**Звіт**

**Лабораторна работа 9. Параметризація в Java**

**Мета роботи**: Оволодіння навичками управління введенням / виведенням даних з використанням класів платформи Java SE.

**ВИМОГИ**

* Вивчення принципів параметризації в Java.
* Розробка параметризованих класів та методів.
  1. **Розробник**: Момот Роман Євгенійович, КІТ119-а, варіант №14.

1. **ОПИС ПРОГРАМИ**
   1. **Засоби ООП**: клас, метод класу, поле класу.
   2. **Ієрархія та структура класів:** один публічний клас Main, публічний клас Event, 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 Node, який виконує роль покажчика на елемент і клас MyContainer, який містить покажчик на головний елемент та методи обробки масиву елементів.
   3. **Важливі фрагменти програми:**

public class Node<T> implements Serializable{

public T element;

public Node<T> next;

private static final long serialVersionUID = -6298777302126321006L;

public Node() {}

public Node(T el) {

super();

this.element = el;

}

}

public class MyContainer<T> implements Iterable<T>, Serializable {

public Node<T> head;

private int size;

private static final long serialVersionUID = -6153946567197878052L;

public MyContainer() {

super();

}

public int getSize() { return size; }

public void setSize(int size) { this.size = size; }

public T getElement(int id) {

if(id < 0 || id >= size) {

System.out.println("Wrong id.");

return null;

}

Node<T> temp = head;

for(int i = 0; i < id; i++) {

temp = temp.next;

}

return temp.element;

}

public void add(T el) {

Node<T> temp = new Node<T>();

if(head == null) {

head = new Node<T>(el);

}

else {

temp = head;

while(temp.next != null) {

temp = temp.next;

}

temp.next = new Node<T>(el);

}

size++;

}

public void delete(int id) {

Node<T> temp = head;

if(head != null)

{

if(id == 0) {

head = head.next;

}

else {

for(int i = 0; i < id - 1; i++) {

temp = temp.next;

}

if(temp.next != null) {

temp.next = temp.next.next;

}

else {

temp.next = null;

}

}

size--;

}

else {

System.out.println("Container is empty.");

}

}

public void clear() {

this.head = null;

size = 0;

}

public Object[] toArray() {

Object[] arr = new Object[size];

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

arr[i] = getElement(i);

}

return arr;

}

public String toString() {

StringBuilder str = new StringBuilder();

for(T value : this) {

str.append(value + "\n");

}

return str.toString();

}

public boolean checkExistance(T el) {

for(T element : this) {

if(element.equals(el)) {

return true;

}

}

return false;

}

public boolean isEmpty() {

if(size == 0)

return true;

else

return false;

}

public Iterator<T> iterator(){

return new Iterator<T>() {

int index = 0;

boolean check = false;

@Override

public boolean hasNext() {

return index < size;

}

@Override

public T next() {

if (index == size) {

throw new NoSuchElementException();

}

check = true;

return getElement(index++);

}

@Override

public void remove() {

if (check) {

MyContainer.this.delete(index - 1);

check = false;

} else {

throw new IllegalStateException();

}

}

};

}

}

public class Main {

public static void main(String[] args) {

MyContainer<Event> arr = new MyContainer<Event>();

Scanner scan = new Scanner(System.in);

boolean stop = false;

int choise, choise2;

ArrayList<String> people = new ArrayList<String>();

people.add("John");

people.add("Bill");

people.add("Івасик");

Event evToCompare = new Event(new GregorianCalendar(2002,3,28), 120, "ул. Революции",

people, "Pest party ever");

arr.add(evToCompare);

do {

System.out.println("What to do?");

System.out.println("1. Output data");

System.out.println("2. Add element");

System.out.println("3. Delete element");

System.out.println("4. Is empty?");

System.out.println("5. Serialization");

System.out.println("6. Deserialization");

System.out.println("7. Terminate program");

System.out.println("=====================");

System.out.print("Your choise: ");

choise = scan.nextInt();

switch(choise) {

case 1:

System.out.println("\nChoose the output method");

System.out.println("1. Using foreach");

System.out.println("2. Using toArray");

System.out.println("3. Return");

System.out.println("================");

System.out.print("Your choise: ");

choise2 = scan.nextInt();

System.out.println( );

switch(choise2) {

case 1:

if(arr.getSize() > 0){

for(var i : arr) {

i.outputData();

}

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

}

else {

System.out.println("Array is empty.\n");

}

break;

case 2:

if(arr.getSize() > 0) {

Object[] tempArr = arr.toArray();

for (int i = 0; i < tempArr.length; i++) {

System.out.println(i+1 + ")");

((Event)tempArr[i]).outputData();

System.out.println( );

}

}

else {

System.out.println("\nArray is empty.\n");

