

# Лабораторна робота №8

## Основи введення/виведення Java SE

### *Мета:*

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

### **1. Вимоги:**

#### **1.1 Розробник**

Інформація про розробника:

- Кедровський Максим
- КІТ-119а
- 10 варіант

#### **1.2 Загальне завдання**

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.

## 2 ОПИС ПРОГРАМИ

Використані у роботі класи Product і Stat наведені у звіті до лабораторної роботи 7.

```
1 package task08;
2
3 import java.beans.XMLDecoder;
4
5 public class Main {
6     private static ArrayList<Product> list = new ArrayList<Product>();
7     // private static Path iopath = Paths.get("D://");
8     private static String name, unitOfMeasurement, stats;
9     private static int amount, price, yearOfArrival, monthOfArrival, dayOfArrival, num;
10    private static Scanner in = new Scanner(System.in);
11    String args[];
12    public static void main(String[] args) {
13        String text;
14        while(true) {
15            System.out.print("1 - Добавление элемента. \n"
16                + "2 - Очистка списка.\n"
17                + "3 - Сохранение в XML.\n"
18                + "4 - Чтение из XML.\n"
19                + "5 - Вывод содержимого. \n"
20                + "6 - Выход. \n");
21            System.out.print("Выберите команду: \n");
22            int number = Integer.parseInt(in.nextLine());
23            switch (number) {
24                case 1:
25                    System.out.println("Название продукта: ");
26                    name = in.nextLine();
27                    System.out.println("Единица измерения: ");
28                    unitOfMeasurement = in.nextLine();
29                    System.out.println("Количество: ");
30                    amount = Integer.parseInt(in.nextLine());
31                    System.out.println("Цена: ");
32                    price = Integer.parseInt(in.nextLine());
33                    System.out.println("Год поставки: ");
34                    yearOfArrival = Integer.parseInt(in.nextLine());
35                    System.out.println("Месяц поставки: ");
36                    monthOfArrival = Integer.parseInt(in.nextLine());
37                    System.out.println("День поставки: ");
38                    dayOfArrival = Integer.parseInt(in.nextLine());
39                    System.out.println("Теперь введите не ограниченное количество характеристик в виде пары \"название-значение\", чтобы прекратить ввод напишите \"выход\"");
40                    System.out.println("Сколько будет характеристик: ");
41                    num = Integer.parseInt(in.nextLine());
42                    args = new String[num*2];
43                    for(int i = 0; i < num; i++) {
44                        System.out.println("Название: ");
45                        args[i*2] = in.nextLine();
46                        System.out.println("Значение: ");
47                        args[i*2+1] = in.nextLine();
48                    }
49                    list.add(new Product(name, unitOfMeasurement, amount, price, yearOfArrival, monthOfArrival, dayOfArrival, args));
50                    break;
51                case 2:
52                    list.clear();
53                    break;
54                case 3:
55                    WriteArrayList(list, PathfinderXML().toString());
56                    break;
57                case 4:
58                    ReadArrayList(list, PathfinderXML().toString());
59                    break;
60                case 5:
61                    num = 1;
62                    System.out.println("-----");
63                    for(Product item : list) {
64                        System.out.println(num++ + ". " + item.toString());
65                        System.out.println("-----");
66                    }
67                    break;
68                case 6:
69                    return;
70            }
71        }
72    }
73    public static Path PathfinderXML() {
74        System.out.println("Из предложенного списка выберите директорию(нет указанного типа файла) для перемещения в нее или введите название файла с окончанием.xml чтобы открыть");
75        Path iopath = Paths.get("");
76        Path newIopath;
77        String text;
78        while(true) {
79            try (DirectoryStream<Path> stream = Files.newDirectoryStream(iopath)) {
80                System.out.println(iopath.toAbsolutePath());
81                System.out.println("-----");
82                for (Path file: stream) {
83                    System.out.println(file.getFileName());
84                }
85                System.out.println("-----");
86            } catch (IOException | DirectoryIteratorException x) {
87                System.err.println("Ошибка! Такого пути не существует");
88            }
89            System.out.println("Выберете .xml-файл или директорию: ");
90            text = in.nextLine();
91            if(text.endsWith(".xml")){
92                return iopath.resolve(Paths.get(text));
93            }
94            else {
95                newIopath = iopath.resolve(Paths.get(text));
96                try (DirectoryStream<Path> stream = Files.newDirectoryStream(newIopath)) {
97                    iopath = newIopath;
98                } catch (IOException | DirectoryIteratorException x) {
99                    System.err.println("Ошибка! Такого пути не существует");
100                }
101            }
102        }
103    }
104}
```

```

124 public static boolean DisplayDirectory(Path iopath) {
125     try (DirectoryStream<Path> stream = Files.newDirectoryStream(iopath)) {
126         for (Path file: stream) {
127             System.out.println(file.getFileName());
128             return true;
129         }
130     } catch (IOException | DirectoryIteratorException x) {
131         System.err.println(x);
132         return false;
133     }
134     return false;
135 }
136 public static boolean WriteArrayList(ArrayList<Product> arr, String dir) {
137     if (dir.endsWith(".xml")) {
138         XMLEncoder encoder;
139         try {
140             encoder = new XMLEncoder(new BufferedOutputStream(new FileOutputStream(dir)));
141             encoder.writeObject(arr);
142             encoder.close();
143             return true;
144         } catch (FileNotFoundException e) {
145             System.out.println("ERROR: No " + dir + " found!");
146             e.printStackTrace();
147             return false;
148         }
149     }
150     else {
151         System.out.println("File name must end with .xml !");
152         return false;
153     }
154 }
155 }
156 public static boolean ReadArrayList(ArrayList<Product> arr, String dir) {
157     if (dir.endsWith(".xml")) {
158         XMLDecoder decoder;
159         try {
160             decoder = new XMLDecoder(new BufferedInputStream(new FileInputStream(dir)));
161             ArrayList<Product> result = (ArrayList<Product>) decoder.readObject();
162             arr.clear();
163             arr.addAll(result);
164             decoder.close();
165             return true;
166         } catch (FileNotFoundException e) {
167             // TODO Auto-generated catch block
168             e.printStackTrace();
169             return false;
170         }
171     }
172     else {
173         System.out.println("File name must end with .xml !");
174         return false;
175     }
176 }

```

Рисунок 8.1 - Код main.java

### 3 ВАРІАНТИ ВИКОРИСТАННЯ

Реалізований функціонал дозволяє у діалоговому режимі працювати зі списком продуктів, зберігати та читати список з .xml-файлів, використовуючи розроблений метод пересування по директоріям комп'ютеру.

### Висновки

Оволодів навичками управління введенням/виведенням даних з використанням класів платформи Java SE.