

Лабораторна робота 6. Серіалізація/десеріалізація об'єктів. Бібліотека класів користувача.

Мета роботи:

- Тривале зберігання та відновлення стану об'єктів.
- Ознайомлення з принципами серіалізації/десеріалізації об'єктів.
- Використання бібліотек класів користувача.

ВИМОГИ

1. Реалізувати і продемонструвати тривале зберігання/відновлення раніше розробленого контейнера за допомогою серіалізації/десеріалізації.
2. Обмінятися відкомпільованим (без початкового коду) службовим класом (Utility Class) рішення задачі л.р. №3 з іншим студентом (визначає викладач).
3. Продемонструвати послідовну та вибірккову обробку елементів розробленого контейнера за допомогою власного і отриманого за обміном службового класу.
4. Реалізувати та продемонструвати порівняння, сортування та пошук елементів у контейнері.
5. Розробити консольну програму та забезпечити діалоговий режим роботи з користувачем для демонстрації та тестування рішення.

1.1. Розробник: Капелька Ярослав Іванович, КІТ119-а, 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 MyContainer, у полі якого знаходиться приватний клас MyIterator та публічний UtilityClass.

2.3. Важливі фрагменти програми:

```
public class Main06
{
    public static void main(String[] args)
    {
        MyContainer06 array = new MyContainer06();
        Scanner scan = new Scanner(System.in);
        int number;
        boolean stop = false;
        String choice;
        String data;

        while(stop != true)
        {
            System.out.println("Введите номер пункта, который хотите выполнить");
            System.out.println("1. Добавить данные");
            System.out.println("2. Вывести данные на экран");
            System.out.println("3. Удалить элемент");
            System.out.println("4. Найти элемент");
            System.out.println("5. Персональное задание");
```

```

System.out.println("6. Сортировать данные");
System.out.println("7. Сравнить массивы");
System.out.println("8. Сериализовать данные");
System.out.println("9. Десериализовать данные");
System.out.println("10. Завершение программы");
System.out.println("-----");
System.out.print("Ваш выбор: ");
choise = scan.nextLine();

switch(choise)
{
case "1":
    System.out.print("Введите обрабатываемый текст: ");
    array.add(scan.nextLine());
    System.out.print("\n");

    break;

case "2":
    if(array.size() != 0)
    {
        System.out.println("\nДанные в массиве:");
        for (int i = 0; i < array.size(); i++)
        {
            System.out.println(i+1 + ". " + array.getLine(i));
        }
        System.out.print("\n");
    }
    else
    {
        System.out.println("\nМассив пуст.\n");
    }

    break;

case "3":
    if(array.size() != 0)
    {
        System.out.print("\nВведите строку для ее удаления из
массива: ");

        array.remove(scan.nextLine());
    }
    else
    {
        System.out.println("\nМассив пуст.\n");
    }

    break;

case "4":
    if(array.size() != 0)
    {
        System.out.print("Введите строку для ее поиска в массиве:

");

        if(array.contains(scan.nextLine()))
        {
            System.out.println("\nСтрока найдена.\n");
        }
        else
        {
            System.out.println("\nСтрока не найдена.\n");
        }
    }
    else

