|  |  |
| --- | --- |
|  | **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 |  |  | И. Д. Капкин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

1. ****Введение и задание****

**Целью лабораторной работы является изучение внутренних классов и интерфейсов в java.**

**Номер в списке – 10.**

**Вариант 1.**

**Задание 10. Создать класс Cinema (кино) с внутренним классом, с помощью объектов которого можно хранить информацию об адресах кинотеатров, фильмах и времени сеансов.**

**Задание 1.** Создать класс City (город) с внутренним классом, с помощью объектов которого можно хранить информацию о проспектах, улицах, площадях.

**Вариант 2.**

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

**Задание 10.** interface Фильм <- class Отечественный Фильм <- class Комедия.

**Задание 1.** Абстрактный класс Книга (Шифр, Автор, Название, Год, Издательство). Подклассы Справочник и Энциклопедия.

1. ****Ход работы****

**Выполнение задания №10 первого варианта лабораторной работы:**

**import java.time.LocalDateTime;**

**import java.util.ArrayList;**

**import java.util.Arrays;**

**import java.util.List;**

**public class Main {**

**public static void main(String[] args) {**

**Cinema cinema = new Cinema("some address");**

**cinema.addSeance(cinema.new Seance(**

**new Cinema.Film("awdwd", "awdo", 2012, 130),**

**LocalDateTime.of(2023, 03, 31, 23, 55)**

**));**

**cinema.addSeance(cinema.new Seance(**

**new Cinema.Film("awndawd", "awd", 2021, 230),**

**LocalDateTime.of(2023, 04, 01, 10, 30)**

**));**

**System.out.println(Arrays.toString(cinema.getFilms()));**

**}**

**}**

**class Cinema {**

**private String address;**

**private List<Seance> seances;**

**public Cinema (String address) {**

**this.address = address;**

**this.seances = new ArrayList<>();**

**}**

**public void addSeance(Seance seance) {**

**this.seances.add(seance);**

**}**

**public class Seance {**

**private LocalDateTime start;**

**private Film film;**

**public Seance (Film film, LocalDateTime start) {**

**this.film = film;**

**this.start = start;**

**}**

**public Film getFilm() {**

**return film;**

**}**

**}**

**public static class Film {**

**private String name;**

**private String director;**

**private int year;**

**private int duration;**

**public Film (String name, String director, int year, int duration) {**

**this.name = name;**

**this.director = director;**

**this.year = year;**

**this.duration = duration;**

**}**

**@Override**

**public String toString() {**

**return name + "\t" + year + "\t" + director + "\t" + duration;**

**}**

**}**

**public Film[] getFilms() {**

**return seances.stream().map(Seance::getFilm).toArray(Film[]::new);**

**}**

**}**

**Выполнение задания №1 первого варианта лабораторной работы:**

**import java.util.ArrayList;**

**import java.util.Arrays;**

**import java.util.List;**

**public class Main {**

**public static void main(String[] args) {**

**City zel = new City("Zelenograd");**

**zel.addProspect(zel.new GeoObject("Panfilovsky",**

**new City.GPSCoordinates(55.9884, 37.1993),**

**new City.GPSCoordinates(55.9863, 37.2123)));**

**zel.addProspect(zel.new GeoObject("Moskovsky",**

**new City.GPSCoordinates(55.9869, 37.1933),**

**new City.GPSCoordinates(55.9818, 37.2062)));**

**zel.addProspect(zel.new GeoObject("Tsentralny",**

**new City.GPSCoordinates(55.9914, 37.2084),**

**new City.GPSCoordinates(55.9828, 37.2045)));**

**zel.addStreet(zel.new GeoObject("Akademika Valieva",**

**new City.GPSCoordinates(55.9812, 37.1964),**

**new City.GPSCoordinates(55.9854, 37.1918)));**

**zel.addStreet(zel.new GeoObject("Junosti",**

**new City.GPSCoordinates(55.9856, 37.1942),**

**new City.GPSCoordinates(55.9784, 37.1936)));**

**zel.addSquare(zel.new GeoObject("Junosti",**

**new City.GPSCoordinates(55.9827, 37.1915),**

**new City.GPSCoordinates(55.9826, 37.1903),**

**new City.GPSCoordinates(55.9815, 37.1903),**

**new City.GPSCoordinates(55.9816, 37.1915)));**

**System.out.println("Zelenograd prospects:");**

**System.out.println(zel.prospects.toString());**

**System.out.println("Zelenograd streets:");**

**System.out.println(zel.streets.toString());**

**System.out.println("Zelenograd squares:");**

**System.out.println(zel.squares.toString());**

**}**

**}**

**class City {**

**public String name;**

**public List<GeoObject> prospects;**

**public List<GeoObject> streets;**

**public List<GeoObject> squares;**

**public City (String name) {**

**this.name = name;**

**this.prospects = new ArrayList<>();**

**this.streets = new ArrayList<>();**

**this.squares = new ArrayList<>();**

**}**

**public void addProspect (GeoObject prospect) {**

**this.prospects.add(prospect);**

**}**

**public void addStreet (GeoObject street) {**

**this.streets.add(street);**

**}**

**public void addSquare (GeoObject square) {**

