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

**Отчет**

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

**Название:**

Интерфейс

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | Т.А. Малкина |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

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 com.company;  public class Main {   public static void main(String[] args) {  /\*  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.  \*/  City city = new City(0,"Moscow");  System.out.println(city);  CD cd = new CD(0);  System.out.println(cd);  } }  package com.company;  public class Infr {  String prosp, street, square, nameCity;    public Infr(String nameCity) {  this.nameCity = nameCity;  this.prosp = nameCity+", prospect:1,2,3,4,5";  this.street = prosp+", street: number, kv";  this.square = nameCity+", square:1,2";  }   @Override  public String toString() {  return "Infr{" +  "prosp='" + prosp + '\'' +  ", street='" + street + '\'' +  ", square='" + square + '\'' +  '}';  } }  package com.company;  public class Information {  int idCd;  String cat,podcat,zap;  public Information(int idCd) {  this.idCd = idCd;  }   public String getCat() {  return cat;  }   public void setCat(String cat) {  this.cat = cat;  }   public String getPodcat() {  return podcat;  }   public void setPodcat(String podcat) {  this.podcat = podcat;  }   public String getZap() {  return zap;  }   public void setZap(String zap) {  this.zap = zap;  }   @Override  public String toString() {  return "Information{" +  "cat='" + cat + '\'' +  ", podcat='" + podcat + '\'' +  ", zap='" + zap + '\'' +  '}';  } }  package com.company;  public class City {  int idCity;  String nameCity;  Infr infrCity;   public City(int idCity, String nameCity) {  this.idCity = idCity;  this.nameCity = nameCity;  infrCity = new Infr(this.nameCity);  }   @Override  public String toString() {  return "City{" +  "idCity=" + idCity +  ", nameCity='" + nameCity + '\'' +  ", infrCity=" + infrCity +  '}';  } }  package com.company;  public class CD {  Information infa;   int idCd;   public CD(int idCd) {  this.idCd = idCd;  infa = new Information(idCd);  set\_information();  }   void set\_information(){  infa.setCat("C: /fjkdg/jhfjk/");  infa.setPodcat("/test.txt");  infa.setZap("Information about zap");  }   @Override  public String toString() {  return "CD{" +  "idCd=" + idCd +  ", infa=" + infa +  '}';  } } |

Вариант 2:

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

Решение:

|  |
| --- |
| package com.company;  public interface Applicant {  void setSumm\_ex(int summ\_ex);  void setForma(String forma); }  package com.company;  public abstract class Book {  int code,year;  String name,author,house;  public abstract void setCode(int code);  public abstract void setYear(int year);  public abstract void setName(String name);  public abstract void setAuthor(String author);  public abstract void setHouse(String house); }  package com.company;  public class Correspondence\_Student extends Student{    public Correspondence\_Student() {  this.setFio("SName Name !");  this.setForma("Corr");  this.setId(12);  this.setKyrs("Information");  this.setSumm\_ex(227);  this.setYear(2006);  }    @Override  public void setFio(String fio) {  this.fio = fio;  }   @Override  public String toString() {  return "Correspondence\_Student{" +  "fio='" + fio + '\'' +  ", year=" + year +  ", id=" + id +  ", kyrs='" + kyrs + '\'' +  ", forma='" + forma + '\'' +  ", summ\_ex=" + summ\_ex +  '}';  } }  package com.company;  public class Encyclopedia extends Book {   public Encyclopedia() {  this.setCode(2343);  this.setYear(2020);  this.setAuthor("Somebody");  this.setName("Animal");  this.setHouse("23");  }   @Override  public void setCode(int code) {  this.code = code;  }  @Override  public void setYear(int year) {  this.year = year;  }  @Override  public void setName(String name) {  this.name = name;  }  @Override  public void setAuthor(String author) {  this.author = author;  }  @Override  public void setHouse(String house) {  this.house = house;  }   @Override  public String toString() {  return "Encyclopedia{" +  "code=" + code +  ", year=" + year +  ", name='" + name + '\'' +  ", author='" + author + '\'' +  ", house='" + house + '\'' +  '}';  } }  package com.company;  public class Handbook extends Book{    public Handbook() {  this.setCode(1);  this.setYear(2000);  this.setAuthor("Franck Thilliez");  this.setName("Puzzle");  this.setHouse("123");  }    @Override  public void setCode(int code) {  this.code = code;  }   @Override  public void setYear(int year) {  this.year = year;  }   @Override  public void setName(String name) {  this.name = name;  }   @Override  public void setAuthor(String author) {  this.author = author;  }   @Override  public void setHouse(String house) {  this.house = house;  }   @Override  public String toString() {  return "Handbook{" +  "code=" + code +  ", year=" + year +  ", name='" + name + '\'' +  ", author='" + author + '\'' +  ", house='" + house + '\'' +  '}';  } }  package com.company;  public class Main {   public static void main(String[] args) {  /\*  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.  \*/  Book book = new Handbook();  System.out.println(book);  Encyclopedia encyclopedia = new Encyclopedia();  System.out.println(encyclopedia);  System.out.println("-------------------------");  Correspondence\_Student student = new Correspondence\_Student();  System.out.println(student);   } }  package com.company;  public abstract class Student implements Applicant{  String fio,kyrs,forma;  int year,summ\_ex,id;   public abstract void setFio(String fio);   public void setKyrs(String kyrs) {  this.kyrs = kyrs;  }   public void setYear(int year) {  this.year = year;  }  public void setId(int id) {  this.id = id;  }  public void setSumm\_ex(int summ\_ex) {  this.summ\_ex = summ\_ex;  }  public void setForma(String forma) {  this.forma = forma;  } } |

Ссылка на репозиторий:

https://github.com/Carpediem2026/BigData