

# Звіт

## Лабораторна робота 9. Параметризація в Java

**Мета роботи:** Оволодіння навичками управління введенням / виведенням даних з використанням класів платформи Java SE.

### ВИМОГИ

- Вивчення принципів параметризації в Java.
- Розробка параметризованих класів та методів.

**1.1. Розробник:** Капелька Ярослав Іванович, КІТ-119а, варіант №9.

## 2. ОПИС ПРОГРАМИ

**2.1. Засоби ООП:** клас, метод класу, поле класу.

**2.2. Ієрархія та структура класів:** один публічний клас Main, публічний клас Route, у якого є поля: назва маршруту, загальна кількість місць, дні тижня; номер рейсу, назва станції, час прибуття, час відправлення, кількість вільних місць, статус станції, гетери, сетери, конструктор класу та метод виведення даних класу. Також є клас Data, який виконує роль покажчика на елемент і клас MyCollection, який містить покажчик на головний елемент та методи обробки масиву елементів.

**2.3. Важливі фрагменти програми:**

Data.java

```
public class Data <T>
{
    public T obj;

    public Data<T> next;

    public Data<T> prev;

    Data()
    {
    }

    Data(T obj, Data<T> prev, Data<T> next)
    {
        this.obj = obj;
        this.next = next;
        this.prev = prev;
    }
}
```

Main09.java

```
public class Main09
```

```

{
    public static void main(String[] args)
    {
        MyCollection<Route> ticketOffice = new MyCollection<Route>();
        Route routeData = new Route("Лозовая-
Харьков", "Лозовая", "15:20", "15:24", 55, "Начальная", 150, new GregorianCalendar(2021, 11,
7), 1);
        ticketOffice.add(routeData);
        ticketOffice.add( new Route("Харьков-
Изюм", "Маяк", "14:52", "14:53", 61, "Промежуточная", 200, new GregorianCalendar(2021, 11,
8), 2));
        ticketOffice.add( new Route("Красноград-
Харьков", "Харьков", "19:34", "Времени отправления нету, так как это конечная
станция.", 78, "Конечная", 170, new GregorianCalendar(2021, 1, 20), 3));
        ticketOffice.add( new Route("Львов-
Мариуполь", "Льзов", "01:07", "01:12", 55, "Начальная", 230, new GregorianCalendar(2021, 2,
15), 4));
        ticketOffice.add( new Route("Ужгород-
Киев", "Мукачево", "17:59", "18:05", 69, "Промежуточная", 200, new GregorianCalendar(2021,
2, 25), 5));

        System.out.println("Происходит поиск элемента в коллекции...");
        System.out.println(ticketOffice.find(routeData));
        if (ticketOffice.find(routeData) == true)
        {
            System.out.println("Элемент в коллекции найден.");
        }
        else if (ticketOffice.find(routeData) == false)
        {
            System.out.println("Элемент в коллекции не найден.");
        }
        System.out.println("-----");
        System.out.println("Вывод коллекции:");
        System.out.println(ticketOffice);
        System.out.println("-----");
        System.out.println("Происходит удаление элемента из коллекции...");
        ticketOffice.delete(4);
        System.out.println("Элемент успешно удален");
        System.out.println("-----");
        System.out.println("Вывод коллекции с удаленным элементом:");
        System.out.println(ticketOffice);
        System.out.println("-----");
        System.out.println("Происходит сериализация и десериализация
коллекции...");
        try
        {
            ticketOffice.saveXml("SaveXML.xml");

            ticketOffice.clear();

            System.out.println(ticketOffice);

            ticketOffice.downloadXml("SaveXML.xml");

            System.out.println(ticketOffice);

        } catch (FileNotFoundException e)
        {
            e.printStackTrace();
        }

        System.out.println("Вывод коллекции в виде массива:");
        Route []mass = ticketOffice.toArray(new Route[ticketOffice.getSize()]);
        for(Route p : mass)
        {
            System.out.println(p);
        }
    }
}

```

```

    }
}
}

```

## MyCollection.java

```

public class MyCollection <T> implements Iterable<T>, Serializable
{
    static final long serialVersionUID = 1L;
    private int size;
    private Data<T> start;

    private Data<T> last;

    public void saveSer(String fileName) throws IOException
    {
        FileOutputStream outputStream = new FileOutputStream(fileName);
        ObjectOutputStream objectOutputStream = new
ObjectOutputStream(outputStream);
        for(T value : this)
            objectOutputStream.writeObject(value);
        objectOutputStream.close();
    }
    @SuppressWarnings("unchecked")
    public void downloadSer(String fileName) throws IOException,
ClassNotFoundException
    {
        FileInputStream inStream = new FileInputStream(fileName);
        ObjectInputStream objectInStream = new ObjectInputStream(inStream);
        try
        {
            while(true)
            {
                add((T) objectInStream.readObject());
            }
        }catch(EOFException e)
        {
            objectInStream.close();
        }
    }

    public boolean find(T obj)
    {
        for(T value : this)
        {
            if(value.equals(obj))
                return true;
        }
        return false;
    }

    public String toString()
    {
        String str = new String();
        for(T value : this)
        {
            str += value + "\n";
        }
        return str;
    }

    public void clear()

