ЛАБОРАТОРНАЯ РАБОТА №6

ПРОЕКТИРОАНИЕ КЛАССОВ

**Цель:** получить основные понятия и навыки по проектированию и

созданию классов с использованием UML.

**Необходимая теоретическая подготовка:**

* основы UML;
* объектно-ориентированное программирование;

**Задания к лабораторной работе**

Общее для всех вариантов: реализовать механизм сериализации

(десериализации) объекта согласно индивидуальному заданию

Год. Определить иерархию года. Создать несколько объектов-месяцев.

Собрать год

### Листинг

import java.io.\*;

public class Main {

public static void main(String[] args) {

Month[] mYear;

Month january = new Month("January");

Month february = new Month("February");

Month march= new Month("March");

Month april = new Month("April");

Month may = new Month("May");

Month june = new Month("June");

Month july= new Month("July");

Month august = new Month("August");

Month sept = new Month("September");

Month oct= new Month("October");

Month nov = new Month("November");

Month dec = new Month("December");.

Year year = new Year(new Month[]{january, february,march,april,may,june,july,august,sept,oct,nov,dec});

try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("year.ser"))) {

out.writeObject(year);

System.out.println("Объект успешно сериализован в файл year.ser");

} catch (IOException e) {

e.printStackTrace();

}

try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("year.ser"))) {

Year deserializedYear = (Year) in.readObject();

System.out.println("Объект успешно десериализован:");

System.out.println(deserializedYear);

} catch (IOException | ClassNotFoundException e) {

e.printStackTrace();

}

}

}

import java.io.Serializable;

public class Year implements Serializable {

private Month[] months;

public Year(Month[] months) {

this.months = months;

}

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

for (Month month : months) {

sb.append(month.toString()).append("\n");

}

return sb.toString();

}

}

import java.io.Serializable;

public class Month implements Serializable {

private String name;

public Month(String name) {

this.name = name;

}

@Override

public String toString() {

return name;

}

}

**Результат**

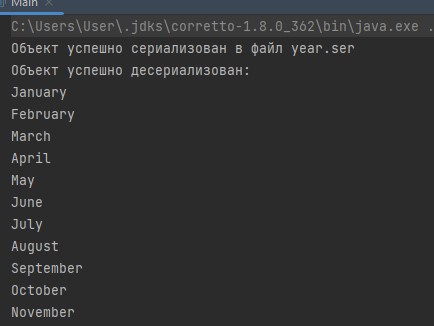


Рисунок 1- результат работы

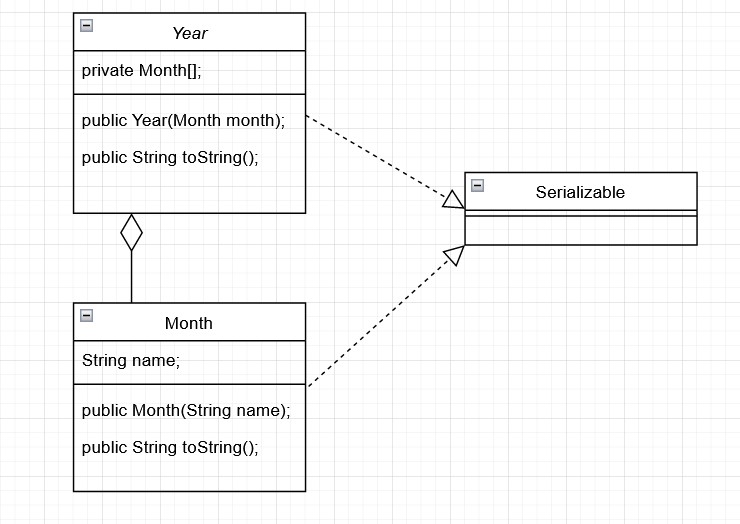


Рисунок 2- диаграмма классов

**Вывод:** Научились проектировать и создавать классы