

Лабораторна робота 8. Основи введення/виведення Java SE

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

ВИМОГИ

1. Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання лабораторної роботи №7.
2. Забороняється використання стандартного протокола серіалізації.
3. Продемонструвати використання моделі Long Term Persistence.
4. Забезпечити діалог з користувачем у вигляді простого текстового меню.
5. При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

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 та публічний клас RouteClass, у якого є поля: загальна кількість місць; дні тижня; номер рейсу, гетери, сетери, конструктор класу та метод виведення даних класу, клас StationClass, у якого є поля: назва станції, час прибуття (для проміжних і кінцевої), час відправлення (для початкової та проміжних), кількість вільних місць"; гетери, сетери, конструктор класу та метод виведення даних класу. Також є клас RouteClassList, у полях якого є масив елементів класу RouteClass, розмір масиву, гетери та сетери поля розміру, методи додавання та видалення елементів.

2.3. Важливі фрагменти програми:

```
public class RouteClassList
{
    private int size = 0;
    RouteClass[] route = new RouteClass[size];

    public int getSize()
    {
        return size;
    }

    public void setSize(int size)
    {
        this.size = size;
    }

    public void addElement(RouteClass Route)
    {
        RouteClass[] newRoute = new RouteClass[size + 1];
```

```

        for (int i = 0; i < size; i++)
        {
            newRoute[i] = route[i];
        }

        newRoute[size] = Route;
        size++;
        route = newRoute;
    }

    public void deleteElement(int position)
    {
        if(size != 0)
        {
            RouteClass[] newRoute = new RouteClass[size-1];
            for (int i = 0; i < position-1; i++)
            {
                newRoute[i] = route[i];
            }
            for (int i = position-1, j = position; j < size; i++, j++)
            {
                newRoute[i] = route[j];
            }

            size--;
            route = newRoute;
        }
        else
        {
            System.out.println("Массив пуст.");
        }
    }
}

public class RouteClass
{
    private int size = 0;
    private int total_number_of_seats;
    private GregorianCalendar days_of_the_week;
    private int flight_number;
    StationClass station[] = new StationClass[size];
    public int getSize()
    {
        return size;
    }

    public void setSize(int size)
    {
        this.size = size;
    }

    public int getTotal_number_of_seats()
    {
        return total_number_of_seats;
    }
    public void setTotal_number_of_seats(int total_number_of_seats)
    {
        this.total_number_of_seats = total_number_of_seats;
    }

    public GregorianCalendar getDay_of_the_week()
    {
        return days_of_the_week;
    }
    public void setDays_of_the_week(GregorianCalendar days_of_the_week)
    {

```

```

        this.days_of_the_week = days_of_the_week;
    }
    public int getFlight_number()
    {
        return flight_number;
    }
    public void setFlight_number(int flight_number)
    {
        this.flight_number = flight_number;
    }
    public RouteClass() {};

    public RouteClass(int total_number, GregorianCalendar days, int flight)
    {
        total_number_of_seats = total_number;
        this.days_of_the_week = days;
        flight_number = flight;
    }
    public void outputData()
    {
        System.out.println("Общее количество мест: " + total_number_of_seats);
        System.out.println("День недели: " + days_of_the_week.getTime());
        System.out.println("Номер рейса: " + flight_number);
    }
    public void addElementStation(StationClass Route)
    {
        StationClass[] newStation = new StationClass[size + 1];
        for (int i = 0; i < size; i++)
        {
            newStation[i] = station[i];
        }

        newStation[size] = Route;
        size++;
        station = newStation;
    }

    public void deleteElementStation(int position)
    {
        if(size != 0)
        {
            StationClass[] newStation = new StationClass[size-1];
            for (int i = 0; i < position-1; i++)
            {
                newStation[i] = station[i];
            }
            for (int i = position-1, j = position; j < size; i++, j++)
            {
                newStation[i] = station[j];
            }

            size--;
            station = newStation;
        }
        else
        {
            System.out.println("Массив пуст.");
        }
    }
}