}

break;

case 3:

break;

default:

System.out.println("You've entered the wrong number");

break;

}

break;

case 2:

Event newEvent = inputNewEvent();

arr.add(newEvent);

break;

case 3:

if(arr.getSize() > 0) {

System.out.print("Enter the index of element: ");

choise = scan.nextInt();

arr.delete(choise);

} else {

System.out.println("Array is empty.");

}

break;

case 4:

if(arr.isEmpty()) {

System.out.println("\nArray is empty.\n");

} else {

System.out.println("\nArray isn't empty.\n");

}

break;

case 5:

System.out.println("\nChoose the method");

System.out.println("1. Standard serialization");

System.out.println("2. XML serialization");

System.out.println("3. Return");

System.out.println("=========================");

System.out.print("Your choise: ");

choise2 = scan.nextInt();

switch(choise2) {

case 1:

scan.nextLine();

System.out.print("\nEnter the name of file: ");

String filename = scan.nextLine();

if (filename.indexOf(".ser") == -1) {

filename += ".ser";

}

try(ObjectOutputStream oos = new ObjectOutputStream(new BufferedOutputStream(new FileOutputStream(filename)))){

oos.writeObject(arr);

System.out.println("Serialization successful.\n");

}catch(Exception ex){

System.out.println(ex.getMessage());

ex.printStackTrace();

}

break;

case 2:

scan.nextLine();

System.out.print("\nEnter the name of file: ");

filename = scan.nextLine();

if (filename.indexOf(".xml") == -1) {

filename += ".xml";

}

try(XMLEncoder encoder = new XMLEncoder(new BufferedOutputStream(new FileOutputStream(filename)))){

encoder.writeObject(arr);

System.out.println("Serialization successful.\n");

}

catch(Exception ex){

System.out.println(ex.getMessage());

}

break;

case 3:

break;

default:

System.out.println("You've entered the wrong command.");

break;

}

break;

case 6:

System.out.println("\nChoose the method");

System.out.println("1. Standard deserialization");

System.out.println("2. XML deserialization");

System.out.println("3. Return");

System.out.println("=========================");

System.out.print("Your choise: ");

choise2 = scan.nextInt();

switch(choise2) {

case 1:

scan.nextLine();

System.out.print("\nEnter the name of file: ");

String filename = scan.nextLine();

if (filename.indexOf(".ser") == -1) {

filename += ".ser";

}

try(ObjectInputStream oos = new ObjectInputStream(new BufferedInputStream(new FileInputStream(filename)))){

arr.clear();

arr = (MyContainer<Event>) oos.readObject();

System.out.println("\nDeserialization successful.");

}catch(Exception ex){

System.out.println(ex.getMessage());

}

break;

case 2:

scan.nextLine();

System.out.print("\nEnter the name of file: ");

filename = scan.nextLine();

if (filename.indexOf(".xml") == -1) {

filename += ".xml";

}

try(XMLDecoder decoder = new XMLDecoder(new BufferedInputStream(new FileInputStream(filename)))){

arr.clear();

arr = (MyContainer<Event>) decoder.readObject();

System.out.println("Deserialization successful.\n");

}catch(IOException ex){

System.out.println( );

}

break;

case 3:

break;

default:

System.out.println("You've entered the wrong command.");

break;

}

break;

case 7:

System.out.println("Terminating the program.\n");

stop = true;

break;

default:

System.out.println("You have entered the wrong number.");

break;

}

}while(!stop);

scan.close();

}

private static Event inputNewEvent(){

Scanner scan = new Scanner(System.in);

int value;

boolean ready = false;

do {

System.out.print("\nEnter number of participants: ");

value = scan.nextInt();

if(value < 1)

{

System.out.println("Error. Wrong list size.\n");

}

ready = true;

}while(!ready);

ArrayList<String> list = new ArrayList<String>();

String temp;

System.out.println("Enter list of names:");

scan.nextLine();

for (int i = 0; i < value; i++) {

System.out.print(i+1 + ". ");

temp = scan.nextLine();

list.add(temp);

}

GregorianCalendar date = new GregorianCalendar();

System.out.print("Enter event year: ");

value = scan.nextInt();

date.set(Calendar.YEAR, value);

System.out.print("Enter event month: ");

value = scan.nextInt();

date.set(Calendar.MONTH, value-1);

System.out.print("Enter event day: ");

value = scan.nextInt();

date.set(Calendar.DAY\_OF\_MONTH, value);

System.out.print("Enter event hour: ");

value = scan.nextInt();

date.set(Calendar.HOUR\_OF\_DAY, value);

System.out.print("Enter event minute: ");

value = scan.nextInt();

date.set(Calendar.MINUTE, value);

date.set(Calendar.SECOND, 0);

