

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, публічний клас RouteClass, у полях якого є час початку події, тривалість, адреса події, імена людей, опис події, 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 Main08
{
    public static void main(String[] args)
    {
        RouteClassList Route_list = new RouteClassList();
        String[] listOfStation_name1 = {"Лозовая-Харьков", "Харьков-Гавриловка",
"Харьков-Изюм"};
        String[] listOfStation_name2 = {"Красноград-Харьков", "Мерефа-Харьков",
"Славянск-Харьков"};
        String[] listOfdeparture_time1 = {"15:24", "15:59", "06:40"};
        String[] listOfdeparture_time2 = {"16:44", "06:58", "06:27"};
        String[] listOfarrival_time1 = {"19:13", "20:37", "10:14"};
        String[] listOfarrival_time2 = {"19:34", "7:45", "10:51"};
        String[] listOfstarting_station1 = {"Лозовая", "Харьков", "Харьков"};
        String[] listOfstarting_station2 = {"Красноград", "Мерефа", "Славянск"};
        String[] listOfterminal_station1 = {"Харьков", "Гавриловка", "Изюм"};

        String[] listOfterminal_station2 = {"Харьков", "Харьков", "Харьков"};
        String[] listOfintermediate_stations1 = {"Панютино", "Мерефа", "Маяк"};
        String[] listOfintermediate_stations2 = {"Власовка", "Покотиловка",
"Гусаровка"};
        String[] listOfdeparture_time_of_intermediate_stations1 = {"15:35",
"16:43", "09:24"};
        String[] listOfdeparture_time_of_intermediate_stations2 = {"17:37",
"07:25", "6:56"};
        String[] listOfarrival_time_of_intermediate_stations1 = {"15:36",
"16:54", "09:25"};
        String[] listOfarrival_time_of_intermediate_stations2 = {"17:38",
"07:26", "6:57"};
        int[] listOfnumber_of_free_seats1 = { 35, 48, 72};
        int[] listOfnumber_of_free_seats2 = { 65, 37, 86};

        ArrayList<GregorianCalendar> List_days_of_the_week1 = new
ArrayList<GregorianCalendar>();
    }
}

```

```

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);
    List_days_of_the_week1.add(days_of_the_week1);
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);
    List_days_of_the_week1.add(days_of_the_week2);
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);
    List_days_of_the_week1.add(days_of_the_week3);

    RouteClass routeclass1 = new RouteClass(listOfStation_name1,
listOfdeparture_time1, listOfarrival_time1, listOfstarting_station1,
listOfterminal_station1, listOfintermediate_stations1,
listOfdeparture_time_of_intermediate_stations1,
listOfarrival_time_of_intermediate_stations1, listOfnumber_of_free_seats1, 150,
List_days_of_the_week1, 1);
    Route_list.addElement(routeclass1);

    ArrayList<GregorianCalendar> List_days_of_the_week2 = new
ArrayList<GregorianCalendar>();
    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);
    List_days_of_the_week2.add(days_of_the_week4);
    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);
    List_days_of_the_week2.add(days_of_the_week5);
    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);
    List_days_of_the_week2.add(days_of_the_week6);

    RouteClass routeclass2 = new RouteClass(listOfStation_name2,
listOfdeparture_time2, listOfarrival_time2, listOfstarting_station2,
listOfterminal_station2, listOfintermediate_stations2,
listOfdeparture_time_of_intermediate_stations2,
listOfarrival_time_of_intermediate_stations2, listOfnumber_of_free_seats2, 200,
List_days_of_the_week2, 2);
    Route_list.addElement(routeclass2);

