

**Министерство образования Республики Беларусь
Учреждение Образования
«Брестский Государственный Технический Университет»
Кафедра ИИТ**

**Лабораторная работа №3
По дисциплине СПП за 5 семестр
Тема: «Java»**

Выполнил:
Студент 3-го курса
Группы ПО-5
Крощук В.В.
Проверил:
Крощенко А.А.

Брест 2021

Лабораторная работа №3

Цель работы:

научиться создавать и использовать классы в программах на языке программирования Java

Вариант: 8

Задания и выполненные решения:

Реализовать простой класс.

Требования к выполнению

- *Реализовать пользовательский класс по варианту.*
- *Создать другой класс с методом `main`, в котором будут находиться примеры использования пользовательского класса.*

Для каждого класса

- *Создать поля классов*
- *Создать методы классов*
- *Добавьте необходимые `get` и `set` методы (по необходимости)*
- *Укажите соответствующие модификаторы видимости*
- *Добавьте конструкторы*
- *Переопределить методы `toString()` и `equals()`*

1. ***Множество целых чисел переменной мощности – Предусмотреть возможность пересечения двух множеств, вывода на печать элементов множества, а так же метод, определяющий, принадлежит ли указанное значение множеству. Класс должен содержать методы, позволяющие добавлять и удалять элемент в/из множества. Конструктор должен позволить создавать объекты с начальной инициализацией. Реализацию множества осуществить на базе структуры `ArrayList`. Реализовать метод `equals`, выполняющий сравнение объектов данного типа.***

Код:

Main.java

```
package com.company;
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        IntegerSet integerSet1 = new IntegerSet();
        integerSet1.addItem(123);
        integerSet1.addItem(230);
        integerSet1.addItem(456);
        ArrayList<Integer> integerList = new ArrayList() {{
            add(10);
        }}
    }
}
```

```

        add(12);
        add(456);
    });
    IntegerSet integerSet2 = new IntegerSet(integerList);

    System.out.println("Первое множество: " + integerSet1);
    System.out.println("Второе множество: " + integerSet2);

    System.out.println("integerSet1 == integerSet2? - " +
(integerSet1.equals(integerSet2)));

    System.out.println("Добавление двух элементов ...");
    integerSet2.addItem(123);
    integerSet1.addItem(10);
    System.out.println("Первое множество: " + integerSet1);
    System.out.println("Второе множество: " + integerSet2);

    System.out.println("Пересечение: " +
integerSet2.intersections(integerSet1));

    System.out.println("Второй элемент integerSet2 = " +
integerSet2.getItemById(2));
    System.out.println("integerSet2 содержит 12? - " +
integerSet2.contains(12));

    System.out.println("Удаление элемента... ");
    integerSet1.deleteItemById(2);
    integerSet2.deleteItemById(3);
    System.out.println("Первое множество: " + integerSet1);
    System.out.println("Второе множество: " + integerSet2);
}
}

```

IntegerSet.java

```

package com.company;
import java.util.ArrayList;

public class IntegerSet {
    private ArrayList<Integer> setOfIntegers;

    public IntegerSet() {
        this.setOfIntegers = new ArrayList();
    }

    public IntegerSet(ArrayList<Integer> setOfIntegers) {
        this.setOfIntegers = new ArrayList<>();
    }
}

```

```

        for(Integer el: setOfIntegers) {
            if (!this.setOfIntegers.contains(el))
                this.setOfIntegers.add(el);
        }
    }

    public ArrayList<Integer> intersections(IntegerSet set) {
        ArrayList<Integer> list = set.getSetOfIntegers();
        list.retainAll(this.setOfIntegers);
        return list;
    }

    public boolean contains(int item) {
        return this.setOfIntegers.contains(item);
    }

    public int getItemById(int id) {
        --id;
        return (Integer)this.setOfIntegers.get(id);
    }

    public void addItem(int item) {
        this.setOfIntegers.add(item);
    }

    public void deleteItemById(int id) {
        this.setOfIntegers.remove(id);
    }

    public ArrayList<Integer> getSetOfIntegers() {
        return this.setOfIntegers;
    }

    public void setSetOfIntegers(ArrayList<Integer> setOfIntegers) {
        this.setOfIntegers = setOfIntegers;
    }

    @Override
    public String toString() {
        return "IntegerSet = " + this.setOfIntegers;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) {
            return true;
        } else if (o != null && this.getClass() == o.getClass()) {
            IntegerSet that = (IntegerSet)o;
            return this.setOfIntegers.equals(that.setOfIntegers);
        } else {
            return false;
        }
    }
}

