

## 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: назва маршруту, 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, час відправлення, кількість вільних місць, статус станції"; 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 String name_route;
    private int total_number_of_seats;
    private GregorianCalendar days_of_the_week;
    private int flight_number;
    private int size = 0;
    StationClass station[] = new StationClass[size];
    RouteClass route;

    public int getSize()
    {
        return size;
    }

    public void setSize(int size)
    {
        this.size = size;
    }
    public StationClass[] getStation()
    {
        return station;
    }
    public void setStation(StationClass[] station)
    {
        this.station = station;
    }
    public String getName_route()
    {
        return name_route;
    }
    public void setName_route(String name_route)

```

```

{
    this.name_route = name_route;
}
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(String name, StationClass[] station, int
total_number, GregorianCalendar days, int flight)
{
    this.name_route = name;
    this.station = station;
    total_number_of_seats = total_number;
    this.days_of_the_week = days;
    flight_number = flight;
}
public void outputData()
{
    System.out.println("Название маршрута: " + name_route);
    System.out.println("Информация о станциях, присутствующих в данном
маршруте: ");
    for (int i = 0; i < station.length; i++)
    {
        System.out.println("Информация о станции номер: " + (i + 1) + ".
");
        station[i].outputDataStation();
    }
    System.out.println("Общее количество мест: " + total_number_of_seats);
    System.out.println("День недели, когда доступен маршрут: " +
days_of_the_week.getTime());
    System.out.println("Номер рейса: " + flight_number);
}
}

```

```

public class StationClass
{
    private String station_name;
    private String departure_time;
    private String arrival_time;
    private int number_of_free_seats;
    private String status_station;

    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 StationClass() {};

    public StationClass(String name, String time,String time1, int number, String
status)
    {
        this.station_name = name;
        this.arrival_time = time;
        this.departure_time = time1;
        this.number_of_free_seats = number;
        this.status_station = status;
    }
    public void outputDataStation()
    {
        System.out.println("Названия станции: " + station_name);
        System.out.println("Время прибытия на станцию: " + arrival_time);
        System.out.println("Время отправления с станции: " + departure_time);
        System.out.println("Количество пустых мест: " + number_of_free_seats);
        System.out.println("Статус станции: " + status_station);
    }
}

```

```

    }
}

public class Main08
{
    public static void main(String[] args)
    {
        RouteClassList Route_list = new RouteClassList();
        String Name_route1 = "Лозовая-Харьков";
        String Name_route2 = "Харьков-Изюм";
        String Station_name1 = "Лозовая";
        String Station_name2 = "Панютино";
        String Station_name3 = "Харьков";
        String Station_name4 = "Харьков";
        String Station_name5 = "Маяк";
        String Station_name6 = "Изюм";
        String arrival_time1 = "15:20";
        String arrival_time2 = "15:35";
        String arrival_time3 = "19:13";
        String arrival_time4 = "11:55";
        String arrival_time5 = "14:52";
        String arrival_time6 = "15:41";
        String departure_time1 = "15:24";
        String departure_time2 = "15:36";
        String departure_time3 = "Времени отправления нету, так как это конечная
станция.";
        String departure_time4 = "12:00";
        String departure_time5 = "14:53";
        String departure_time6 = "Времени отправления нету, так как это конечная
станция.";
        int number_of_free_seats1 = 55;
        int number_of_free_seats2 = 78;
        int number_of_free_seats3 = 63;
        int number_of_free_seats4 = 87;
        int number_of_free_seats5 = 61;
        int number_of_free_seats6 = 75;
        String Status_station1 = "Начальная";
        String Status_station2 = "Промежуточная";
        String Status_station3 = "Конечная";
        String Status_station4 = "Начальная";
        String Status_station5 = "Промежуточная";
        String Status_station6 = "Конечная";
        GregorianCalendar days_of_the_week1 = new GregorianCalendar(2020, 11,
7);
        days_of_the_week1.set(Calendar.HOUR_OF_DAY, 00);
        days_of_the_week1.set(Calendar.MINUTE, 00);
        days_of_the_week1.set(Calendar.SECOND, 00);
        GregorianCalendar days_of_the_week2 = new GregorianCalendar(2020, 11,
8);
        days_of_the_week2.set(Calendar.HOUR_OF_DAY, 00);
        days_of_the_week2.set(Calendar.MINUTE, 00);
        days_of_the_week2.set(Calendar.SECOND, 00);

