**Звіт**

**Лабораторна работа 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. **Розробник**: Капелька Ярослав Іванович, КІТ-119а, варіант №9.
6. **ОПИС ПРОГРАМИ**
   1. **Засоби ООП**: клас, метод класу, поле класу.
   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, класс 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.
   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** choise;

**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 названии 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.***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(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 текущею 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(); //список файлов в текущей директории

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");

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())

{

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(); //список файлов в текущей директории

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**;

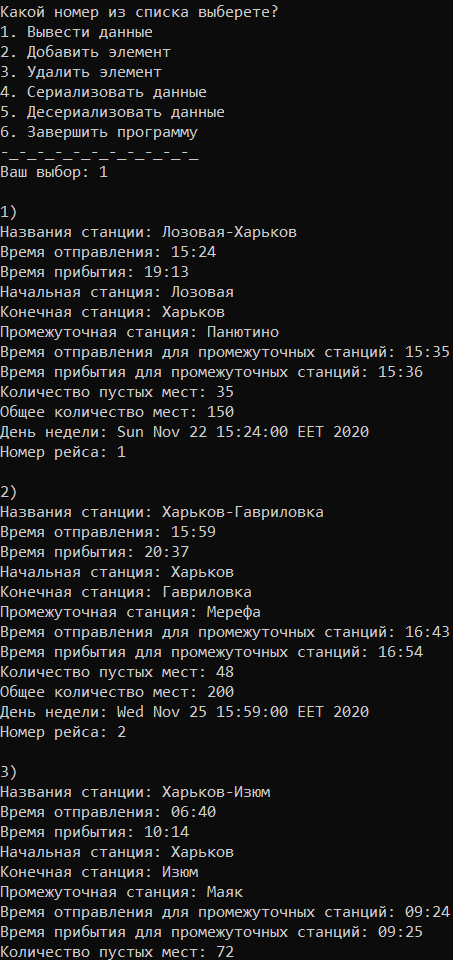
}

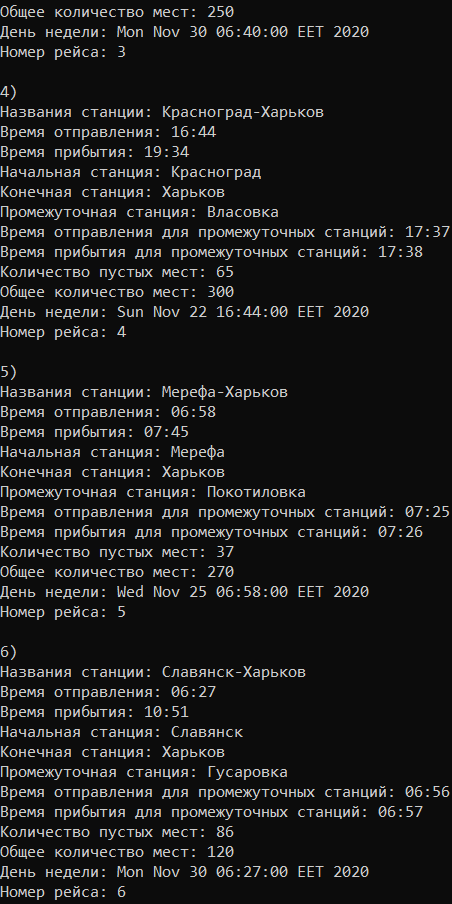
}

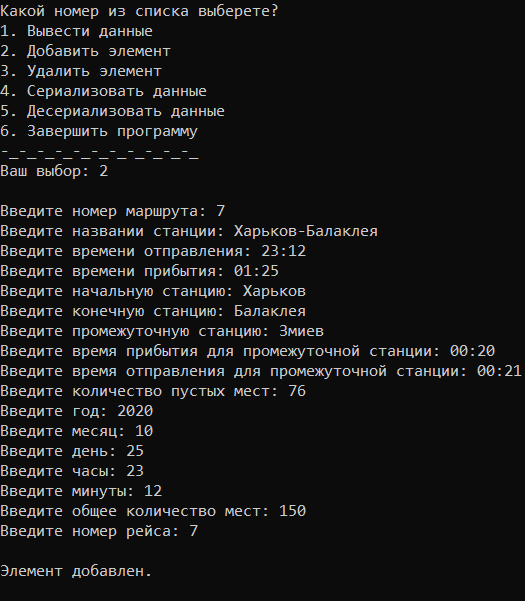
}

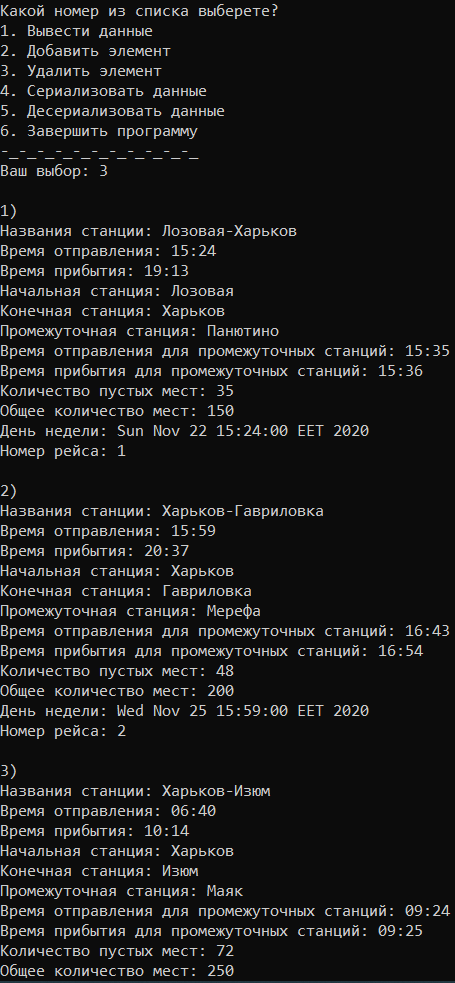
}

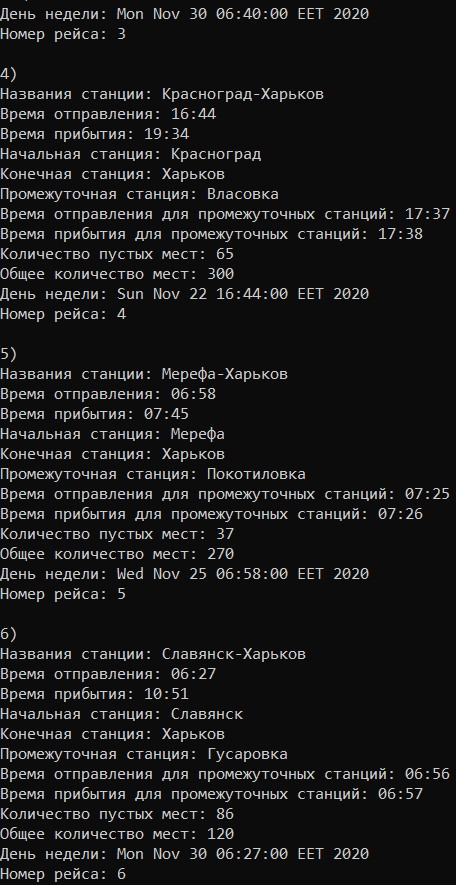