```

Результат:

```
C:\Users\Viktor\.jdk\openjdk-17\bin\java.exe "-javaagent:0
Первое множество: IntegerSet = [123, 230, 456]
Второе множество: IntegerSet = [10, 12, 456]
integerSet1 == integerSet2? - false
Добавление двух элементов ...
Первое множество: IntegerSet = [123, 230, 456, 10]
Второе множество: IntegerSet = [10, 12, 456, 123]
Пересечение: [123, 456, 10]
Второй элемент integerSet2 = 12
integerSet2 содержит 12? - true
Удаление элемента...
Первое множество: IntegerSet = [123, 456]
Второе множество: IntegerSet = [10, 12, 456]

Process finished with exit code 0
```

Выполнение происходит согласно поставленной задачи. Работает корректно!

2. *Разработать автоматизированную систему на основе некоторой структуры данных, манипулирующей объектами пользовательского класса. Реализовать требуемые функции обработки данных.*

Автоматизированная система обработки информации об авиарейсах
Написать программу для обработки информации об авиарейсах (Airlines):

Каждый рейс имеет следующие характеристики:

- Пункт назначения;
- Номер рейса;
- Тип самолета;
- Время вылета;
- Дни недели, по которым совершаются рейсы.

Программа должна обеспечить:

- Генерацию списка рейсов;
- Вывод списка рейсов для заданного пункта назначения;
- Вывод списка рейсов для заданного дня недели;
- Вывод списка рейсов для заданного дня недели, время вылета для которых больше заданного;
- Все рейсы самолетов некоторого типа;
- Группировка рейсов по числу пассажиров (маломестные - 1-100 чел, средместные (100-200), крупные рейсы (200-350));

Код:

Main.java

```
package com.company;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;

public class Main {

    private static final SimpleDateFormat dateFormat = new
SimpleDateFormat("dd.MM.yyyy HH:mm:ss");
    public static void main(String[] args) throws ParseException {
        Airline airline = getAirline();
        System.out.println("-----Все полёты-----");
        airline.printListOfFlights();
        System.out.println("\nВылеты с выбранным расстоянием: ");
        airline.printListOfFlightsForSelectedDestination("Брест");
        System.out.println("\nВылеты по выбранным дням: ");
        airline.printListOfFlightsForSelectedDay(DayOfWeek.MONDAY);
        System.out.println("\nВылеты по выбранному дню и времени: ");
        airline.printListOfFlightsForSelectedDayAndTime(DayOfWeek.SUNDAY,
dateFormat.parse("15.10.2021 03:12:10"));
        System.out.println("\nВылеты по выбранному типу: ");
        airline.printListOfFlightsForSelectedType(.TypeOfAircraft.AVERAGE);
    }
    private static Airline getAirline() throws ParseException {
        Airline airline = new Airline(new ArrayList<>());

        airline.addFlights(new Flight("Брест", 1, TypeOfAircraft.AVERAGE,
dateFormat.parse("12.09.2020 10:12:10"),
Arrays.asList(DayOfWeek.MONDAY, DayOfWeek.FRIDAY)));

        airline.addFlights(new Flight("Витебск", 2, TypeOfAircraft.SMALL,
dateFormat.parse("14.10.2020 10:56:10"),
Arrays.asList(DayOfWeek.THURSDAY, DayOfWeek.FRIDAY)));

        airline.addFlights(new Flight("Гродно", 3, TypeOfAircraft.MAJOR,
dateFormat.parse("15.03.2020 09:30:10"),
Arrays.asList(DayOfWeek.SUNDAY, DayOfWeek.TUESDAY)));

        airline.addFlights(new Flight("Могилёв", 4, TypeOfAircraft.AVERAGE,
dateFormat.parse("19.02.2023 11:42:22"),
Arrays.asList(DayOfWeek.WEDNESDAY, DayOfWeek.FRIDAY)));

        airline.addFlights(new Flight("Гомель", 5, TypeOfAircraft.MAJOR,
dateFormat.parse("15.03.2020 09:30:10"),
Arrays.asList(DayOfWeek.SUNDAY, DayOfWeek.WEDNESDAY)));

        airline.addFlights(new Flight("Минск", 6, TypeOfAircraft.AVERAGE,
dateFormat.parse("14.10.2017 08:21:11"),
```

```

        Arrays.asList(DayOfWeek.MONDAY, DayOfWeek.TUESDAY)));

        airline.addFlights(new Flight("Bpect", 7, TypeOfAircraft.SMALL,
dateFormat.parse("15.10.2013 05:13:10"),
        Arrays.asList(DayOfWeek.WEDNESDAY, DayOfWeek.FRIDAY)));

        return airline;
    }
}