```

```

public class StationClass
{
    private String station_name;
    private String departure_time;
    private String arrival_time;
    private String starting_station;
    private String terminal_station;
    private String intermediate_stations;
    private String departure_time_of_intermediate_stations;
    private String arrival_time_of_intermediate_stations;
    private int number_of_free_seats;

    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 String getStarting_station()
    {
        return starting_station;
    }
    public void setStarting_station(String starting_station)
    {
        this.starting_station = starting_station;
    }
    public String getTerminal_station()
    {
        return terminal_station;
    }
    public void setTerminal_station(String terminal_station)
    {
        this.terminal_station = terminal_station;
    }
    public String getIntermediate_stations()
    {
        return intermediate_stations;
    }
    public void setIntermediate_stations(String intermediate_stations)
    {
        this.intermediate_stations = intermediate_stations;
    }
    public String getDeparture_time_of_intermediate_stations()
    {
        return departure_time_of_intermediate_stations;
    }
    public void setDeparture_time_of_intermediate_stations(String
departure_time_of_intermediate_stations)

```

```

    {
        this.departure_time_of_intermediate_stations =
departure_time_of_intermediate_stations;
    }
    public String getArrival_time_of_intermediate_stations()
    {
        return arrival_time_of_intermediate_stations;
    }
    public void setArrival_time_of_intermediate_stations(String
arrival_time_of_intermediate_stations)
    {
        this.arrival_time_of_intermediate_stations =
arrival_time_of_intermediate_stations;
    }
    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 StationClass() {};

    public StationClass(String name, String time,String time1,String start,String
terminal,String intermediate, String time_of_intermediate, String
time1_of_intermediate,int number)
    {
        this.station_name = name;
        this.departure_time = time;
        this.arrival_time = time1;
        this.starting_station = start;
        this.terminal_station = terminal;
        this.intermediate_stations = intermediate;
        this.departure_time_of_intermediate_stations = time_of_intermediate;
        this.arrival_time_of_intermediate_stations = time1_of_intermediate;
        this.number_of_free_seats = number;
    }
    public void outputDataStation()
    {
        System.out.println("Названия станции: " + station_name);
        System.out.println("Время отправления: " + departure_time);
        System.out.println("Время прибытия: " + arrival_time);
        System.out.println("Начальная станция: " + starting_station);
        System.out.println("Конечная станция: " + terminal_station);
        System.out.println("Промежуточная станция: " + intermediate_stations);
        System.out.println("Время отправления для промежуточных станций: " +
departure_time_of_intermediate_stations);
        System.out.println("Время прибытия для промежуточных станций: " +
arrival_time_of_intermediate_stations);
        System.out.println("Количество пустых мест: " + number_of_free_seats);
    }
}

public class Main08
{
    public static void main(String[] args)
    {
        RouteClassList Route_list = new RouteClassList();
        RouteClass Station_list = new RouteClass();
        String Station_name1 = "Лозовая-Харьков";
        String Station_name2 = "Харьков-Гавриловка";
        String Station_name3 = "Харьков-Изюм";
        String Station_name4 = "Красноград-Харьков";
        String Station_name5 = "Мерефа-Харьков";
        String Station_name6 = "Славянск-Харьков";
    }
}