```

```

        {
            System.out.println("\nМассив пуст.\n");
        }

        break;

    case "5":
        if(array.size() != 0)
        {
            for (int i = 0; i < array.size(); i++)
            {
                StringBuilder ln = new
StringBuilder(array.getLine(i));
                String ln1 = new String(array.getLine(i));
                System.out.println("Работа метода чужого класса
лаб. работы № 3" + "\n");
                Helper.start(ln);
                System.out.println("\n" + "Работа метода своего
класса лаб. работы № 3" + "\n");
                util.task(ln1);
                System.out.print("\n");
            }
        }
        else
        {
            System.out.println("\nМассив пуст.\n");
        }

        break;

    case "6":
        if(array.size() != 0)
        {
            array.sort();
        }
        else
        {
            System.out.println("\nМассив пуст.\n");
        }

        System.out.println("\nМассив отсортирован\n");

        break;

    case "7":
        if(array.size() != 0)
        {
            System.out.print("\nВведите количество строк в сравниваемом
массиве: ");
            number = scan.nextInt();

            if(number >= 0)
            {
                MyContainer06 arrayToCompare = new MyContainer06();
                System.out.print("\nВведите обрабатываемый текст:\n");
                scan.nextLine();
                for (int i = 0; i < number; i++)
                {
                    System.out.print(i+1 + ". ");
                    arrayToCompare.add(scan.nextLine());
                    //System.out.println( );
                }
                if(array.containsAll(arrayToCompare) == true)
                {
                    System.out.println("Массивы равны.\n");
                }
            }
            else

```

```

        {
            System.out.println("Массивы не равны.\n");
        }
    }
}
else
{
    System.out.println("\nМассив пуст.\n");
}

break;

case "8":
    if(array.size() != 0)
    {
        System.out.print("\nВведите имя файла: ");
        data = scan.nextLine();
        if(data.indexOf(".ser") == -1)
            data += ".ser";

        try
        {
            FileOutputStream file = new FileOutputStream(data);
            ObjectOutputStream serial = new
ObjectOutputStream(file);

            serial.writeObject(array);
            serial.close();
            System.out.println("\nДанные сериализованы.\n");
        }
        catch(Exception ex)
        {
            System.out.println("\n" + ex.getMessage() + "\n");
        }
    }
    else
    {
        System.out.println("\nМассив пуст.\n");
    }

    break;

case "9":
    System.out.print("\nВведите имя файла: ");
    data = scan.nextLine();
    if(data.indexOf(".ser") == -1)
        data += ".ser";

    try(ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(data)))
    {
        array = (MyContainer06)ois.readObject();
        ois.close();
    }
    catch(Exception ex)
    {
        System.out.println("\n" + ex.getMessage() + "\n");
    }

    break;

case "10":
    stop = true;
    break;

default:
    System.out.println("Ошибка. Неправильная команда.\n");
    break;

```

```

    }
}

System.out.println("\nЗавершение программы.");
array.clear();
scan.close();
}

}

public class MyContainer06 implements Serializable
{
    private String[] array;
    private int size;

    public int size()
    {
        return size;
    }
    public String getLine(int index)
    {
        return array[index];
    }
    public String toString()
    {
        StringBuilder string = new StringBuilder();

        for(int i = 0; i < size; i++)
            string.append(array[i] + " ");

        return string.toString();
    }
    public void add(String string)
    {
        String[] newArray = new String[size+1];

        for (int i = 0; i < size; i++)
            newArray[i] = array[i];

        size++;
        newArray[size - 1] = string;
        array = newArray;
    }
    public void clear()
    {
        for(int i = 0; i < size; i++)
            array[i] = null;

        size = 0;
    }
    public boolean remove(String string)
    {
        boolean result = false;
        int position = 0;

        for (int i = 0; i < size; i++)
            if(array[i].equals(string))
            {
                result = true;
                position = i;
                break;
            }

        if(result)
        {
            String[] newArray = new String[size-1];

```

```

        for (int i = 0; i < position; i++)
            newArray[i] = array[i];
        for (int i = position; i+1 < size; i++)
            newArray[i]=array[i+1];

        size--;
        array=newArray;
    }

    return result;
}
public Object[] toArray()
{
    Object[] object = new Object[size];

    for (int i = 0; i < size; i++)
        object[i]=array[i];

    return object;
}
public boolean contains(String string)
{
    for (int i = 0; i < size; i++)
        if (array[i].equals(string))
            return true;

    return false;
}
public boolean containsAll(MyContainer06 container)
{
    boolean result = false;

    for (int i = 0; i < size; i++)
    {
        result = false;

        for (int j = 0; j < container.size(); j++)
        {
            if(array[i].equals(container.getLine(j)))
            {
                result = true;
                break;
            }
        }

        if(!result)
        {
            return false;
        }
    }

    return result;
}
public MyContainer06(String... strings)
{
    if(strings.length > 0)
    {
        size = strings.length;
        array = new String[size];

        for (int i = 0; i < size; i++)
            array[i]=strings[i];
    }
}
public void sort()

