**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 **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

О Т Ч Е Т

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

**Дисциплина:** Внутренние классы. Интерфейсы

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

|  |  |  |  |
| --- | --- | --- | --- |
| Студент | ИУ6-22М |  | Д.М. Карасёв\_\_ |
|  | (Группа) | (Подпись, дата) | (И.О. Фамилия) |
| Преподаватель |  | П.В. Степанов | |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель:** ознакомиться с базовыми принципами языка Java для работы с большими данными

**Вариант 1:**

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

2. Создать класс CD (mp3-диск) с внутренним классом, с помощью объектов которого можно хранить информацию о каталогах, подкаталогах и записях.

**Решение:**

package lr4\_v1\_task1;  
  
import java.util.ArrayList;  
  
public class City {  
 private String NameOfCity;  
 private int population;  
 public City(String nameOfCity, int population){  
 this.NameOfCity = nameOfCity;  
 this.population = population;  
 }  
  
 public int getPopulation() {  
 return population;  
 }  
  
 public String getNameOfCity() {  
 return NameOfCity;  
 }  
 public class StreetsAndSquare{  
 private ArrayList<String> streets;  
 private ArrayList<String> squares;  
 private ArrayList<String> avenue;  
 public StreetsAndSquare(ArrayList<String> squares, ArrayList<String> streets, ArrayList<String> avenue){  
 this.squares = squares;  
 this.streets = streets;  
 this.avenue = avenue;  
 }  
  
 public ArrayList<String> getAvenue() {  
 return avenue;  
 }  
  
 public ArrayList<String> getSquares() {  
 return squares;  
 }  
  
 public ArrayList<String> getStreets() {  
 return streets;  
 }  
 }  
}

package lr4\_v1\_task1;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
  
public class lr4\_v1\_task1 {  
 public static void main(String[] args) {  
 City city = new City("Москва", 12655050);  
 ArrayList<String> streets = new ArrayList<>(Arrays.asList("Тверская", "Новый Арбат", "Пресненская набережная"));  
 ArrayList<String> squares = new ArrayList<>(Arrays.asList("Красная площадь", "Пушкинская площадь", "Театральная площадь"));  
 ArrayList<String> avenues = new ArrayList<>(Arrays.asList("Ленинградский проспект", "Пресненский вал", "Кутузовский проспект"));  
 City.StreetsAndSquare streetsAndSquares = city.new StreetsAndSquare( squares, streets, avenues);  
 ArrayList<String> cityStreets = streetsAndSquares.getStreets();  
 ArrayList<String> citySquares = streetsAndSquares.getSquares();  
 ArrayList<String> cityAvenues = streetsAndSquares.getAvenue();  
 System.out.println("Улицы:");  
 System.out.println(cityStreets);  
 System.out.println("Проспекты:");  
 System.out.println(cityAvenues);  
 System.out.println("Площади:");  
 System.out.println(citySquares);  
 }  
}

На рисунке 1 представлен результат выполнения.

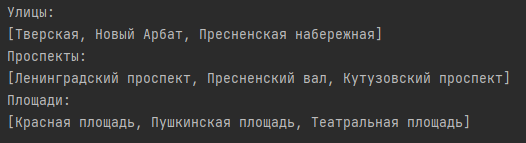
****

Рисунок 1 – Результат выполнения

package lr4\_v1\_task2;  
  
import java.util.ArrayList;  
  
public class CD {  
  
 private String name;  
 private int year;  
  
 // Внутренний класс для хранения информации о каталогах, подкаталогах и записях  
 public class Directory {  
  
 private String name;  
 private ArrayList<String> subdirectories;  
 private ArrayList<String> tracks;  
  
 public Directory(String name) {  
 this.name = name;  
 this.subdirectories = new ArrayList<>();  
 this.tracks = new ArrayList<>();  
 }  
  
 public void addSubdirectory(String subdirectory) {  
 subdirectories.add(subdirectory);  
 }  
  
 public void addTrack(String track) {  
 tracks.add(track);  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public ArrayList<String> getSubdirectories() {  
 return subdirectories;  
 }  
  
 public ArrayList<String> getTracks() {  
 return tracks;  
 }  
 }  
  
 private ArrayList<Directory> directories;  
  
 public CD(String name, int year) {  
 this.name = name;  
 this.year = year;  
 this.directories = new ArrayList<>();  
 }  
  
 public void addDirectory(Directory directory) {  
 directories.add(directory);  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getYear() {  
 return year;  
 }  
  
 public ArrayList<Directory> getDirectories() {  
 return directories;  
 }  
}

package lr4\_v1\_task2;  
  
public class lr4\_v1\_task2 {  
 public static void main(String[] args) {  
 CD cd = new CD("My CD", 2022);  
  
 CD.Directory directory1 = cd.new Directory("Rock");  
 directory1.addTrack("Track 1");  
 directory1.addTrack("Track 2");  
  
 CD.Directory directory2 = cd.new Directory("Pop");  
 directory2.addSubdirectory("Ballads");  
 directory2.addTrack("Track 3");  
  
 cd.addDirectory(directory1);  
 cd.addDirectory(directory2);  
  
 System.out.println("CD name: " + cd.getName() + ", year: " + cd.getYear());  
  
 for (CD.Directory directory : cd.getDirectories()) {  
 System.out.println("Directory name: " + directory.getName());  
  
 System.out.println("Subdirectories:");  
 for (String subdirectory : directory.getSubdirectories()) {  
 System.out.println("- " + subdirectory);  
 }  
  
 System.out.println("Tracks:");  
 for (String track : directory.getTracks()) {  
 System.out.println("- " + track);  
 }  
 }  
 }  
}

На рисунке 2 представлен результат выполнения.

