**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, 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 RoutrClassList, 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** 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** 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 + ") ");

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.***HOUR\_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** GregorianCalendar();

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.***HOUR\_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\_the\_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 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**;

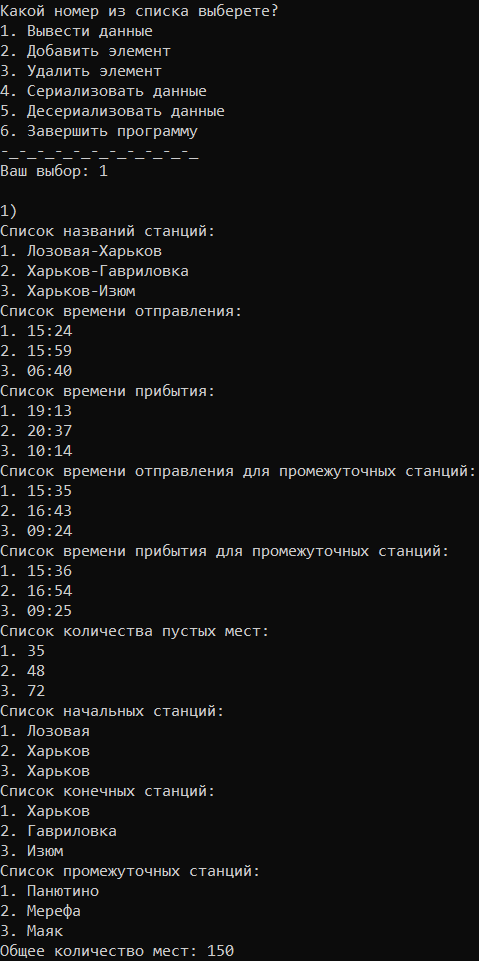
}