```

```

{
    start.next = last;
    last.prev = start;
    size = 0;
}

public void saveXml(String fileName) throws FileNotFoundException
{
    XMLEncoder encoder = new XMLEncoder(new BufferedOutputStream(new
FileOutputStream(fileName)));

    for(T value : this )
        encoder.writeObject(value);
    encoder.close();
    System.out.println("Сериализация прошла успешно\n");
}

@SuppressWarnings("unchecked")
public void downloadXml(String fileName) throws FileNotFoundException
{
    XMLDecoder d = new XMLDecoder(new BufferedInputStream(new
FileInputStream(fileName)));
    try
    {
        while(true)
        {
            add((T) d.readObject());
        }
    } catch (ArrayIndexOutOfBoundsException e)
    {
        d.close();
        System.out.println("Десериализация прошла успешно\n");
    } catch (Exception e)
    {
        throw e;
    }
}

public int getSize()
{
    return size;
}

MyCollection()
{
    size = 0;
    start = new Data<T>(null, null, last);
    last = new Data<T>(null, start, null);
}

void add(T obj)
{
    size++;
    Data<T> temp = last.prev;
    temp.next = new Data<T>();
    last.prev = temp.next;

    temp.next.obj = obj;

    temp.next.next = last;
    temp.next.prev = temp;
}

```

```

void delete(int itr)
{
    if(itr>=size)
        return;

    Data<T> temp = start.next;
    Data<T> temp2 = null;

    for(int i = 0; i < itr;i++)
    {
        temp = temp.next;
    }

    temp2 = temp.prev;
    temp2.next = temp.next;
    temp.next.prev = temp2;

    size--;
}
T get(int itr)
{
    if(itr>=size && itr<0)
        return null;

    Data<T> temp = start.next;

    for(int i = 0; i < itr;i++)
    {
        temp = temp.next;
    }

    return temp.obj;
}

public T[] toArray(T[] arr)
{
    for(int i = 0; i < size; i ++)
    {
        arr[i] = get(i);
    }
    return arr;
}

@Override
public Iterator<T> iterator()
{
    return new Iterator<T>()
    {
        int itr = 0;
        @Override
        public boolean hasNext()
        {
            return itr < size;
        }

        @Override
        public T next()
        {
            return get(itr++);
        }
        @Override

```

```

        public void remove()
        {
            delete(itr-1);
        }
    };
}

```

Route.java

```

public class Route implements Serializable
{
    private static final long serialVersionUID = 1L;
    private String name_route;
    private String station_name;
    private String departure_time;
    private String arrival_time;
    private int number_of_free_seats;
    private String status_station;
    private int total_number_of_seats;
    private GregorianCalendar days_of_the_week;
    private int flight_number;
    public void setNameRoute(String name_route)
    {
        this.name_route = name_route;
    }
    public String getNameRoute()
    {
        return name_route;
    }
    public String getStation_name()
    {
        return station_name;
    }
    public void setStation_name(String station_name)
    {
        this.station_name = station_name;
    }
    public String getDeparture_time()
    {
        return departure_time;
    }
    public void setDeparture_time(String departure_time)
    {
        this.departure_time = departure_time;
    }
    public String getArrival_time()
    {
        return arrival_time;
    }
    public void setArrival_time(String arrival_time)
    {
        this.arrival_time = arrival_time;
    }
    public int getNumber_of_free_seats()
    {
        return number_of_free_seats;
    }
    public void setNumber_of_free_seats(int number_of_free_seats)
    {
        this.number_of_free_seats = number_of_free_seats;
    }
    public String getStatus_station()
    {
        return status_station;
    }
}