```

```

String departure_time1 = "15:24";
String departure_time2 = "15:59";
String departure_time3 = "06:40";
String departure_time4 = "16:44";
String departure_time5 = "06:58";
String departure_time6 = "06:27";
String arrival_time1 = "19:13";
String arrival_time2 = "20:37";
String arrival_time3 = "10:14";
String arrival_time4 = "19:34";
String arrival_time5 = "07:45";
String arrival_time6 = "10:51";
String starting_station1 = "Лозовая";
String starting_station2 = "Харьков";
String starting_station3 = "Харьков";
String starting_station4 = "Красноград";
String starting_station5 = "Мерефа";
String starting_station6 = "Славянск";
String terminal_station1 = "Харьков";
String terminal_station2 = "Гавриловка";
String terminal_station3 = "Изюм";
String terminal_station4 = "Харьков";
String terminal_station5 = "Харьков";
String terminal_station6 = "Харьков";
String intermediate_stations1 = "Панютино";
String intermediate_stations2 = "Мерефа";
String intermediate_stations3 = "Маяк";
String intermediate_stations4 = "Власовка";
String intermediate_stations5 = "Покотиловка";
String intermediate_stations6 = "Гусаровка";
String departure_time_of_intermediate_stations1 = "15:35";
String departure_time_of_intermediate_stations2 = "16:43";
String departure_time_of_intermediate_stations3 = "09:24";
String departure_time_of_intermediate_stations4 = "17:37";
String departure_time_of_intermediate_stations5 = "07:25";
String departure_time_of_intermediate_stations6 = "06:56";
String arrival_time_of_intermediate_stations1 = "15:36";
String arrival_time_of_intermediate_stations2 = "16:54";
String arrival_time_of_intermediate_stations3 = "09:25";
String arrival_time_of_intermediate_stations4 = "17:38";
String arrival_time_of_intermediate_stations5 = "07:26";
String arrival_time_of_intermediate_stations6 = "06:57";
int number_of_free_seats1 = 35;
int number_of_free_seats2 = 48;
int number_of_free_seats3 = 72;
int number_of_free_seats4 = 65;
int number_of_free_seats5 = 37;
int number_of_free_seats6 = 86;

GregorianCalendar days_of_the_week1 = new GregorianCalendar(2020, 10,
22);
days_of_the_week1.set(Calendar.HOUR_OF_DAY, 15);
days_of_the_week1.set(Calendar.MINUTE, 24);
days_of_the_week1.set(Calendar.SECOND, 00);
GregorianCalendar days_of_the_week2 = new GregorianCalendar(2020, 10,
25);
days_of_the_week2.set(Calendar.HOUR_OF_DAY, 15);
days_of_the_week2.set(Calendar.MINUTE, 59);
days_of_the_week2.set(Calendar.SECOND, 00);
GregorianCalendar days_of_the_week3 = new GregorianCalendar(2020, 10,
30);
days_of_the_week3.set(Calendar.HOUR_OF_DAY, 6);
days_of_the_week3.set(Calendar.MINUTE, 40);
days_of_the_week3.set(Calendar.SECOND, 00);
GregorianCalendar days_of_the_week4 = new GregorianCalendar(2020, 10,
22);

```

```

        days_of_the_week4.set(Calendar.HOUR_OF_DAY, 16);
        days_of_the_week4.set(Calendar.MINUTE, 44);
        days_of_the_week4.set(Calendar.SECOND, 00);
        GregorianCalendar days_of_the_week5 = new GregorianCalendar(2020, 10,
25);

        days_of_the_week5.set(Calendar.HOUR_OF_DAY, 6);
        days_of_the_week5.set(Calendar.MINUTE, 58);
        days_of_the_week5.set(Calendar.SECOND, 00);
        GregorianCalendar days_of_the_week6 = new GregorianCalendar(2020, 10,
30);

        days_of_the_week6.set(Calendar.HOUR_OF_DAY, 6);
        days_of_the_week6.set(Calendar.MINUTE, 27);
        days_of_the_week6.set(Calendar.SECOND, 00);

        StationClass stationclass1 = new StationClass(Station_name1,
departure_time1, arrival_time1, starting_station1, terminal_station1,
intermediate_stations1, departure_time_of_intermediate_stations1,
arrival_time_of_intermediate_stations1, number_of_free_seats1);
        StationClass stationclass2 = new StationClass(Station_name2,
departure_time2, arrival_time2, starting_station2, terminal_station2,
intermediate_stations2, departure_time_of_intermediate_stations2,
arrival_time_of_intermediate_stations2, number_of_free_seats2);
        StationClass stationclass3 = new StationClass(Station_name3,
departure_time3, arrival_time3, starting_station3, terminal_station3,
intermediate_stations3, departure_time_of_intermediate_stations3,
arrival_time_of_intermediate_stations3, number_of_free_seats3);
        StationClass stationclass4 = new StationClass(Station_name4,
departure_time4, arrival_time4, starting_station4, terminal_station4,
intermediate_stations4, departure_time_of_intermediate_stations4,
arrival_time_of_intermediate_stations4, number_of_free_seats4);
        StationClass stationclass5 = new StationClass(Station_name5,
departure_time5, arrival_time5, starting_station5, terminal_station5,
intermediate_stations5, departure_time_of_intermediate_stations5,
arrival_time_of_intermediate_stations5, number_of_free_seats5);
        StationClass stationclass6 = new StationClass(Station_name6,
departure_time6, arrival_time6, starting_station6, terminal_station6,
intermediate_stations6, departure_time_of_intermediate_stations6,
arrival_time_of_intermediate_stations6, number_of_free_seats6);

        RouteClass routeclass1 = new RouteClass(150, days_of_the_week1, 1);
        RouteClass routeclass2 = new RouteClass(200, days_of_the_week2, 2);
        RouteClass routeclass3 = new RouteClass(250, days_of_the_week3, 3);
        RouteClass routeclass4 = new RouteClass(300, days_of_the_week4, 4);
        RouteClass routeclass5 = new RouteClass(270, days_of_the_week5, 5);
        RouteClass routeclass6 = new RouteClass(120, days_of_the_week6, 6);
        Station_list.addElementStation(stationclass1);
        Station_list.addElementStation(stationclass2);
        Station_list.addElementStation(stationclass3);
        Station_list.addElementStation(stationclass4);
        Station_list.addElementStation(stationclass5);
        Station_list.addElementStation(stationclass6);
        Route_list.addElement(routeclass1);
        Route_list.addElement(routeclass2);
        Route_list.addElement(routeclass3);
        Route_list.addElement(routeclass4);
        Route_list.addElement(routeclass5);
        Route_list.addElement(routeclass6);

        boolean stop = false;
        Scanner scan = new Scanner(System.in);
        int choose;

        while(!stop)
        {
            System.out.println("Какой номер из списка выберете?");
            System.out.println("1. Вывести данные");