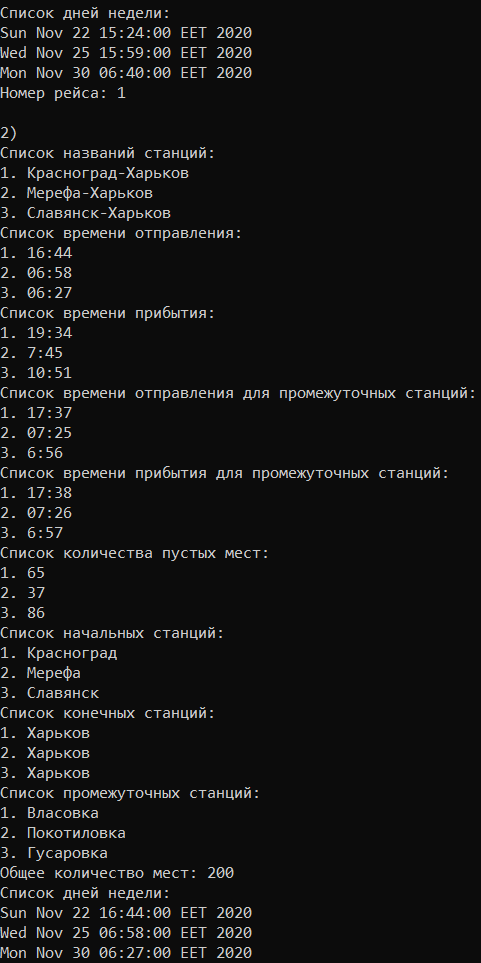
}

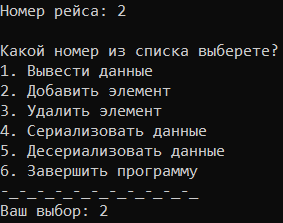
}

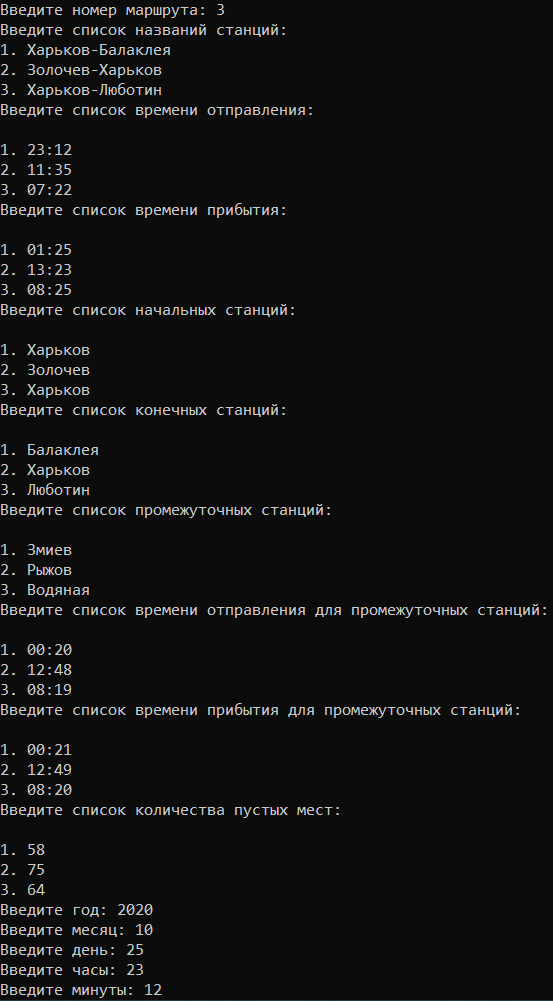
}

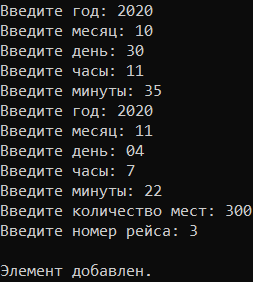
**Результат роботи програми**

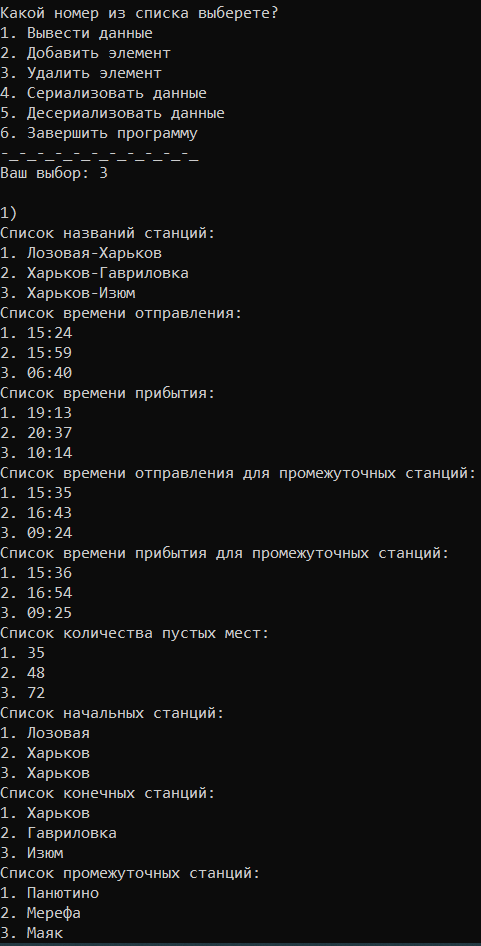


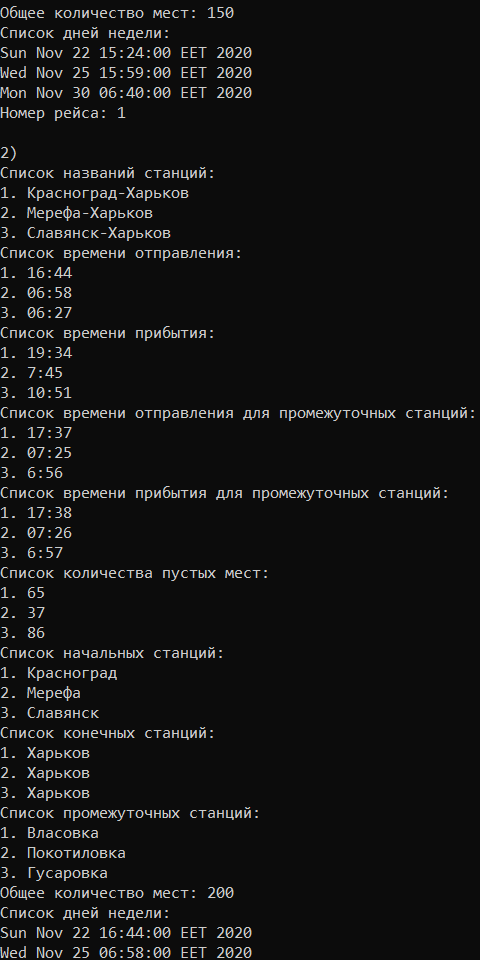


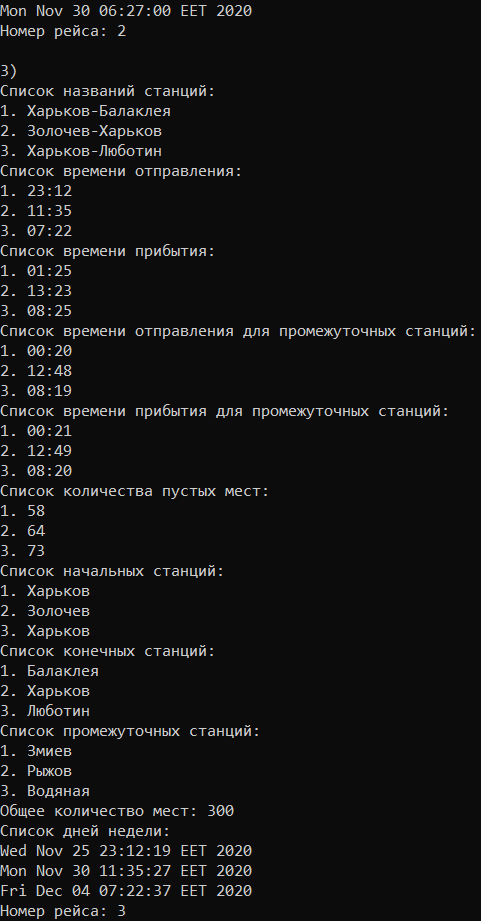


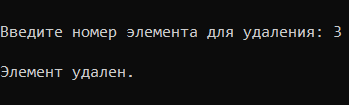