**Результат роботи програми**

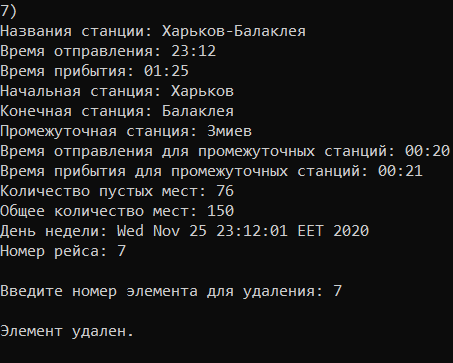
****

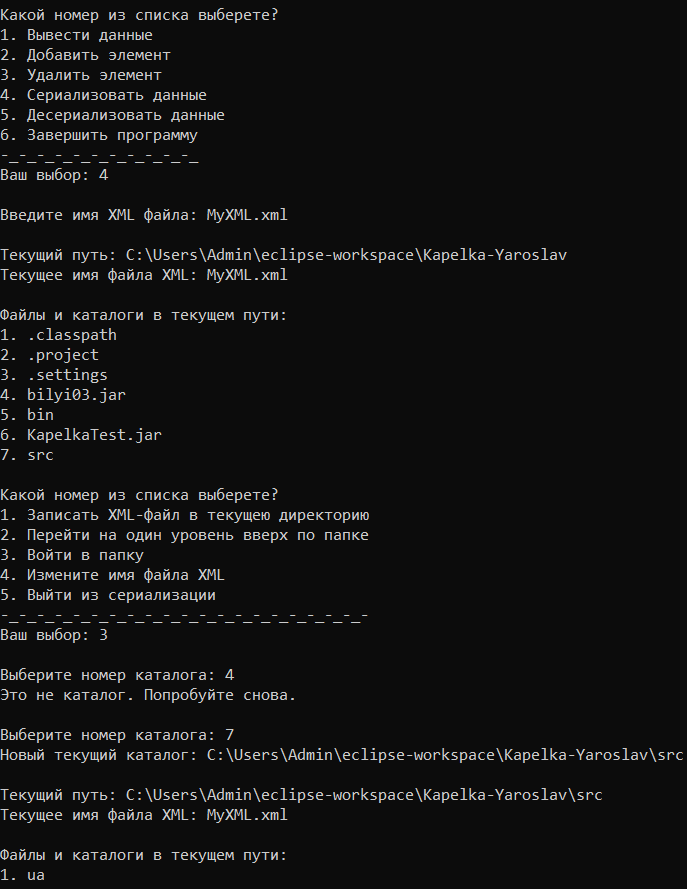
****

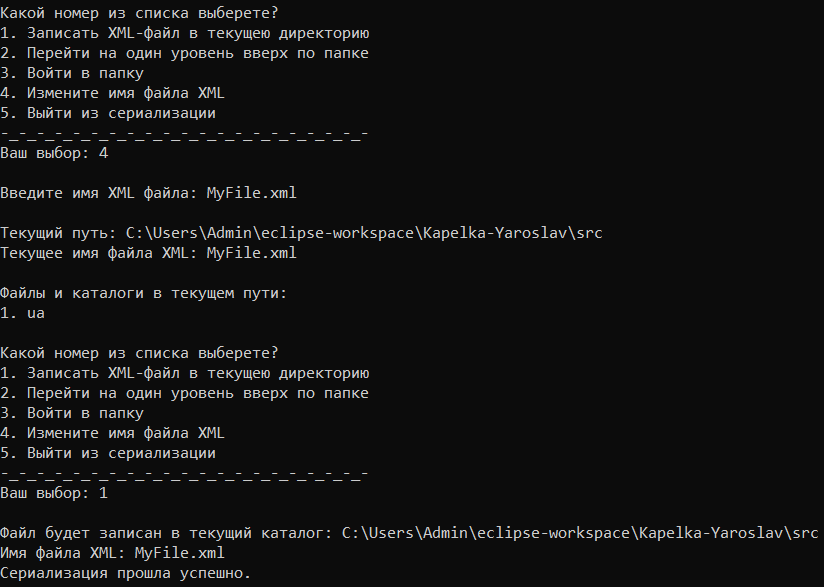


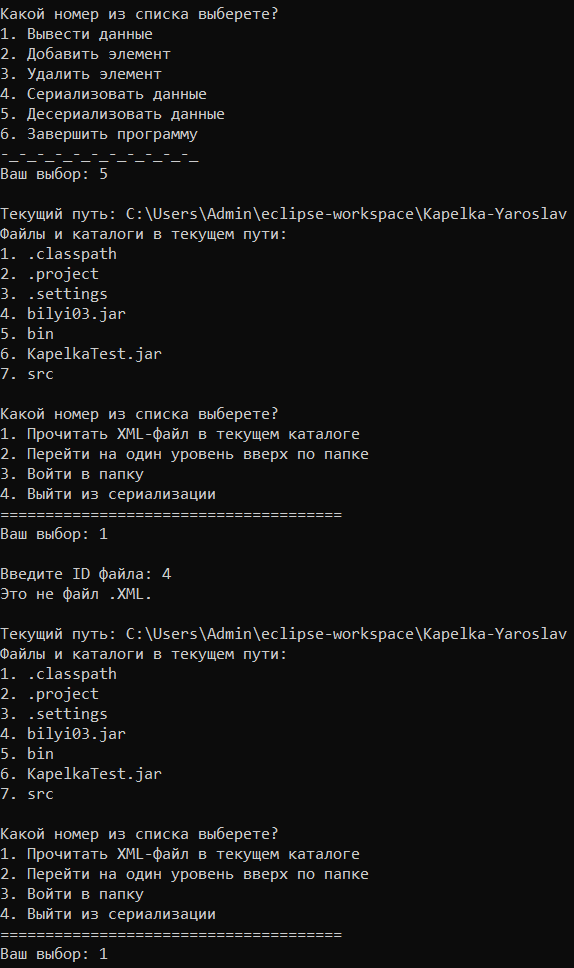


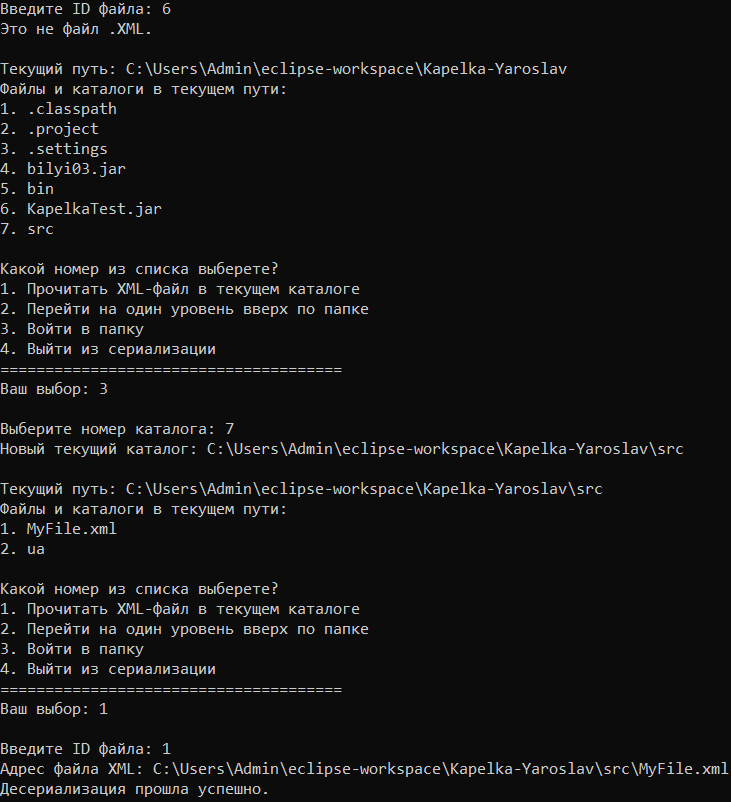


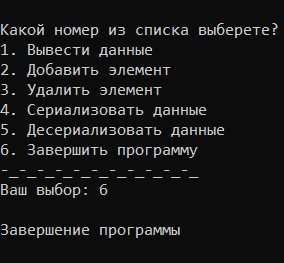
****

****

****

****

****

****

**Висновки**

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

Програма протестована, виконується без помилок.