**this.squares.add(square);**

**}**

**class GeoObject {**

**public String name;**

**private GPSCoordinates [] coordinates;**

**public GeoObject (String name, GPSCoordinates... coordinates) {**

**this.name = name;**

**this.coordinates = coordinates;**

**}**

**@Override**

**public String toString() {**

**return this.name + " " + Arrays.toString(this.coordinates);**

**}**

**}**

**public static class GPSCoordinates {**

**public double latitude;**

**public double longitude;**

**public GPSCoordinates (double latitude, double longitude) {**

**this.latitude = latitude;**

**this.longitude = longitude;**

**}**

**@Override**

**public String toString() {**

**return "lat " + latitude + " lon " + longitude;**

**}**

**}**

**}**

**Выполнение задания №10 второго варианта лабораторной работы:**

**public class Main {**

**public static void main(String[] args) {**

**Film[] films = new Film[3];**

**films[0] = new DomesticFilm("Солярис", "Андрей Тарковский", "1972");**

**films[1] = new Comedy("Осенний марафон", "Георгий Данелия", "1979", "Георгий Данелия, Олег Басилашвили");**

**films[2] = new DomesticFilm("Иди и смотри", "Элем Климов", "1985");**

**for (Film film : films) {**

**film.display();**

**System.out.println();**

**}**

**}**

**}**

**interface Film {**

**void display();**

**}**

**class DomesticFilm implements Film {**

**private String title;**

**private String director;**

**private String year;**

**public DomesticFilm(String title, String director, String year) {**

**this.title = title;**

**this.director = director;**

**this.year = year;**

**}**

**@Override**

**public void display() {**

**System.out.println("Domestic Film:");**

**System.out.println("Title: " + title);**

**System.out.println("Director: " + director);**

**System.out.println("Year: " + year);**

**}**

**}**

**class Comedy extends DomesticFilm {**

**private String cast;**

**public Comedy(String title, String director, String year, String cast) {**

**super(title, director, year);**

**this.cast = cast;**

**}**

**@Override**

**public void display() {**

**System.out.println("Comedy:");**

**super.display();**

**System.out.println("Cast: " + cast);**

**}**

**}**

**Выполнение задания №1 второго варианта лабораторной работы:**

**public class Main {**

**public static void main(String[] args) {**

**Book [] books = new Book[3];**

**books[0] = new Directory("ISBN 978-1-42-007544-1", "Klaus D. Sattler", "Handbook of Nanophysics", 2010, "CRC Press", 22);**

**books[1] = new Directory("ISBN 978-1-49-870306-2", "Yury Gogotsi", "Nanomaterials Handbook", 2017, "CRC Press", 38);**

**books[2] = new Encyclopedia("ISBN 978-0-08-045046-9", "Larry R. Squire", "Encyclopedia of Neuroscience", 2009, "Academic Press", 6000);**

**for (Book book : books) {**

**book.display();**

**System.out.println();**

**}**

**}**

**}**

**abstract class Book {**

**protected String cipher;**

**protected String author;**

**protected String title;**

**protected int year;**

**protected String publisher;**

**public Book (String cipher, String author, String title, int year, String publisher) {**

**this.cipher = cipher;**

**this.author = author;**

**this.title = title;**

**this.year = year;**

**this.publisher = publisher;**

**}**

**public abstract int rate();**

**public abstract void display();**

**}**

**class Directory extends Book {**

**private int hIndex;**

**public Directory (String cipher, String author, String title, int year, String publisher, int hIndex) {**

**super(cipher, author, title, year, publisher);**

**this.hIndex = hIndex;**

**}**

**public int rate() {**

**return this.hIndex;**

**}**

**public void display() {**

**System.out.println("Directory");**

**System.out.println("cipher: " + this.cipher);**

**System.out.println("editor: " + this.author);**

**System.out.println("title: " + this.title);**

**System.out.println("publisher: " + this.publisher);**

**System.out.println("h-Index: " + this.rate());**

**}**

**}**

**class Encyclopedia extends Book {**

**private int citation;**

**public Encyclopedia (String cipher, String author, String title, int year, String publisher, int citation) {**

**super(cipher, author, title, year, publisher);**

**this.citation = citation;**

**}**

**public int rate() {**

**return this.citation;**

**}**

**public void display() {**

**System.out.println("Encyclopedia");**

**System.out.println("cipher: " + this.cipher);**

**System.out.println("editor-in-chief: " + this.author);**

**System.out.println("title: " + this.title);**

**System.out.println("publisher: " + this.publisher);**

**System.out.println("citation count: " + this.rate());**

**}**

**}**

1. ****Заключение****

**В процессе выполнения лабораторной работы были получены навыки работы с языком программирования java, были изучены внутренние классы и интерфейсы.**