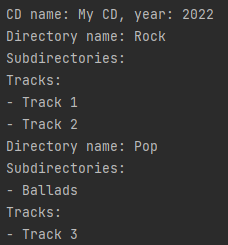


Рисунок 2 – Результат выполнения

**Вариант 2:**

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

1. Абстрактный класс Книга (Шифр, Автор, Название, Год, Издательство). Подклассы Справочник и Энциклопедия.
2. interface Абитуриент <- abstract class Студент <- class Студент-Заочник.

**Решение:**

package lr4\_v2\_task1;  
  
public abstract class Book {  
 protected String code,title,author,publisher;  
 protected int year;  
 public Book(String code, String title, String author, String publisher, int year){  
 this.author = author;  
 this.code = code;  
 this.title = title;  
 this.publisher = publisher;  
 this.year = year;  
 }  
  
 public abstract void print();  
  
}

package lr4\_v2\_task1;  
  
public class Encyclopedia extends Book{  
 private String volume;  
  
 public Encyclopedia(String code, String title, String author, String publisher, int year, String volume){  
 super(code, title, author, publisher, year);  
 this.volume= volume;  
 }  
  
 public String getVolume() {  
 return volume;  
 }  
  
 @Override  
 public void print() {  
 System.*out*.println("Книга: " + title);  
 System.*out*.println("Автор: " + author);  
 System.*out*.println("Издательство: " + publisher);  
 System.*out*.println("Год издания: " + year);  
 System.*out*.println("Шифр: " + code);  
 System.*out*.println("Объём: " + volume);  
 System.*out*.println();  
 }  
}

package lr4\_v2\_task1;  
  
public class Guide extends Book{  
 private String thematic;  
 public Guide(String code, String title, String author, String publisher, int year, String thematic){  
 super(code, title, author, publisher, year);  
 this.thematic = thematic;  
 }  
  
 public String getThematic() {  
 return thematic;  
 }  
  
 @Override  
 public void print() {  
 System.out.println("Книга: " + title);  
 System.out.println("Автор: " + author);  
 System.out.println("Издательство: " + publisher);  
 System.out.println("Год издания: " + year);  
 System.out.println("Шифр: " + code);  
 System.out.println("Тематика: " + thematic);  
 System.out.println();  
 }  
}

package lr4\_v2\_task1;  
  
public class lr4\_v2\_task1 {  
 public static void main(String[] args) {  
 Encyclopedia encyclopedia = new Encyclopedia("ISBN-10: 2-266-11156-6", "Космос", "Галилео Галилей", "Умка",2002,"650 страниц");  
 Guide guide = new Guide("ISBN-10: 2-266-134236-6", "Джунгли", "Автор Авторов", "Умка",2002,"Выживание");  
 encyclopedia.print();  
 guide.print();  
 }  
}

На рисунке 3 представлен результат выполнения.

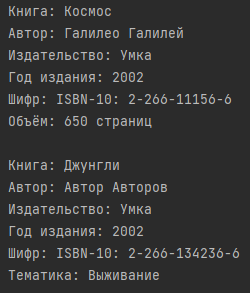


Рисунок 3 – Результат выполнения

package lr4\_v2\_task2;  
  
public interface Applicant {  
 public void apply();  
}

package lr4\_v2\_task2;  
  
public class lr4\_v2\_task2 {  
 public static void main(String[] args) {  
 String[] schedule = {"Высшая математика", "Физика", "Основы программирования"};  
 PartTimeStudent newStudent = new PartTimeStudent("Данила", 18, "МГТУ", schedule, "Информатиа и вычислительная техника");  
 newStudent.apply();  
 newStudent.study();  
 }  
}

package lr4\_v2\_task2;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
  
public class PartTimeStudent extends Student{  
 private String[] schedule;  
 private String specialization;  
 public PartTimeStudent(String name, int age, String university, String[] schedule, String specialization){  
 super(name, age, university);  
 this.schedule = schedule;  
 this.specialization = specialization;  
 }  
  
 public String getSpecialization() {  
 return specialization;  
 }  
  
 public String[] getSchedule() {  
 return schedule;  
 }  
  
 @Override  
 public void apply() {  
 System.out.println("Форма подачи документов студента: " + Name + " - заочная");  
 }  
  
 @Override  
 public void study() {  
 System.out.println("Специальность студента: " + specialization);  
 System.out.println("Расписание студента: " + Arrays.toString(schedule));  
 }  
}

package lr4\_v2\_task2;  
  
public abstract class Student implements Applicant{  
 protected String Name;  
 protected int age;  
 protected String university;  
 public Student(String name, int age, String university){  
 this.Name = name;  
 this.age = age;  
 this.university = university;  
 }  
  
 public abstract void study();  
}

На рисунке 4 представлен результат выполнения.

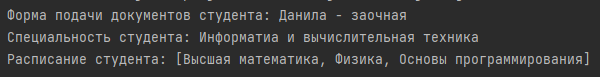


Рисунок 4 – Результат выполнения

**Вывод:** были разработаны методы и классы согласно заданию.