral() {  
 System.out.println("Отработал конструктор Peripheral");  
}  
protected void On(){  
 System.out.println("Включили");  
 this.turn = true;  
}  
protected void Off(){  
 System.out.println("Выключили");  
 this.turn = false;  
}  
protected void get\_firma(){  
 System.out.println("Производитель: "+ firma);  
}  
  
protected void get\_OnOff(){  
 if(turn){  
 System.out.println("Состояние: Воркает");  
 }else{  
 System.out.println("Состояние: Не воркакет");  
 }  
}  
  
protected abstract void power\_on\_action();  
  
protected abstract void set(int n, String m);  
  
public abstract void get\_unique();

Class: Monitor

private int hertz = 60;  
public void power\_on\_action(){  
 if(!turn){  
 System.out.println("Свечусь");  
 this.turn = true;  
 }else{  
 System.out.println("Power-off");  
 this.turn = false;  
 }  
}  
  
public void set(int hertz, String m){  
 this.hertz=hertz;  
}  
  
public void get\_unique(){  
 System.out.println("Частота обновления экрана: "+hertz);  
}  
public Monitor() {  
 System.out.println("Отработал конструктор Monitor!");  
}

Class: Keyboard

private String layout = "qwerty";  
  
public void power\_on\_action(){  
 if(!turn){  
 System.out.println("Тык тык тык");  
 this.turn = true;  
 }else{  
 System.out.println("Power-off");  
 this.turn = false;  
 }  
}  
  
public void set(int n, String layout){  
 this.layout=layout;  
}  
  
public void get\_unique(){  
 System.out.println("Раскладка клавиатуры: "+layout);  
}  
public Keyboard() {  
 System.out.println("Отработал конструктор Keyboard!");  
 this.firma = "Sony";  
}

**Заключение**

В результате выполнения лабораторной работы мы ознакомились с основами ООП.