```

```

    }
    public void setStatus_station(String status_station)
    {
        this.status_station = status_station;
    }
    public void setTotal_number_of_seats(int total_number_of_seats)
    {
        this.total_number_of_seats = total_number_of_seats;
    }
    public int getTotal_number_of_seats()
    {
        return total_number_of_seats;
    }
    public void setDays_of_the_week(GregorianCalendar days_of_the_week)
    {
        this.days_of_the_week = days_of_the_week;
    }
    public GregorianCalendar getDays_of_the_week()
    {
        return days_of_the_week;
    }
    public int getFlight_number()
    {
        return flight_number;
    }
    public void setFlight_number(int flight_number)
    {
        this.flight_number = flight_number;
    }

    public Route()
    {
        super();
    }
    public void print()
    {
        System.out.println("\nИмя маршрута: " + this.getNameRoute()+"\nИмя
станции: " + this.getStation_name() + "\nВремя пребывания на станцию: " +
this.getArrival_time()+ "\nВремя отправления со станции: " +
this.getDeparture_time()+"\nКоличество пустых мест: "+
this.getNumber_of_free_seats()+"\nСтатус станции: "+
this.getStatus_station()+"\nОбщее количество мест: "+
this.getTotal_number_of_seats()+"\nДень недели: "+
this.getDays_of_the_week().getTime()+"\nНомер рейсу: " + this.getFlight_number());
    }

    @Override
    public String toString()
    {
        return new String("\nИмя маршрута: " + this.getNameRoute()+"\nИмя
станции: " + this.getStation_name() + "\nВремя пребывания на станцию: " +
this.getArrival_time()+ "\nВремя отправления со станции: " +
this.getDeparture_time()+"\nКоличество пустых мест: "+
this.getNumber_of_free_seats()+"\nСтатус станции: "+
this.getStatus_station()+"\nОбщее количество мест: "+
this.getTotal_number_of_seats()+"\nДень недели: "+
this.getDays_of_the_week().getTime()+"\nНомер рейсу: " + this.getFlight_number());
    }
    Route(String name, int total_number, GregorianCalendar days, int flight)
    {
        this.setNameRoute(name);
        setTotal_number_of_seats(total_number);
        this.setDays_of_the_week(days);
    }

```

```
        setFlight_number(flight);
    }
    Route(String name, String name1, String time,String time1, int number, String
status, int total_number, GregorianCalendar days,int flight)
    {
        this.setNameRoute(name);
        this.setStation_name(name1);
        this.setArrival_time(time);
        this.setDeparture_time(time1);
        this.setNumber_of_free_seats(number);
        this.setStatus_station(status);
        setTotal_number_of_seats(total_number);
        this.setDays_of_the_week(days);
        setFlight_number(flight);
    }
}
```

## Результат роботи програми



Происходит поиск элемента в коллекции...

true

Элемент в коллекции найден.

-----

Вывод коллекции:

Имя маршрута: Лозовая-Харьков

Имя станции: Лозовая

Время пребывания на станцию: 15:20

Время отправления со станции: 15:24

Количество пустых мест: 55

Статус станции: Начальная

Общее количество мест: 150

День недели: Tue Dec 07 00:00:00 EET 2021

Номер рейсу: 1

Имя маршрута: Харьков-Изюм

Имя станции: Маяк

Время пребывания на станцию: 14:52

Время отправления со станции: 14:53

Количество пустых мест: 61

Статус станции: Промежуточная

Общее количество мест: 200

День недели: Wed Dec 08 00:00:00 EET 2021

Номер рейсу: 2

Имя маршрута: Красноград-Харьков

Имя станции: Харьков

Время пребывания на станцию: 19:34

Время отправления со станции: Времени отправления нету, так как это конечная станция.

Количество пустых мест: 78

Статус станции: Конечная

Общее количество мест: 170

День недели: Sat Feb 20 00:00:00 EET 2021

Номер рейсу: 3

Имя маршрута: Львов-Мариуполь

Имя станции: Льзов

Время пребывания на станцию: 01:07

Время отправления со станции: 01:12

Количество пустых мест: 55

Статус станции: Начальная

Общее количество мест: 230

День недели: Mon Mar 15 00:00:00 EET 2021

Номер рейсу: 4

Имя маршрута: Ужгород-Киев

Имя станции: Мукачево

Время пребывания на станцию: 17:59

Время отправления со станции: 18:05  
Количество пустых мест: 69  
Статус станции: Промежуточная  
Общее количество мест: 200  
День недели: Thu Mar 25 00:00:00 EET 2021  
Номер рейсу: 5

-----  
Происходит удаление элемента из коллекции...  
Элемент успешно удален  
-----

Вывод коллекции с удаленным элементом:

Имя маршрута: Лозовая-Харьков  
Имя станции: Лозовая  
Время пребывания на станцию: 15:20  
Время отправления со станции: 15:24  
Количество пустых мест: 55  
Статус станции: Начальная  
Общее количество мест: 150  
День недели: Tue Dec 07 00:00:00 EET 2021  
Номер рейсу: 1