```

```

System.out.println("2. Добавить элемент");
System.out.println("3. Удалить элемент");
System.out.println("4. Сериализовать данные");
System.out.println("5. Десериализовать данные");
System.out.println("6. Завершить программу");
System.out.println("-----");
System.out.print("Ваш выбор: ");

choise = scan.nextInt();

switch (choise)
{
case 1:
    System.out.println();
    for (int i = 0; i < Route_list.getSize(); i++)
    {
        System.out.println(i+1 + " ");
        Station_list.station[i].outputDataStation();
        Route_list.route[i].outputData();

        System.out.println();
    }
    break;

case 2:
    System.out.print("\nВведите номер маршрута: ");
    int value = scan.nextInt();
    if(value < 1)
    {
        System.out.print("Ошибка. Неверный размер списка.");
        break;
    }
    scan.nextLine();
    System.out.print("Введите название станции: ");
    String list = scan.nextLine();
    System.out.print("Введите времени отправления: ");
    String list1 = scan.nextLine();
    System.out.print("Введите времени прибытия: ");
    String list2 = scan.nextLine();
    System.out.print("Введите начальную станцию: ");
    String list3 = scan.nextLine();
    System.out.print("Введите конечную станцию: ");
    String list4 = scan.nextLine();
    System.out.print("Введите промежуточную станцию: ");
    String list5 = scan.nextLine();
    System.out.print("Введите время прибытия для промежуточной
станции: ");

    String list6 = scan.nextLine();
    System.out.print("Введите время отправления для
промежуточной станции: ");
    String list7 = scan.nextLine();
    System.out.print("Введите количество пустых мест: ");
    int list8 = scan.nextInt();

    GregorianCalendar days_of_the_week7 = new
GregorianCalendar();

    System.out.print("Введите год: ");
    value = scan.nextInt();
    days_of_the_week7.set(Calendar.YEAR, value);
    System.out.print("Введите месяц: ");
    value = scan.nextInt();
    days_of_the_week7.set(Calendar.MONTH, value);
    System.out.print("Введите день: ");
    value = scan.nextInt();
    days_of_the_week7.set(Calendar.DAY_OF_MONTH, value);
    System.out.print("Введите часы: ");

```



```

        value = scan.nextInt();
        days_of_the_week7.set(Calendar.HOURLY_OF_DAY, value);
        System.out.print("Введите минуты: ");
        value = scan.nextInt();
        days_of_the_week7.set(Calendar.MINUTE, value);

        System.out.print("Введите общее количество мест: ");
        int total = scan.nextInt();
        System.out.print("Введите номер рейса: ");
        value = scan.nextInt();
        System.out.println("\nЭлемент добавлен.\n");
        StationClass newStation = new
StationClass(list,list1,list2,list3,list4,list5,list6,list7,list8);
        Station_list.addElementStation(newStation);
        RouteClass newRoute = new
RouteClass(total,days_of_the_week7,value);
        Route_list.addElement(newRoute);

        break;

    case 3:
        System.out.println();
        for (int i = 0; i < Route_list.getSize(); i++)
        {
            System.out.println(i+1 + ") ");
            Station_list.station[i].outputDataStation();
            Route_list.route[i].outputData();
            System.out.println();
        }

        System.out.print("Введите номер элемента для удаления: ");
        int position = scan.nextInt();
        if(position > Route_list.getSize() || position < 1 &&
position > Station_list.getSize() || position < 1)
        {
            System.out.println("Ошибка.Неправильный номер.");
            break;
        }
        Route_list.deleteElement(position);
        Station_list.deleteElementStation(position);
        System.out.println("\nЭлемент удален.\n");

        break;

    case 4:
        String address = new File("").getAbsolutePath(); //адрес
начальной директории
        File folder = new File(address); //создание
файла
        File[] arrayFiles = folder.listFiles(); //список
файлов в текущей директории
        String filename;
        //название файла для записи
        String currentDirectory = address;
        //адресс текущей директории
        String highestDir = folder.getName(); //название
максимально допустимой высокой директории

        boolean stop2 = false; //выход из цикла выбора директории
        int index = 0;
        int choise2 = 0;

        System.out.print("\nВведите имя XML файла: ");
        scan.nextLine();
        filename = scan.nextLine();

