|  |  |
| --- | --- |
|  | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет  имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ: Информатика и системы управления

КАФЕДРА: Компьютерные системы и сети

НАПРАВЛЕНИЕ ПОДГОТОВКИ: 09.04.01 Информатика и вычислительная техника

**О т ч е т**

**по лабораторной работе №4**

**Дисциплина: Языки программирование при работе с большими данными**

**Название: Интерейсы**

студент группы ИУ6-21М  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Коваленко М.В.**

(Подпись, дата) (Фамилия И.О.)

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Степанов П.В.**

(Подпись, дата) (Фамилия И.О.)

Москва, 2020

**Задание 1**

Создать класс City (город) с внутренним классом, с помощью объектов которого можно хранить информацию о проспектах, улицах, площадях.

**Программа**

|  |
| --- |
| import java.util.Scanner;  public class City {  public static Scanner ob = new Scanner(System.in);  public String city;  public info info;  public City() {  System.out.println("Введите название города");  setCity (ob.next());  info = new info();  }  public String getCity() {  return city;  }  public void setCity(String city) {  this.city=city;  }  public class info {  int pr,st,sq;  public info() {  System.out.println("Количество проспектов в городе:");  setPr(ob.nextInt());  System.out.println("Количество улиц в городе:");  setSt(ob.nextInt());  System.out.println("Количество площадей в городе:");  setSq(ob.nextInt());  }  public int getPr() {  return pr;  }  public void setPr(int pr) {  this.pr=pr;  }  public int getSt() {  return st;  }  public void setSt(int st) {  this.st=st;  }  public int getSq() {  return sq;  }  public void setSq(int sq) {  this.sq=sq;  }  public void print() {  System.out.print("В городе "+getCity()+" "+getPr()+" проспектов,"+getSt()+" улиц и "+getSq()+" площадей");  }  }  public static void main(String[] args) {  City ct = new City();  ct.info.print();  }  } |

**Задание 2**

Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов.

interface Mobile <- abstract class Siemens Mobile <- class Model.

**Программа**

|  |
| --- |
| public interface Mobile {  void startCall (String number);  String searchForCall ();  void acceptCall (String number);  void breakCall ();  public static void main(String[] args){  SiemensMobile mob = new Model();  mob.connectToSiemensOnline();  mob.startCall("89999999999");  System.out.println(mob.searchForCall());  mob.acceptCall("8111111119");  mob.breakCall();  }  }  abstract class SiemensMobile implements Mobile {  Boolean incomingCall=true;  Boolean isTalkingNow;  int siemensSerialNumber;  abstract void connectToSiemensOnline();  }  class Model extends SiemensMobile {  @Override  void connectToSiemensOnline() {  System.out.println("Подсоединились к сервису Сименс Онлайн");  }  @Override  public void startCall(String number) {  isTalkingNow = true;  }  @Override  public String searchForCall() {  while (true){  if (incomingCall){  if (isTalkingNow){  return "Линяя занята";  } else return "Вызов принят";  }  }  }  @Override  public void acceptCall(String number) {  isTalkingNow = true;  }  @Override  public void breakCall() {  isTalkingNow = false;  System.out.println("Вызов завершен");  }  } |

**Вывод**

В результате выполнения данной лабораторной работы разобралась с внутренними классами и интерфейсами.