```

```

{
    String temp;

    for(int i = 0; i < size - 1; i++)
    {
        for(int j = i + 1; j < array.length; j++)
        {
            if(array[i].compareTo(array[j]) > 0)
            {
                temp = array[i];
                array[i] = array[j];
                array[j] = temp;
            }
        }
    }
}

public MyIterator<String> getIterator()
{
    return new MyIterator<String>();
}

private class MyIterator<String> implements Iterator
{
    int index;

    @Override public boolean hasNext()
    {
        if(index < size)
            return true;
        else
            return false;
    }

    @Override public Object next()
    {
        if(index==size)
            throw new NoSuchElementException();

        return array[index++];
    }

    @Override public void remove()
    {
        MyContainer06.this.remove(array[--index]);
    }
}
}

```

Результат виконання програми:

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 1

Введите обрабатываемый текст: Сегодня хорошая погода

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 1

Введите обрабатываемый текст: Сегодня хорошая погода

|

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 1

Введите обрабатываемый текст: Сегодня хорошая погода, поют птицы, светит солнце

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 2

|

Данные в массиве:

1. Сегодня хорошая погода
2. Сегодня хорошая погода
3. Сегодня хорошая погода, поют птицы, светит солнце

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 3

Введите строку для ее удаления из массива: Сегодня хорошая погода

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 2

|

Данные в массиве:

1. Сегодня хорошая погода
2. Сегодня хорошая погода, поют птицы, светит солнце

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 4

Введите строку для ее поиска в массиве: Сегодня хорошая погода

Строка найдена.

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 5

Работа метода чужого класса лаб. работы № 3

Сегодня хорошая погода
Строки больше средней длины:
Сегодня
Длина строки:7
хорошая
Длина строки:7
погода
Длина строки:6
Строки меньше средней длины:

Работа метода своего класса лаб. работы № 3

Повторения	Слово
1	погода
1	хорошая
1	Сегодня

Работа метода чужого класса лаб. работы № 3

Сегодня хорошая погода, поют птицы, светит солнце
Строки больше средней длины:
Сегодня
Длина строки:7
хорошая
Длина строки:7
погода,
Длина строки:7
птицы,
Длина строки:6
светит
Длина строки:6
солнце
Длина строки:6
Строки меньше средней длины:
поют

Длина строки:4

Работа метода своего класса лаб. работы № 3

Повторения	Слово
0	погода
1	погода,
1	хорошая
1	поют
1	птицы,
1	светит
1	Сегодня
1	солнце

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 6

|

Массив отсортирован

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 7

Введите количество строк в сравниваемом массиве: 2

Введите обрабатываемый текст:

1. Сегодня хорошая погода
2. Сегодня хорошая погода, поют птицы, светит солнце

Массивы равны.

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 7

Введите количество строк в сравниваемом массиве: 2

Введите обрабатываемый текст:

1. Сегодня хорошая погода
2. Сегодня хорошая погода

Массивы не равны.

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 8

Введите имя файла: Погода.ser

|

Данные сериализованы.

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 9

Введите имя файла: Погода.ser

Введите номер пункта, который хотите выполнить

1. Добавить данные
2. Вывести данные на экран
3. Удалить элемент
4. Найти элемент
5. Персональное задание
6. Сортировать данные
7. Сравнить массивы
8. Сериализовать данные
9. Десериализовать данные
10. Завершение программы

Ваш выбор: 10

|

Завершение программы.

Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи с серіалізацією та десеріалізацією об'єктів .

Програма протестована, виконується без помилок.