```

```

        if (filename.indexOf(".xml") == -1) {
            filename += ".xml";
        }

        while(!stop2)
        {
            index = 0;

            System.out.println("\nТекущий путь: " +
currentDirectory);

            System.out.println("Текущее имя файла XML: " +
filename);

            System.out.println("\nФайлы и каталоги в текущем
пути:");

            for (index = 0; index < arrayFiles.length; index++)
            {
                System.out.println(index+1 + ". " +
arrayFiles[index].toString().substring(currentDirectory.length()+1));
            }

            System.out.println();
            System.out.println("Какой номер из списка
выберете?");

            System.out.println("1. Записать XML-файл в текущую
директорию");

            System.out.println("2. Перейти на один уровень вверх
по папке");

            System.out.println("3. Войти в папку");
            System.out.println("4. Изменить имя файла XML");
            System.out.println("5. Выйти из сериализации");
            System.out.println("-----");

            System.out.print("Ваш выбор: ");
            choose2 = scan.nextInt();

            switch(choose2)
            {
                case 1:
                    stop2 = true;
                    break;

                case 2:
                    if(folder.getName().equals(highestDir))
                    {
                        System.out.print("\nВы не можете
подняться на один уровень вверх по папке");
                        break;
                    }
                    currentDirectory =
currentDirectory.substring(0, currentDirectory.indexOf(folder.getName())-1);
                    folder = new File(currentDirectory);
                    arrayFiles = folder.listFiles();//список
файлов в текущей директории

                    break;

                case 3:
                    boolean choose3 = false;

                    while(!choose3)
                    {
                        System.out.print("\nВыберите номер
каталога: ");

                        index = scan.nextInt();
                        if(index < 1 || index >
arrayFiles.length || !arrayFiles[index-1].isDirectory())

```

```

        {
            System.out.println("Это не
каталог. Попробуйте снова.");
        }
        else
        {
            currentDirectory =
            System.out.println("Новый текущий
            folder = new
            arrayFiles = folder.listFiles();
            choise3 = true;
        }
    }
    break;

    case 4:
        System.out.print("\nВведите имя XML файла: ");
        scan.nextLine();
        filename = scan.nextLine();

        if (filename.indexOf(".xml") == -1) {
            filename += ".xml";
        }
        break;

    case 5:
        System.out.println("Выход из раздела
сериализации");
        break;

    default:
        System.out.println("Ошибка. Неправильная
команда. Попробуйте еще раз");
        break;
}

}
address = currentDirectory;
System.out.println("\nФайл будет записан в текущий каталог:
" + address);

System.out.println("Имя файла XML: " + filename);
folder = new File(address);
File realFile = new File(folder, filename);
try {
    XMLEncoder encoder = new XMLEncoder(new
BufferedOutputStream(new FileOutputStream(realFile)));
    encoder.writeObject(Route_list.route);
    encoder.close();
} catch (Exception e) {
    System.out.println(e);
    break;
}
System.out.println("Сериализация прошла успешно.\n");

break;

    case 5:
        address = new File("").getAbsolutePath();    //адрес
начальной директории
        folder = new File(address);
        //создание файла

