|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования 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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ и системы  
 управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,**

**обработки и интерпретации больших данных**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 4 |

**Название:**

Внутренние классы. Интерфейсы

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | А.А. Морозова |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

**Цель работы** — ознакомление с внутренними классами и принципом интерфейсов в языке Java.

1. Создать класс Mobile с внутренним классом, с помощью объектов которого можно хранить информацию о моделях телефонов и их свойствах.

Часть кода задания приведена в листинге 1, результат выполнения – на рисунке 1.

Листинг 1 – Внутренний класс свойств телефона и его конструктор

|  |
| --- |
| public class MobileProperties {  private String color;  private String OS;  private int RAM;  private int ROM;  public MobileProperties(String color, String OS, int RAM, int ROM) {  this.color = color;  this.OS = OS;  this.RAM = RAM;  this.ROM = ROM;  }  } |

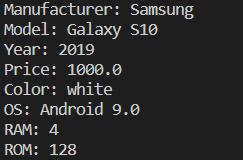


Рисунок 1 – Пример вывода свойств телефона

1. Создать класс Художественная Выставка с внутренним классом, с помощью объектов которого можно хранить информацию о картинах, авторах и времени проведения выставок.

Часть кода задания приведена в листинге 2, результат выполнения – на рисунке 2.

Листинг 2 – Внутренний класс события выставки

|  |
| --- |
| public ExhibitionEvent(String[] artists, String[] paintings, String date) {  this.artists = artists;  this.paintings = paintings;  this.date = date;  }  public String[] getArtists() {  return artists;  }  public String[] getPaintings() {  return paintings;  }  public String getDate() {  return date;  }  } |

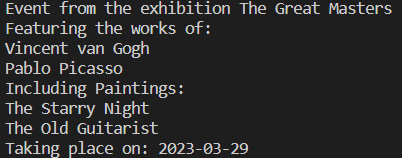


Рисунок 2 – Пример работы класса Выставка

1. Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов:
   1. interface Сотрудник <- class Инженер <- class Руководитель.

Часть кода задания приведена в листинге 3, результат выполнения – на рисунке 3.

Листинг 3 – Класс Инженер

|  |
| --- |
| public class Engineer implements Employee {  protected String name;  private double salary;  public Engineer(String name, double salary) {  this.name = name;  this.salary = salary;  }  @Override  public void work() {  System.out.println(this.name + " is working...");  }  @Override  public void getPaid() {  System.out.println(this.name + " is getting paid " + this.salary + ".");  }  } |

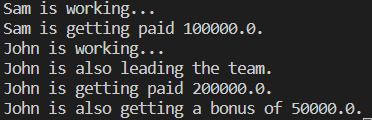


Рисунок 3 – Пример работы классов

* 1. interface Здание <- abstract class Общественное Здание <- class Театр.

Часть кода задания приведена в листинге 4, результат выполнения – на рисунке 4.

Листинг 4 – Класс Театр

|  |
| --- |
| public class Theater extends PublicBuilding {      private String name;      private int numSeats;      public Theater(String address, int numFloors, String name, int numSeats) {          super(address, numFloors);          this.name = name;          this.numSeats = numSeats;      }      @Override      public void printInfo() {          super.printInfo();          System.out.println("A theater named " + this.name + " with " + this.numSeats + " seats...");      }  } |



Рисунок 4 – Пример работы классов

Полный код заданий размещен в репозитории по ссылке – https://github.com/moroz-matros/BDL.

**Вывод** — в ходе работы были получены навыки работы с внутренними классами и интерфейсами.