        RouteClass Station_list = new RouteClass();
        StationClass Station_list_mas[] = new StationClass[3];
        Station_list_mas[0] = new StationClass(Station_name1, arrival_time1,
departure_time1, number_of_free_seats1, Status_station1);
        Station_list_mas[1] = new StationClass(Station_name2, arrival_time2,
departure_time2, number_of_free_seats2, Status_station2);
        Station_list_mas[2] = new StationClass(Station_name3, arrival_time3,
departure_time3, number_of_free_seats3, Status_station3);
        StationClass Station_list1_mas[] = new StationClass[3];
        Station_list1_mas[0] = new StationClass(Station_name4, arrival_time4,
departure_time4, number_of_free_seats4, Status_station4);
    }
}

```

```

        Station_list1_mas[1] = new StationClass(Station_name5, arrival_time5,
departure_time5, number_of_free_seats5, Status_station5);
        Station_list1_mas[2] = new StationClass(Station_name6, arrival_time6,
departure_time6, number_of_free_seats6, Status_station6);

        RouteClass routeclass1 = new RouteClass(Name_route1, Station_list_mas,
150, days_of_the_week1, 1);
        RouteClass routeclass2 = new RouteClass(Name_route2, Station_list1_mas,
200, days_of_the_week2, 2);
        Route_list.addElement(routeclass1);
        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: ");

            choose = scan.nextInt();

            switch (choose)
            {
                case 1:
                    System.out.println();
                    for(RouteClass route : Route_list.route)
                    {
                        route.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 маршрута: ");
                    String list = scan.nextLine();
                    System.out.print("Введите название станции: ");
                    String list1 = scan.nextLine();
                    System.out.print("Введите время прибытия на станцию: ");
                    String list2 = scan.nextLine();
                    System.out.print("Введите время отправления с станции: ");
                    String list3 = scan.nextLine();
                    System.out.print("Введите количество пустых мест: ");
                    int list4 = scan.nextInt();
                    System.out.print("Введите статус станиции: ");
                    String list5 = scan.nextLine();

                    scan.nextLine();
                    System.out.print("Введите название станции: ");
                    String list6 = scan.nextLine();

```

```

System.out.print("Введите время прибытия на станцию: ");
String list7 = scan.nextLine();
System.out.print("Введите время отправления с станции: ");
String list8 = scan.nextLine();
System.out.print("Введите количество пустых мест: ");
int list9 = scan.nextInt();
System.out.print("Введите статус станции: ");
String list10 = scan.nextLine();

scan.nextLine();
System.out.print("Введите название станции: ");
String list11 = scan.nextLine();
System.out.print("Введите время прибытия на станцию: ");
String list12 = scan.nextLine();
System.out.print("Введите время отправления с станции: ");
String list13 = scan.nextLine();
System.out.print("Введите количество пустых мест: ");
int list14 = scan.nextInt();
System.out.print("Введите статус станции: ");
String list15 = scan.nextLine();

scan.nextLine();
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);

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[3];
newStation[0] = new
StationClass(list1,list2,list3,list4,list5);
newStation[1] = new
StationClass(list6,list7,list8,list9,list10);
newStation[2] = new
StationClass(list11,list12,list13,list14,list15);

RouteClass newRoute = new
RouteClass(list,newStation,total,days_of_the_week7,value);
Route_list.addElement(newRoute);

break;

case 3:
System.out.println();
for(RouteClass route : Route_list.route)
{
route.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;
//адрес 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. Измените 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;
            }
            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 =
arrayFiles[index-1].toString();
                    System.out.println("Новый текущий
каталог: " + currentDirectory);
                    folder = new
File(currentDirectory);
                    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;
    //адресс текущей директории
    highestDir = folder.getName();
    //название макчимально 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");

```