```


файлов в текущей директории

arrayFiles = folder.listFiles();//список

```
        break;

    case 3:
        boolean choose3 = false;

        while(!choose3)
        {
            System.out.print("\nВыберите номер
каталога: ");

            index = scan.nextInt();
            if(index < 1 || index >
arrayFiles.length || !arrayFiles[index-1].isDirectory())
            {
                System.out.println("Это не
каталог. Попробуйте снова.");
            }
            else
            {
                currentDirectory =
arrayFiles[index-1].toString();
                System.out.println("Новый текущий
каталог: " + currentDirectory);
                folder = new
File(currentDirectory);
                arrayFiles = folder.listFiles();
                choose3 = true;
            }
        }
        break;

    case 4:
        System.out.println("Выход из раздела
сериализации");

        stop2 = true;
        break;

    default:
        System.out.println("Ошибка. Неправильная
команда. Попробуйте еще раз");
        break;
}

}
address = currentDirectory;
System.out.println("Адрес файла XML: " + address + "\\\" +
arrayFiles[index-1].getName());
address = address + "\\\" + arrayFiles[index-1].getName();
folder = new File(address);
try {
    XMLDecoder decoder = new XMLDecoder(new
BufferedInputStream(new FileInputStream(folder)));
    Route_list.route =
(RouteClass[])decoder.readObject();
    decoder.close();
    Route_list.setSize(Route_list.route.length);
} catch (Exception e) {
    System.out.println(e);
    break;
}
System.out.println("Десериализация прошла успешно.\n");
```

```

        break;

    case 6:
        System.out.println("\nЗавершение программы");
        stop = true;
        break;

    default:
        System.out.println("Ошибка. Неправильная команда.
        Попробуйте еще раз.");
        break;
    }
}
}

```

Результат работы программы

```

Какой номер из списка выберете?
1. Вывести данные
2. Добавить элемент
3. Удалить элемент
4. Сериализовать данные
5. Десериализовать данные
6. Завершить программу
_ _ _ _ _
Ваш выбор: 1

1)
Названия станции: Лозовая-Харьков
Время отправления: 15:24
Время прибытия: 19:13
Начальная станция: Лозовая
Конечная станция: Харьков
Промежуточная станция: Панютино
Время отправления для промежуточных станций: 15:35
Время прибытия для промежуточных станций: 15:36
Количество пустых мест: 35
Общее количество мест: 150
День недели: Sun Nov 22 15:24:00 EET 2020
Номер рейса: 1

2)
Названия станции: Харьков-Гавриловка
Время отправления: 15:59
Время прибытия: 20:37
Начальная станция: Харьков
Конечная станция: Гавриловка
Промежуточная станция: Мерефа
Время отправления для промежуточных станций: 16:43
Время прибытия для промежуточных станций: 16:54
Количество пустых мест: 48
Общее количество мест: 200
День недели: Wed Nov 25 15:59:00 EET 2020
Номер рейса: 2

3)
Названия станции: Харьков-Изюм
Время отправления: 06:40
Время прибытия: 10:14
Начальная станция: Харьков
Конечная станция: Изюм
Промежуточная станция: Маяк
Время отправления для промежуточных станций: 09:24
Время прибытия для промежуточных станций: 09:25
Количество пустых мест: 72

```

Общее количество мест: 250
День недели: Mon Nov 30 06:40:00 EET 2020
Номер рейса: 3

4)

Названия станции: Красноград-Харьков
Время отправления: 16:44
Время прибытия: 19:34
Начальная станция: Красноград
Конечная станция: Харьков
Промежуточная станция: Власовка
Время отправления для промежуточных станций: 17:37
Время прибытия для промежуточных станций: 17:38
Количество пустых мест: 65
Общее количество мест: 300
День недели: Sun Nov 22 16:44:00 EET 2020
Номер рейса: 4

5)

Названия станции: Мерефа-Харьков
Время отправления: 06:58
Время прибытия: 07:45
Начальная станция: Мерефа
Конечная станция: Харьков
Промежуточная станция: Покотиловка
Время отправления для промежуточных станций: 07:25
Время прибытия для промежуточных станций: 07:26
Количество пустых мест: 37
Общее количество мест: 270
День недели: Wed Nov 25 06:58:00 EET 2020
Номер рейса: 5

6)

Названия станции: Славянск-Харьков
Время отправления: 06:27
Время прибытия: 10:51
Начальная станция: Славянск
Конечная станция: Харьков
Промежуточная станция: Гусаровка
Время отправления для промежуточных станций: 06:56
Время прибытия для промежуточных станций: 06:57
Количество пустых мест: 86
Общее количество мест: 120
День недели: Mon Nov 30 06:27:00 EET 2020
Номер рейса: 6

Какой номер из списка выберете?