    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 + " ");
        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.println("Ошибка. Неверный размер
списка.");
        break;
    }
    String[] list = new String[value];
    System.out.println("Введите список названий станций: ");
    scan.nextLine();
    for (int i = 0; i < value; i++)
    {
        System.out.print(i+1 + ". ");
        list[i] = scan.nextLine();
    }
    String[] list1 = new String[value];
    System.out.println("Введите список времени отправления: ");
    for (int i = 0; i < value; i++)
    {
        System.out.print(i+1 + ". ");
        list1[i] = scan.nextLine();
    }
    String[] list2 = new String[value];
    System.out.println("Введите список времени прибытия: ");
    for (int i = 0; i < value; i++)
    {
        System.out.print(i+1 + ". ");
        list2[i] = scan.nextLine();
    }
    String[] list3 = new String[value];
    System.out.println("Введите список начальных станций: ");
    for (int i = 0; i < value; i++)
    {
        System.out.print(i+1 + ". ");
        list3[i] = scan.nextLine();
    }
    String[] list4 = new String[value];
    System.out.println("Введите список конечных станций: ");
    for (int i = 0; i < value; i++)
    {
        System.out.print(i+1 + ". ");
        list4[i] = scan.nextLine();
    }
    String[] list5 = new String[value];

```

```

");

        System.out.println("Введите список промежуточных станций:");

        for (int i = 0; i < value; i++)
        {
            System.out.print(i+1 + ". ");
            list5[i] = scan.nextLine();
        }
        String[] list6 = new String[value];
        System.out.println("Введите список времени отправления для
промежуточных станций: ");
        for (int i = 0; i < value; i++)
        {
            System.out.print(i+1 + ". ");
            list6[i] = scan.nextLine();
        }
        String[] list7 = new String[value];
        System.out.println("Введите список времени прибытия для
промежуточных станций: ");
        for (int i = 0; i < value; i++)
        {
            System.out.print(i+1 + ". ");
            list7[i] = scan.nextLine();
        }
        int[] list8 = new int[value];
        System.out.println("Введите список количества пустых мест:
");
        for (int i = 0; i < value; i++)
        {
            System.out.print(i+1 + ". ");
            list8[i] = scan.nextInt();
        }

        ArrayList<GregorianCalendar> List_days_of_the_week3 = new
ArrayList<GregorianCalendar>();
        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.HOUR_OF_DAY, value);
        System.out.print("Введите минуты: ");
        value = scan.nextInt();
        days_of_the_week7.set(Calendar.MINUTE, value);
        List_days_of_the_week3.add(days_of_the_week7);

        GregorianCalendar days_of_the_week8 = new
GregorianCalendar();

        System.out.print("Введите год: ");
        value = scan.nextInt();
        days_of_the_week8.set(Calendar.YEAR, value);
        System.out.print("Введите месяц: ");
        value = scan.nextInt();
        days_of_the_week8.set(Calendar.MONTH, value);
        System.out.print("Введите день: ");
        value = scan.nextInt();
        days_of_the_week8.set(Calendar.DAY_OF_MONTH, value);
        System.out.print("Введите часы: ");
        value = scan.nextInt();

```

```

        days_of_the_week8.set(Calendar.HOURLY_OF_DAY, value);
        System.out.print("Введите минуты: ");
        value = scan.nextInt();
        days_of_the_week8.set(Calendar.MINUTE, value);
        List_days_of_the_week3.add(days_of_the_week8);

        GregorianCalendar days_of_the_week9 = new

        System.out.print("Введите год: ");
        value = scan.nextInt();
        days_of_the_week9.set(Calendar.YEAR, value);
        System.out.print("Введите месяц: ");
        value = scan.nextInt();
        days_of_the_week9.set(Calendar.MONTH, value);
        System.out.print("Введите день: ");
        value = scan.nextInt();
        days_of_the_week9.set(Calendar.DAY_OF_MONTH, value);
        System.out.print("Введите часы: ");
        value = scan.nextInt();
        days_of_the_week9.set(Calendar.HOURLY_OF_DAY, value);
        System.out.print("Введите минуты: ");
        value = scan.nextInt();
        days_of_the_week9.set(Calendar.MINUTE, value);
        List_days_of_the_week3.add(days_of_the_week9);

        System.out.print("Введите количество мест: ");
        scan.nextLine();
        int total = scan.nextInt();
        System.out.print("Введите номер рейса: ");
        value = scan.nextInt();
        System.out.println("\nЭлемент добавлен.\n");

