|  |  |
| --- | --- |
| 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 (ИУ-6)

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**

# Дисциплина: Языки программирования для работы с большими данными

**Название:** Внутренние классы. Интерфейсы

Студент гр. **ИУ6-23М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.П.Завьялова**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.В.Степанов**

(Подпись, дата) (И.О. Фамилия)

Москва, 2022

**Условие:**

Вариант 1

7. Создать класс Cправочная Cлужба Oбщественного Tранспорта с внутренним классом, с помощью объектов которого можно хранить информацию о времени, линиях маршрутов и стоимости проезда.

8. Создать класс Computer (компьютер) с внутренним классом, с помощью объектов которого можно хранить информацию об операционной системе, процессоре и оперативной памяти.

Вариант 2

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

7. interface Врач <- class Хирург <- class Нейрохирург.

8. interface Корабль <- class Грузовой Корабль <- class Танкер.

**Фрагмент решения:**

package lab4.p1;  
  
import java.util.ArrayList;  
  
public class Computer {  
 int num;  
 ArrayList<Computer.Info> info = new ArrayList<Computer.Info>();  
  
 public Computer(int num, int[] proc, String[] os, int[] mem) {  
 this.num = num;  
 for (int i = 0; i < num; i++) {  
 int processor = proc[i];  
 String operation = os[i];  
 int memory = mem[i];  
 info.add(new Computer.Info(processor, operation, memory));  
 }  
 }  
  
 @Override  
 public String toString() {  
 String str = new String("");  
 for (int i = 0; i < this.num; i++) {  
 str = str + "Computer{" +  
 "info=" + info.get(i) +  
 '}' + '\n';  
 }  
 return str;  
 }  
  
 public class Info {  
 int proc;  
 String os;  
 int mem;  
  
 public Info(int proc, String os, int mem) {  
 this.proc = proc;  
 this.os = os;  
 this.mem = mem;  
 }  
  
 @Override  
 public String toString() {  
 return "Transport{" +  
 "time=" + proc +  
 ", way='" + os + '\'' +  
 ", cost=" + mem +  
 '}';  
 }  
 }  
  
 public static void main(String[] args) {  
 int[] processor = new int[]{1,2,3,4,5};  
 String[] operation = new String[]{"one", "two", "three", "four", "five"};  
 int[] memory = new int[]{6,7,8,9,10};  
  
 Computer comp = new Computer(5, processor, operation, memory);  
 System.*out*.println(comp.toString());  
 }  
}

**Вывод:**

В ходе выполнения лабораторной работы, получены навыки работы с внутренними классами, интерфейсами. А именно: созданы различные классы их описано их взаимодействие на языке Java.

Репозиторий с кодом: <https://github.com/AnnaZav/Java_labs_bigdata.git>