|  |  |
| --- | --- |
| **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 и системы управления**

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

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

**ОТЧЕТ**

**по лабораторной работе № 4**

|  |  |
| --- | --- |
| **Название:** | Внутренние классы, интерфейсы |
| **Дисциплина:** | Языки программирования для работы с большими данными |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  | 31.03.2023 | И.Б. Нуриддинов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

Цель работы: Изучение и практическое освоение концепций внутренних классов и интерфейсов в языке программирования Java.

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

public class ArtExhibition {

private String name;

private ArrayList<Painting> paintings;

// private ArrayList<String> authors;

private Date date;

public ArtExhibition(String name, Date date) {

this.name = name;

this.date = date;

this.paintings = new ArrayList<>();

// this.authors = new ArrayList<>();

}

public void addPainting(Painting painting) {

paintings.add(painting);

// authors.add(author);

}

public void showExhibitionInfo() {

System.out.println("Name: " + name);

System.out.println("Date: " + date);

System.out.println("Paintings: ");

for (int i = 0; i < paintings.size(); i++) {

System.out.println(paintings.get(i).getTitle() + " by " + paintings.get(i).getAuthor());

}

}

public class Painting {

private String title;

private int year;

private String author;

public Painting(String title, int year, String author) {

this.title = title;

this.year = year;

this.author = author;

}

public String getTitle() {

return title;

}

public int getYear() {

return year;

}

public String getAuthor() {

return author;

}

}

}

*Вариант 2. Задание 1.* Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов

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

public interface Building {

void open();

void close();

}

public abstract class PublicBuilding {

protected String name;

public PublicBuilding(String name) {

this.name = name;

}

public void open() {

System.out.println("Открытие " + name);

}

public void close() {

System.out.println("Закрытие " + name);

}

}

public class Theater extends PublicBuilding{

private int numberOfSeats;

public Theater(String name, int numberOfSeats) {

super(name);

this.numberOfSeats = numberOfSeats;

}

public void showPerformance() {

System.out.println("Показ спектакля в " + name);

}

}

Вывод: Использование внутренних классов и интерфейсов является важным инструментом в разработке программного обеспечения. Внутренние классы позволяют создавать более компактный и модульный код, улучшают читаемость и обеспечивают логическую связь между классами. Интерфейсы, в свою очередь, позволяют определить контракты, которым должны соответствовать классы, и обеспечивают возможность реализации множественного наследования.