System.out.print("Enter event address: ");

scan.nextLine();

temp = scan.nextLine();

System.out.print("Enter event description: ");

String description = scan.nextLine();

System.out.print("Enter event length: ");

value = scan.nextInt();

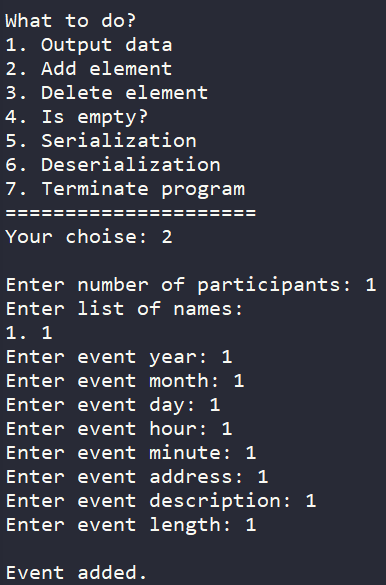
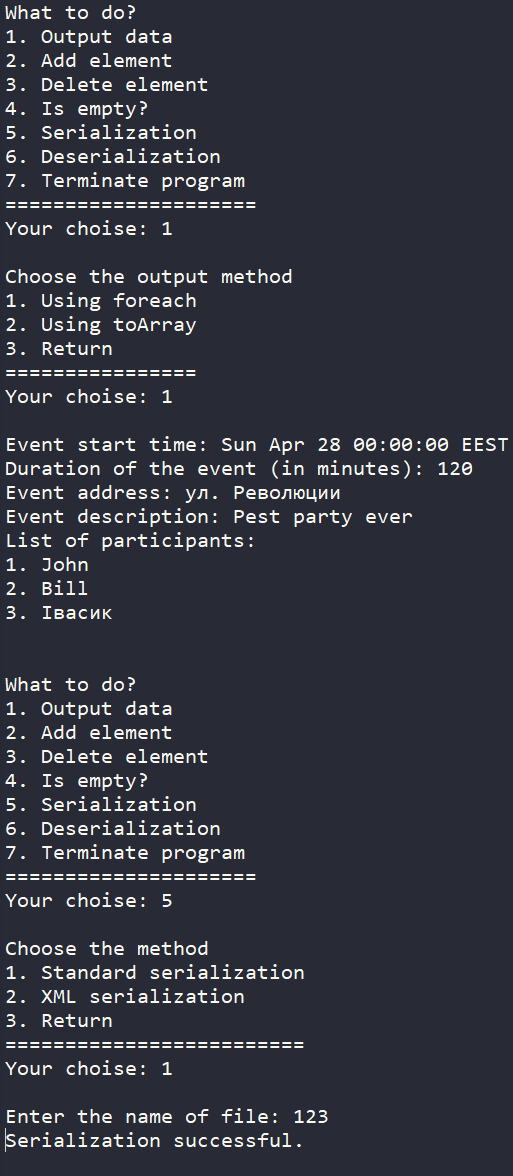
System.out.println("\nEvent added.\n");

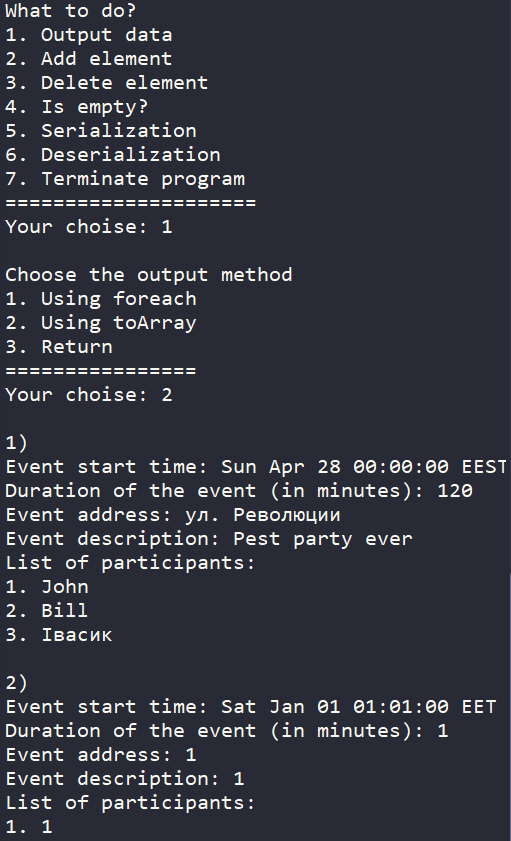
Event newEvent = new Event(date,value,temp,list,description);

return newEvent;

}

}

**Результат роботи програми**



**Висновки**

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з параметризацією.

Програма протестована, виконується без помилок.