```

Airline.java

```

package com.company;
import java.util.Date;
import java.util.List;

public class Airline {
    private List<Flight> flights;
    public Airline(List<Flight> flights) {
        this.flights = flights;
    }
    public List<Flight> getFlights() {
        return flights;
    }
    public void setFlights(List<Flight> flights) {
        this.flights = flights;
    }
    public void addFlights(Flight flight) {
        flights.add(flight);
    }
    public void deleteFlights(int id) {
        flights.remove(--id);
    }
    public void printListOfFlights() {
        flights.forEach(System.out::println);
    }
    public void printListOfFlightsForSelectedDestination(String destination)
{
        flights.stream().filter(flight ->

flight.getDestination().equals(destination)).forEach(System.out::println);
    }
    public void printListOfFlightsForSelectedDay(DayOfWeek dayOfWeek) {
        flights.stream().filter(flight ->

flight.getDaysOfTheWeek().contains(dayOfWeek)).forEach(System.out::println);
    }
    public void printListOfFlightsForSelectedDayAndTime(DayOfWeek dayOfWeek,
Date date)
    {
        flights.stream().filter(flight ->
            flight.getDaysOfTheWeek().contains(dayOfWeek))

```

```

        .filter(flight ->

flight.getDepartureTime().before(date)).forEach(System.out::println);
    }
    public void printListOfFlightsForSelectedType(.TypeOfAircraft
typeOfAircraft) {
        flights.stream().filter(flight ->

flight.getTypeOfAircraft().equals(typeOfAircraft)).forEach(System.out::print
ln);
    }
}

```

Flight.java

```

package com.company;

import java.util.Date;
import java.util.List;

public class Flight {
    private String destination;
    private int flightNumber;
    private TypeOfAircraft typeOfAircraft;
    private Date departureTime;
    private List<DayOfWeek> daysOfTheWeek;
    public Flight(String destination, int flightNumber, TypeOfAircraft
typeOfAircraft,
                    Date departureTime,
                    List<DayOfWeek> daysOfTheWeek) {
        this.destination = destination;
        this.flightNumber = flightNumber;
        this.typeOfAircraft = typeOfAircraft;
        this.departureTime = departureTime;
        this.daysOfTheWeek = daysOfTheWeek;
    }
    public String getDestination() {
        return destination;
    }
    public void setDestination(String destination) {
        this.destination = destination;
    }
    public int getFlightNumber() {
        return flightNumber;
    }
    public void setFlightNumber(int flightNumber) {
        this.flightNumber = flightNumber;
    }
    public TypeOfAircraft getTypeOfAircraft() {
        return typeOfAircraft;
    }
    public void setTypeOfAircraft(TypeOfAircraft typeOfAircraft) {

```



```

        this.typeOfAircraft = typeOfAircraft;
    }
    public Date getDepartureTime() {
        return departureTime;
    }
    public void setDepartureTime(Date departureTime) {
        this.departureTime = departureTime;
    }
    public List<DayOfWeek> getDaysOfTheWeek() {
        return daysOfTheWeek;
    }
    public void setDaysOfTheWeek(List<DayOfWeek> daysOfTheWeek) {
        this.daysOfTheWeek = daysOfTheWeek;
    }
    @Override
    public String toString() {
        return "Полёты{" +
            "Расстояние= '" + destination + '\'' +
            ", Номер вылета= " + flightNumber +
            ", Тип самолета= " + typeOfAircraft +
            ", Время отправл= " + departureTime +
            ", День недели= " + daysOfTheWeek +
            '}';
    }
}

```

DayOfWeek.java

```

package com.company;

public enum DayOfWeek {
    MONDAY(), TUESDAY(), WEDNESDAY(), THURSDAY(), FRIDAY(), SATURDAY(),
    SUNDAY()
}