1. Вывести данные
2. Добавить элемент
3. Удалить элемент
4. Сериализовать данные
5. Десериализовать данные
6. Завершить программу

Ваш выбор: 2

Введите номер маршрута: 7

Введите название станции: Харьков-Балаклея

Введите времени отправления: 23:12

Введите времени прибытия: 01:25

Введите начальную станцию: Харьков

Введите конечную станцию: Балаклея

Введите промежуточную станцию: Змиев

Введите время прибытия для промежуточной станции: 00:20

Введите время отправления для промежуточной станции: 00:21

Введите количество пустых мест: 76

Введите год: 2020

Введите месяц: 10

Введите день: 25

Введите часы: 23

Введите минуты: 12

Введите общее количество мест: 150

Введите номер рейса: 7

Элемент добавлен.

Какой номер из списка выберете?

1. Вывести данные
2. Добавить элемент
3. Удалить элемент
4. Сериализовать данные
5. Десериализовать данные
6. Завершить программу

Ваш выбор: 3

1)

Названия станции: Лозовая-Харьков

Время отправления: 15:24

Время прибытия: 19:13

Начальная станция: Лозовая

Конечная станция: Харьков

Промежуточная станция: Панютино

Время отправления для промежуточных станций: 15:35

Время прибытия для промежуточных станций: 15:36

Количество пустых мест: 35

Общее количество мест: 150

День недели: Sun Nov 22 15:24:00 EET 2020

Номер рейса: 1

2)

Названия станции: Харьков-Гавриловка

Время отправления: 15:59

Время прибытия: 20:37

Начальная станция: Харьков

Конечная станция: Гавриловка

Промежуточная станция: Мерефа

Время отправления для промежуточных станций: 16:43

Время прибытия для промежуточных станций: 16:54

Количество пустых мест: 48

Общее количество мест: 200

День недели: Wed Nov 25 15:59:00 EET 2020

Номер рейса: 2

3)

Названия станции: Харьков-Изюм

Время отправления: 06:40

Время прибытия: 10:14

Начальная станция: Харьков

Конечная станция: Изюм

Промежуточная станция: Маяк

Время отправления для промежуточных станций: 09:24

Время прибытия для промежуточных станций: 09:25

Количество пустых мест: 72

Общее количество мест: 250

День недели: Mon Nov 30 06:40:00 EET 2020

Номер рейса: 3

4)

Названия станции: Красноград-Харьков

Время отправления: 16:44

Время прибытия: 19:34

Начальная станция: Красноград

Конечная станция: Харьков

Промежуточная станция: Власовка

Время отправления для промежуточных станций: 17:37

Время прибытия для промежуточных станций: 17:38

Количество пустых мест: 65

Общее количество мест: 300

День недели: Sun Nov 22 16:44:00 EET 2020

Номер рейса: 4

5)

Названия станции: Мерефа-Харьков

Время отправления: 06:58

Время прибытия: 07:45

Начальная станция: Мерефа

Конечная станция: Харьков

Промежуточная станция: Покотиловка

Время отправления для промежуточных станций: 07:25

Время прибытия для промежуточных станций: 07:26

Количество пустых мест: 37

Общее количество мест: 270

День недели: Wed Nov 25 06:58:00 EET 2020

Номер рейса: 5

6)

Названия станции: Славянск-Харьков

Время отправления: 06:27

Время прибытия: 10:51

Начальная станция: Славянск

Конечная станция: Харьков

Промежуточная станция: Гусаровка

Время отправления для промежуточных станций: 06:56

Время прибытия для промежуточных станций: 06:57

Количество пустых мест: 86

Общее количество мест: 120

День недели: Mon Nov 30 06:27:00 EET 2020

Номер рейса: 6

7)

Названия станции: Харьков-Балаклея

Время отправления: 23:12

Время прибытия: 01:25

Начальная станция: Харьков

Конечная станция: Балаклея

Промежуточная станция: Змиев

Время отправления для промежуточных станций: 00:20

Время прибытия для промежуточных станций: 00:21

Количество пустых мест: 76

Общее количество мест: 150

День недели: Wed Nov 25 23:12:01 EET 2020

Номер рейса: 7

Введите номер элемента для удаления: 7

Элемент удален.