Имя маршрута: Харьков-Изюм  
Имя станции: Маяк  
Время пребывания на станцию: 14:52  
Время отправления со станции: 14:53  
Количество пустых мест: 61  
Статус станции: Промежуточная  
Общее количество мест: 200  
День недели: Wed Dec 08 00:00:00 EET 2021  
Номер рейсу: 2

Имя маршрута: Красноград-Харьков  
Имя станции: Харьков  
Время пребывания на станцию: 19:34  
Время отправления со станции: Времени отправления нету, так как это конечная станция.  
Количество пустых мест: 78  
Статус станции: Конечная  
Общее количество мест: 170  
День недели: Sat Feb 20 00:00:00 EET 2021  
Номер рейсу: 3

Имя маршрута: Львов-Мариуполь  
Имя станции: Львов

Время пребывания на станцию: 01:07  
Время отправления со станции: 01:12  
Количество пустых мест: 55  
Статус станции: Начальная  
Общее количество мест: 230  
День недели: Mon Mar 15 00:00:00 EET 2021  
Номер рейсу: 4

-----  
Происходит сериализация и десериализация коллекции...  
java.lang.NoSuchMethodException: <unbound>=ZoneInfo.getRawOffset();  
Continuing ...  
java.lang.NoSuchMethodException: <unbound>=ZoneInfo.getRawOffset();  
Continuing ...  
java.lang.NoSuchMethodException: <unbound>=ZoneInfo.getRawOffset();  
Continuing ...  
java.lang.NoSuchMethodException: <unbound>=ZoneInfo.getRawOffset();  
Continuing ...  
Сериализация прошла успешно

Десериализация прошла успешно

Имя маршрута: Лозовая-Харьков  
Имя станции: Лозовая  
Время пребывания на станцию: 15:20  
Время отправления со станции: 15:24  
Количество пустых мест: 55  
Статус станции: Начальная  
Общее количество мест: 150  
День недели: Tue Dec 07 00:00:00 EET 2021  
Номер рейсу: 1

Имя маршрута: Харьков-Изюм  
Имя станции: Маяк  
Время пребывания на станцию: 14:52  
Время отправления со станции: 14:53  
Количество пустых мест: 61  
Статус станции: Промежуточная  
Общее количество мест: 200  
День недели: Wed Dec 08 00:00:00 EET 2021  
Номер рейсу: 2

Имя маршрута: Красноград-Харьков  
Имя станции: Харьков  
Время пребывания на станцию: 19:34  
Время отправления со станции: Времени отправления нету, так как это конечная станция.  
Количество пустых мест: 78  
Статус станции: Конечная  
Общее количество мест: 170  
День недели: Sat Feb 20 00:00:00 EET 2021  
Номер рейсу: 3

Имя маршрута: Львов-Мариуполь  
Имя станции: Льзов  
Время пребывания на станцию: 01:07  
Время отправления со станции: 01:12  
Количество пустых мест: 55  
Статус станции: Начальная  
Общее количество мест: 230  
День недели: Mon Mar 15 00:00:00 EET 2021  
Номер рейсу: 4

Вывод коллекции в виде массива:

Имя маршрута: Лозовая-Харьков  
Имя станции: Лозовая  
Время пребывания на станцию: 15:20  
Время отправления со станции: 15:24  
Количество пустых мест: 55  
Статус станции: Начальная  
Общее количество мест: 150  
День недели: Tue Dec 07 00:00:00 EET 2021  
Номер рейсу: 1

Имя маршрута: Харьков-Изюм  
Имя станции: Маяк  
Время пребывания на станцию: 14:52  
Время отправления со станции: 14:53  
Количество пустых мест: 61  
Статус станции: Промежуточная  
Общее количество мест: 200  
День недели: Wed Dec 08 00:00:00 EET 2021  
Номер рейсу: 2

Имя маршрута: Красноград-Харьков  
Имя станции: Харьков  
Время пребывания на станцию: 19:34  
Время отправления со станции: Времени отправления нету, так как это конечная станция.  
Количество пустых мест: 78  
Статус станции: Конечная  
Общее количество мест: 170  
День недели: Sat Feb 20 00:00:00 EET 2021  
Номер рейсу: 3

Имя маршрута: Львов-Мариуполь  
Имя станции: Льзов  
Время пребывания на станцию: 01:07  
Время отправления со станции: 01:12  
Количество пустых мест: 55  
Статус станции: Начальная  
Общее количество мест: 230  
День недели: Mon Mar 15 00:00:00 EET 2021  
Номер рейсу: 4

## **Висновки**

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з параметризацією.

Програма протестована, виконується без помилок.