****

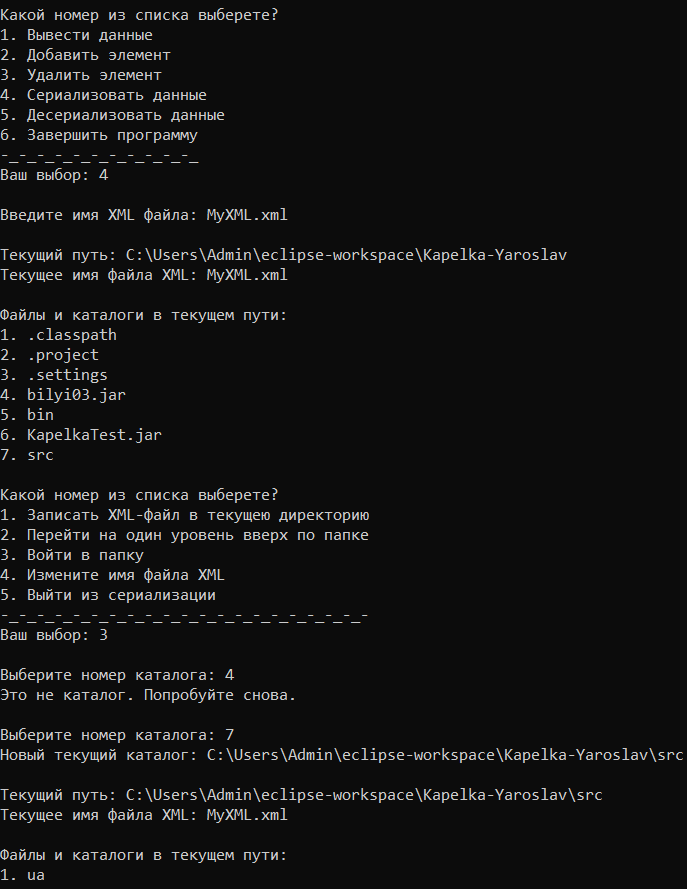
****

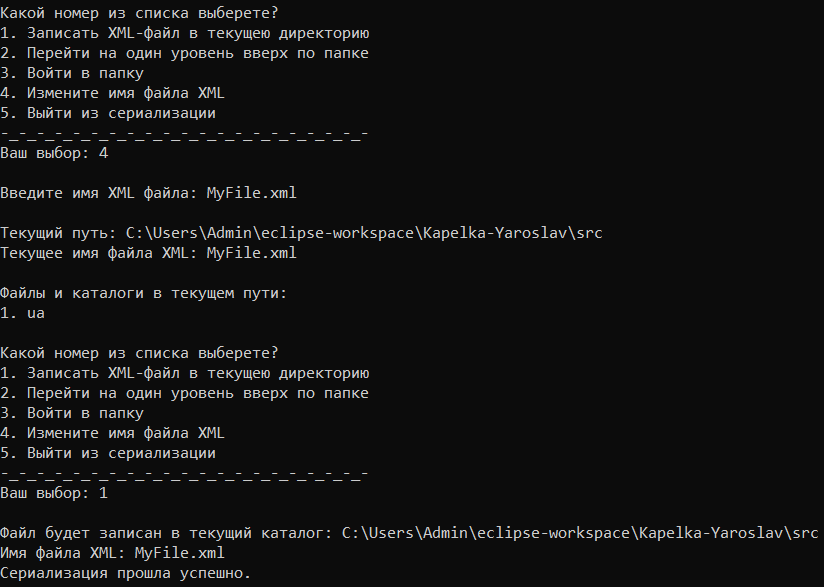
****

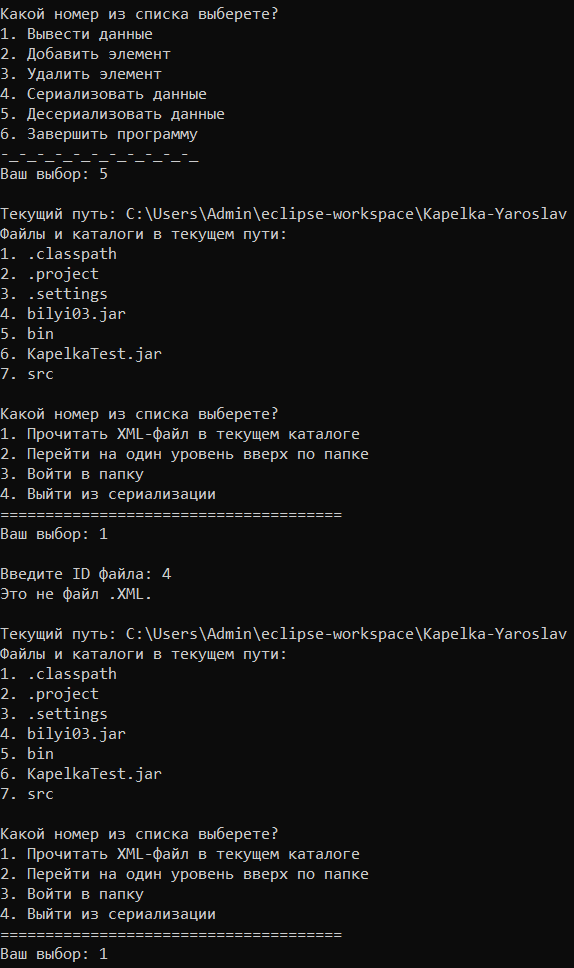
****

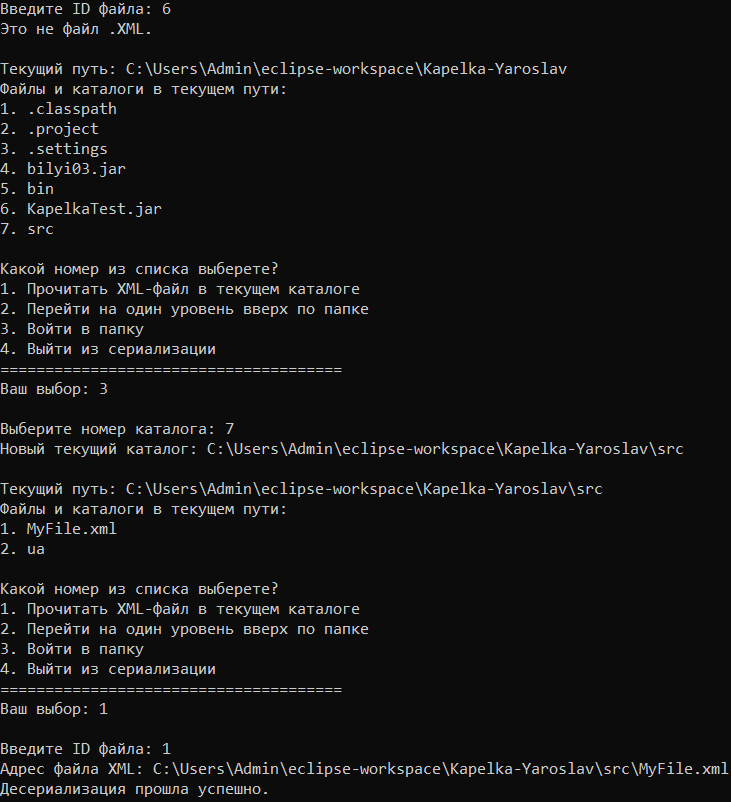
****

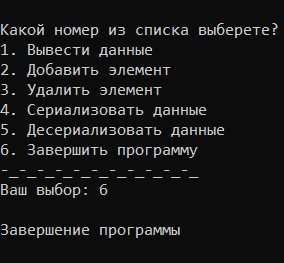
****

****

****

****

****

****

**Висновки**

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

Програма протестована, виконується без помилок.