Какой номер из списка выберете?

1. Вывести данные
2. Добавить элемент
3. Удалить элемент
4. Сериализовать данные
5. Десериализовать данные
6. Завершить программу

Ваш выбор: 4

Введите имя XML файла: MyXML.xml

Текущий путь: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav

Текущее имя файла XML: MyXML.xml

Файлы и каталоги в текущем пути:

1. .classpath
2. .project
3. .settings
4. bilyi03.jar
5. bin
6. KapelkaTest.jar
7. src

Какой номер из списка выберете?

1. Записать XML-файл в текущую директорию
2. Перейти на один уровень вверх по папке
3. Войти в папку
4. Изменить имя файла XML
5. Выйти из сериализации

Ваш выбор: 3

Выберите номер каталога: 4

Это не каталог. Попробуйте снова.

Выберите номер каталога: 7

Новый текущий каталог: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav\src

Текущий путь: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav\src

Текущее имя файла XML: MyXML.xml

Файлы и каталоги в текущем пути:

1. ua

Какой номер из списка выберете?

1. Записать XML-файл в текущую директорию
2. Перейти на один уровень вверх по папке
3. Войти в папку
4. Изменить имя файла XML
5. Выйти из сериализации

Ваш выбор: 4

Введите имя XML файла: MyFile.xml

Текущий путь: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav\src

Текущее имя файла XML: MyFile.xml

Файлы и каталоги в текущем пути:

1. ua

Какой номер из списка выберете?

1. Записать XML-файл в текущую директорию
2. Перейти на один уровень вверх по папке
3. Войти в папку
4. Изменить имя файла XML
5. Выйти из сериализации

Ваш выбор: 1

Файл будет записан в текущий каталог: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav\src

Имя файла XML: MyFile.xml

Сериализация прошла успешно.

Какой номер из списка выберете?

1. Вывести данные
2. Добавить элемент
3. Удалить элемент
4. Сериализовать данные
5. Десериализовать данные
6. Завершить программу

Ваш выбор: 5

Текущий путь: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav

Файлы и каталоги в текущем пути:

1. .classpath
2. .project
3. .settings
4. bilyi03.jar
5. bin
6. KapelkaTest.jar
7. src

Какой номер из списка выберете?

1. Прочитать XML-файл в текущем каталоге
2. Перейти на один уровень вверх по папке
3. Войти в папку
4. Выйти из сериализации

=====

Ваш выбор: 1

Введите ID файла: 4

Это не файл .XML.

Текущий путь: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav

Файлы и каталоги в текущем пути:

1. .classpath
2. .project
3. .settings
4. bilyi03.jar
5. bin
6. KapelkaTest.jar
7. src

Какой номер из списка выберете?

1. Прочитать XML-файл в текущем каталоге
2. Перейти на один уровень вверх по папке
3. Войти в папку
4. Выйти из сериализации

=====

Ваш выбор: 1

```

Введите ID файла: 6
Это не файл .XML.

Текущий путь: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav
Файлы и каталоги в текущем пути:
1. .classpath
2. .project
3. .settings
4. bilyi03.jar
5. bin
6. KapelkaTest.jar
7. src

Какой номер из списка выберете?
1. Прочитать XML-файл в текущем каталоге
2. Перейти на один уровень вверх по папке
3. Войти в папку
4. Выйти из сериализации
=====
Ваш выбор: 3

Выберите номер каталога: 7
Новый текущий каталог: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav\src

Текущий путь: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav\src
Файлы и каталоги в текущем пути:
1. MyFile.xml
2. ua

Какой номер из списка выберете?
1. Прочитать XML-файл в текущем каталоге
2. Перейти на один уровень вверх по папке
3. Войти в папку
4. Выйти из сериализации
=====
Ваш выбор: 1

Введите ID файла: 1
Адрес файла XML: C:\Users\Admin\eclipse-workspace\Kapelka-Yaroslav\src\MyFile.xml
Десериализация прошла успешно.

```

```

Какой номер из списка выберете?
1. Вывести данные
2. Добавить элемент
3. Удалить элемент
4. Сериализовать данные
5. Десериализовать данные
6. Завершить программу
_ _ _ _ _
Ваш выбор: 6

Завершение программы

```

Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з Java SE.

Програма протестована, виконується без помилок.