```

по папке");
System.out.println("2. Перейти на один уровень вверх");
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())
    {
        System.out.println("Это не файл
.XML.");

        break;
    }

    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 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 =
arrayFiles[index-1].toString();
            System.out.println("Новый текущий
каталог: " + currentDirectory);

            folder = new
File(currentDirectory);

            arrayFiles = folder.listFiles();

            choise3 = 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: 2

Введите номер маршрута: 3

Введите название маршрута: Харьков-Балаклея

Введите название станции: Харьков

Введите время прибытия на станцию: 23:10

Введите время отправления с станции: 23:12

Введите количество пустых мест: 78

Введите статус станиции: Начальная

Введите название станции: Змиев

Введите время прибытия на станцию: 00:20

Введите время отправления с станции: 00:21

Введите количество пустых мест: 85

Введите статус станиции: Промежуточная

Введите название станции: Балаклея

Введите время прибытия на станцию: 01:25

Введите время отправления с станции: Времени отправления нету, так как это конечная станция.

Введите количество пустых мест: 60

Введите статус станиции: Конечная

Введите год: 2020

Введите месяц: 10

Введите день: 25

Введите часы: 00

Введите минуты: 00

Введите общее 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

Название маршрута: Лозовая-Харьков

Информация о станциях, присутствующих в данном маршруте:

Информация о станции номер: 1.

Названия станции: Лозовая

Время прибытия на станцию: 15:20

Время отправления с станции: 15:24

Количество пустых мест: 55

Статус станции: Начальная

Информация о станции номер: 2.

Названия станции: Панютино

Время прибытия на станцию: 15:35

Время отправления с станции: 15:36

Количество пустых мест: 78

Статус станции: Промежуточная

Информация о станции номер: 3.

Названия станции: Харьков

Время прибытия на станцию: 19:13

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

Количество пустых мест: 63

Статус станции: Конечная

Общее количество мест: 150

День недели, когда доступен маршрут: Mon Dec 07 00:00:00 EET 2020

Номер рейса: 1

Название маршрута: Харьков-Изюм

Информация о станциях, присутствующих в данном маршруте:

Информация о станции номер: 1.

Названия станции: Харьков

Время прибытия на станцию: 11:55

Время отправления с станции: 12:00

Количество пустых мест: 87

Статус станции: Начальная

Информация о станции номер: 2.

Названия станции: Маяк

Время прибытия на станцию: 14:52

Время отправления с станции: 14:53

Количество пустых мест: 61

Статус станции: Промежуточная

Информация о станции номер: 3.

Названия станции: Изюм

Время прибытия на станцию: 15:41  
Время отправления с станции: Времени отправления нету, так как это конечная станция.  
Количество пустых мест: 75  
Статус станции: Конечная  
Общее количество мест: 200  
День недели, когда доступен маршрут: Tue Dec 08 00:00:00 EET 2020  
Номер рейса: 2

Название маршрута: Харьков-Балаклея  
Информация о станциях, присутствующих в данном маршруте:  
Информация о станции номер: 1.  
Названия станции: Харьков  
Время прибытия на станцию: 23:10  
Время отправления с станции: 23:12  
Количество пустых мест: 78  
Статус станции:  
Информация о станции номер: 2.  
Названия станции: Змиев  
Время прибытия на станцию: 00:20  
Время отправления с станции: 00:21  
Количество пустых мест: 85  
Статус станции:  
Информация о станции номер: 3.  
Названия станции: Балаклея  
Время прибытия на станцию: 01:25  
Время отправления с станции: Времени отправления нету, так как это конечная станция.  
Количество пустых мест: 60  
Статус станции:  
Общее количество мест: 300  
День недели, когда доступен маршрут: Wed Nov 25 00:00:46 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.