        RouteClass newRoute = new
RouteClass(list,list1,list2,list3,list4,list5,list6,list7,list8,total,List_days_of_th
e_week3,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 + " ");
            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)
        {
            System.out.println("Ошибка.Неправильный номер.");
            break;
        }
        Route_list.deleteElement(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: ");
        choise2 = scan.nextInt();

        switch(choise2)
        {
            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;
                }
            case 3:
                currentDirectory = currentDirectory + folder.getName() +
                "\\";
                break;
            case 4:
                filename = scan.nextLine();
                break;
            case 5:
                stop2 = true;
                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 choise3 = false;

while(!choise3)
{
    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
        arrayFiles = folder.listFiles();            //список
файлов в текущей директории
        currentDirectory = address;
        //адрес текущей директории
        highestDir = folder.getName();
        //название макимально допустимой высокой директории

        stop2 = false;    //выход из цыула выбора директории
        index = 0;
        choise2 = 0;

        while(!stop2)
        {
            index = 0;

            System.out.println("\nТекущий путь: " +
currentDirectory);
            System.out.println("Файлы 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. Прочитать XML-файл в текущем
каталоге");
            System.out.println("2. Перейти на один уровень вверх
по папке");
            System.out.println("3. Войти в папку");
            System.out.println("4. Выйти из сериализации");

            System.out.println("=====");
            System.out.print("Ваш выбор: ");
            choise2 = scan.nextInt();

            switch(choise2)
            {
                case 1:
                    System.out.print("\nВведите ID файла: ");
                    index = scan.nextInt();
                    if(arrayFiles[index-
1].getName().indexOf(".xml")==-1 || arrayFiles[index-1].isDirectory())
                    {

```

```

.XML.");

        }

        stop2 = true;
        break;

    case 2:
        if(folder.getName().equals(highestDir))
        {
            System.out.println("Вы не можете
подняться на один уровень в 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();
                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 програми

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)

Список названий станций:

1. Лозовая-Харьков
2. Харьков-Гавриловка
3. Харьков-Изюм

Список времени отправления:

1. 15:24
2. 15:59
3. 06:40

Список времени прибытия:

1. 19:13
2. 20:37
3. 10:14

Список времени отправления для промежуточных станций:

1. 15:35
2. 16:43
3. 09:24

Список времени прибытия для промежуточных станций:

1. 15:36
2. 16:54
3. 09:25

Список количества пустых мест:

1. 35
2. 48
3. 72

Список начальных станций:

1. Лозовая
2. Харьков
3. Харьков

Список конечных станций:

1. Харьков
2. Гавриловка
3. Изюм

Список промежуточных станций:

1. Панютино
2. Мерефа
3. Маяк

Общее количество мест: 150

Список дней недели:  
Sun Nov 22 15:24:00 EET 2020  
Wed Nov 25 15:59:00 EET 2020  
Mon Nov 30 06:40:00 EET 2020  
Номер рейса: 1

2)

Список названий станций:

1. Красноград-Харьков
2. Мерефа-Харьков
3. Славянск-Харьков

Список времени отправления:

1. 16:44
2. 06:58
3. 06:27

Список времени прибытия:

1. 19:34
2. 7:45
3. 10:51

Список времени отправления для промежуточных станций:

1. 17:37
2. 07:25
3. 6:56

Список времени прибытия для промежуточных станций:

1. 17:38
2. 07:26
3. 6:57

Список количества пустых мест:

1. 65
2. 37
3. 86

Список начальных станций:

1. Красноград
2. Мерефа
3. Славянск

Список конечных станций:

1. Харьков
2. Харьков
3. Харьков

Список промежуточных станций:

1. Власовка
2. Покотиловка
3. Гусаровка

Общее количество мест: 200

Список дней недели:

Sun Nov 22 16:44:00 EET 2020  
Wed Nov 25 06:58:00 EET 2020  
Mon Nov 30 06:27:00 EET 2020

Номер рейса: 2

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: 3

Введите список названий станций:

1. Харьков-Балаклея
2. Золочев-Харьков
3. Харьков-Люботин

Введите список времени отправления:

1. 23:12
2. 11:35
3. 07:22

Введите список времени прибытия:

1. 01:25
2. 13:23
3. 08:25

Введите список начальных станций:

1. Харьков
2. Золочев
3. Харьков

Введите список конечных станций:

1. Балаклея
2. Харьков
3. Люботин

Введите список промежуточных станций:

1. Змиев
2. Рыжов
3. Водяная

Введите список времени отправления для промежуточных станций:

1. 00:20
2. 12:48
3. 08:19

Введите список времени прибытия для промежуточных станций:

1. 00:21
2. 12:49
3. 08:20

Введите список количества пустых мест:

1. 58
2. 75
3. 64

Введите год: 2020

Введите месяц: 10

Введите день: 25

Введите часы: 23

Введите минуты: 12

Введите год: 2020
Введите месяц: 10
Введите день: 30
Введите часы: 11
Введите минуты: 35
Введите год: 2020
Введите месяц: 11
Введите день: 04
Введите часы: 7
Введите минуты: 22
Введите количество мест: 300
Введите номер рейса: 3

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)

Список названий станций:

1. Лозовая-Харьков
2. Харьков-Гавриловка
3. Харьков-Изюм

Список времени отправления:

1. 15:24
2. 15:59
3. 06:40

Список времени прибытия:

1. 19:13
2. 20:37
3. 10:14

Список времени отправления для промежуточных станций:

1. 15:35
2. 16:43
3. 09:24

Список времени прибытия для промежуточных станций:

1. 15:36
2. 16:54
3. 09:25

Список количества пустых мест:

1. 35
2. 48
3. 72

Список начальных станций:

1. Лозовая
2. Харьков
3. Харьков

Список конечных станций:

1. Харьков
2. Гавриловка
3. Изюм

Список промежуточных станций:

1. Панютино
2. Мерефа
3. Маяк

Общее количество мест: 150
Список дней недели:
Sun Nov 22 15:24:00 EET 2020
Wed Nov 25 15:59:00 EET 2020
Mon Nov 30 06:40:00 EET 2020
Номер рейса: 1

2)

Список названий станций:

1. Красноград-Харьков
2. Мерефа-Харьков
3. Славянск-Харьков

Список времени отправления:

1. 16:44
2. 06:58
3. 06:27

Список времени прибытия:

1. 19:34
2. 7:45
3. 10:51

Список времени отправления для промежуточных станций:

1. 17:37
2. 07:25
3. 6:56

Список времени прибытия для промежуточных станций:

1. 17:38
2. 07:26
3. 6:57

Список количества пустых мест:

1. 65
2. 37
3. 86

Список начальных станций:

1. Красноград
2. Мерефа
3. Славянск

Список конечных станций:

1. Харьков
2. Харьков
3. Харьков

Список промежуточных станций:

1. Власовка
2. Покотиловка
3. Гусаровка

Общее количество мест: 200

Список дней недели:

Sun Nov 22 16:44:00 EET 2020
Wed Nov 25 06:58:00 EET 2020

Mon Nov 30 06:27:00 EET 2020

Номер рейса: 2

3)

Список названий станций:

1. Харьков-Балаклея
2. Золочев-Харьков
3. Харьков-Люботин

Список времени отправления:

1. 23:12
2. 11:35
3. 07:22

Список времени прибытия:

1. 01:25
2. 13:23
3. 08:25

Список времени отправления для промежуточных станций:

1. 00:20
2. 12:48
3. 08:19

Список времени прибытия для промежуточных станций:

1. 00:21
2. 12:49
3. 08:20

Список количества пустых мест:

1. 58
2. 64
3. 73

Список начальных станций:

1. Харьков
2. Золочев
3. Харьков

Список конечных станций:

1. Балаклея
2. Харьков
3. Люботин

Список промежуточных станций:

1. Змиев
2. Рыжов
3. Водяная

Общее количество мест: 300

Список дней недели:

Wed Nov 25 23:12:19 EET 2020

Mon Nov 30 11:35:27 EET 2020

Fri Dec 04 07:22:37 EET 2020

Номер рейса: 3

Введите номер элемента для удаления: 3

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.