```

TypeOfAircraft.java

```

package com.company;

public enum TypeOfAircraft {
    SMALL(), AVERAGE(), MAJOR()
}

```

Результат:

```
C:\Users\Viktor\.jdk\openjdk-17\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.3\lib\idea_rt.jar=63618:C:\P
-----Все полёты-----
Полёты{Расстояние= 'Брест', Номер вылета= 1, Тип самолета= AVERAGE, Время отправл= Sat Sep 12 10:12:10 MSK 2020, День недели= [MONDAY, FRIDAY]}
Полёты{Расстояние= 'Витебск', Номер вылета= 2, Тип самолета= SMALL, Время отправл= Wed Oct 14 10:56:10 MSK 2020, День недели= [THURSDAY, FRIDAY]}
Полёты{Расстояние= 'Гродно', Номер вылета= 3, Тип самолета= MAJOR, Время отправл= Sun Mar 15 09:30:10 MSK 2020, День недели= [SUNDAY, TUESDAY]}
Полёты{Расстояние= 'Могилёв', Номер вылета= 4, Тип самолета= AVERAGE, Время отправл= Sun Feb 19 11:42:22 MSK 2023, День недели= [WEDNESDAY, FRIDAY]}
Полёты{Расстояние= 'Гомель', Номер вылета= 5, Тип самолета= MAJOR, Время отправл= Sun Mar 15 09:30:10 MSK 2020, День недели= [SUNDAY, WEDNESDAY]}
Полёты{Расстояние= 'Минск', Номер вылета= 6, Тип самолета= AVERAGE, Время отправл= Sat Oct 14 08:21:11 MSK 2017, День недели= [MONDAY, TUESDAY]}
Полёты{Расстояние= 'Брест', Номер вылета= 7, Тип самолета= SMALL, Время отправл= Tue Oct 15 05:13:10 MSK 2013, День недели= [WEDNESDAY, FRIDAY]}

Вылеты с выбранным расстоянием:
Полёты{Расстояние= 'Брест', Номер вылета= 1, Тип самолета= AVERAGE, Время отправл= Sat Sep 12 10:12:10 MSK 2020, День недели= [MONDAY, FRIDAY]}
Полёты{Расстояние= 'Брест', Номер вылета= 7, Тип самолета= SMALL, Время отправл= Tue Oct 15 05:13:10 MSK 2013, День недели= [WEDNESDAY, FRIDAY]}

Вылеты по выбранным дням:
Полёты{Расстояние= 'Брест', Номер вылета= 1, Тип самолета= AVERAGE, Время отправл= Sat Sep 12 10:12:10 MSK 2020, День недели= [MONDAY, FRIDAY]}
Полёты{Расстояние= 'Минск', Номер вылета= 6, Тип самолета= AVERAGE, Время отправл= Sat Oct 14 08:21:11 MSK 2017, День недели= [MONDAY, TUESDAY]}

Вылеты по выбранному дню и времени:
Полёты{Расстояние= 'Гродно', Номер вылета= 3, Тип самолета= MAJOR, Время отправл= Sun Mar 15 09:30:10 MSK 2020, День недели= [SUNDAY, TUESDAY]}
Полёты{Расстояние= 'Гомель', Номер вылета= 5, Тип самолета= MAJOR, Время отправл= Sun Mar 15 09:30:10 MSK 2020, День недели= [SUNDAY, WEDNESDAY]}

Вылеты по выбранному типу:
Полёты{Расстояние= 'Брест', Номер вылета= 1, Тип самолета= AVERAGE, Время отправл= Sat Sep 12 10:12:10 MSK 2020, День недели= [MONDAY, FRIDAY]}
Полёты{Расстояние= 'Могилёв', Номер вылета= 4, Тип самолета= AVERAGE, Время отправл= Sun Feb 19 11:42:22 MSK 2023, День недели= [WEDNESDAY, FRIDAY]}
Полёты{Расстояние= 'Минск', Номер вылета= 6, Тип самолета= AVERAGE, Время отправл= Sat Oct 14 08:21:11 MSK 2017, День недели= [MONDAY, TUESDAY]}

Process finished with exit code 0
```

Исходя из полученных результатов, все работает корректно, согласно поставленному заданию.

Вывод: научился создавать и использовать классы в программах на языке программирования Java.