

Лабораторна робота №3

Тема: Використання узагальнень (generics). Клонування та порівняння об'єктів.

Мета роботи: створити міні проект Game з використанням узагальнень, клонування та порівняння об'єктів. Завдання на лабораторну роботу

Завдання 1. Відкрити заготовлений проект з реалізованою базовою функціональністю.

Завдання 2. За допомогою узагальнень (generics) встановити такі обмеження:

- до команди можна додавати тільки учасників, що відносяться до одної ліги (Scholar, Student або Employee).
- - грати між собою можуть тільки команди з учасниками одної ліги (тобто команда студентів може грати тільки іншою командою студентів).
- продемонструвати створення команд, гравців, додавання гравців до команд, гри між ними.

Завдання 3. Клонування:

- для класу Participant імплементувати інтерфейс Cloneable та перевизначити метод clone.
- для класу Participant перевизначити методи hashCode та equals.
- для класу Participant та його підкласів перевизначити метод toString.
- для класу Team Реалізувати глибоке клонування через статичний метод або конструктор копіювання.
- продемонструвати клонування та використання методів hashCode, equals та toString.

Завдання 4. Порівняння:

- для класу Participant імплементувати інтерфейс Comparable та перевизначити метод compareTo для сортування учасників по імені.
- створити Comparator для порівняння учасників по віку.
- *створити компаратор з пріоритетом використовуючи можливості Java 8 (спочатку порівняння по імені, а потім по віку).
- продемонструвати роботу порівнянь на прикладі сортування учасників команд.

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.01.000 – Пр 1		
Розроб.	Aйсін В.С						
Перевір.	Піонтківський ВІ						
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					Lіт.	Арк.	Аркушів
						1	11
					ФІКТ Гр. ІПЗ-23-1[1]		

```

AgeComparator
package com.education.ztu.game;

import java.util.Comparator;

public class AgeComparator implements Comparator<Participant> {
    @Override
    public int compare(Participant o1, Participant o2) {
        return o1.getAge() - o2.getAge();
    }
}
AgeNameComparator
package com.education.ztu.game;

import java.util.Comparator;

public class AgeNameComparator implements Comparator<Participant> {
    @Override
    public int compare(Participant p1, Participant p2) {
        int nameComparison = p1.getName().compareTo(p2.getName());
        if (nameComparison != 0) {
            return nameComparison;
        }
        return p1.getName().compareTo(p2.getName());
    }
}
Employee
package com.education.ztu.game;

public class Employee extends Participant{
    public Employee(String name, int age) {
        super(name, age);
    }
}
Game
package com.education.ztu.game;

public class Game {
    public static void main(String[] args) {
        Scholar scholar1 = new Scholar("Ivan", 13);
        Scholar scholar2 = new Scholar("Mariya", 15);
        Student student1 = new Student("Mykola", 20);
        Student student2 = new Student("Viktoria", 21);
        Employee employee1 = new Employee("Andriy", 28);
        Employee employee2 = new Employee("Oksana", 25);
        Team<Scholar> scholarTeam = new Team<>("Dragon");
        scholarTeam.addNewParticipant(scholar1);
        scholarTeam.addNewParticipant(scholar2);
        Team<Student> studentTeam = new Team<>("Vpered");
        studentTeam.addNewParticipant(student1);
        studentTeam.addNewParticipant(student2);
        Team<Employee> employeeTeam = new Team<>("Robotyagi");
        employeeTeam.addNewParticipant(employee1);
        employeeTeam.addNewParticipant(employee2);
        Team<Scholar> scholarTeam2 = new Team<>("Rozumnyky");
        Scholar scholar3 = new Scholar("Sergey", 12);
        Scholar scholar4 = new Scholar("Olga", 14);
        scholarTeam2.addNewParticipant(scholar3);
        scholarTeam2.addNewParticipant(scholar4);
        scholarTeam.playWith(scholarTeam2);
    }
}
Main
package com.education.ztu.game;

public class Main {

    public static void main(String[] args) {
        Scholar scholar1 = new Scholar("Ivan", 13);
        System.out.println("Original: " + scholar1);
        Scholar clonedSchoolar = (Scholar) scholar1.clone();
        System.out.println("Cloned: " + clonedSchoolar);
        System.out.println("Equals? " + scholar1.equals(clonedSchoolar));
        System.out.println("Original hashCode: " + scholar1.hashCode());
        System.out.println("Clone hashCode: " + clonedSchoolar.hashCode());
        Team<Scholar> scholarTeam = new Team<>("Dragons");
        scholarTeam.addNewParticipant(scholar1);
    }
}

```

		<i>Аїсін В.С</i>				
		<i>Любомирський В І</i>				
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.01.000 – Пр1	Арк. 2

```

        scholarTeam.addNewParticipant(new Scholar("Mariya", 15));
        Team<Scholar> clonedTeam = scholarTeam.deepClone();
        System.out.println("\nOriginal team participants:");
        scholarTeam.getParticipants().forEach(System.out::println);
        System.out.println("\nCloned team participants:");
        clonedTeam.getParticipants().forEach(System.out::println);
    }
}
Main2
package com.education.ztu.game;

public class Main2 {
    public static void main(String[] args) {
        Team<Student> studentTeam = new Team<>("Vpered");
        studentTeam.addNewParticipant(new Student("Mykola", 20));
        studentTeam.addNewParticipant(new Student("Viktoria", 21));
        studentTeam.addNewParticipant(new Student("Anna", 22));

        System.out.println("\nSorting by name:");
        studentTeam.sortByName();
        studentTeam.getParticipants().forEach(System.out::println);

        System.out.println("\nSorting by age:");
        studentTeam.sortByAge();
        studentTeam.getParticipants().forEach(System.out::println);

        System.out.println("\nSorting by name then age:");
        studentTeam.sortByNameThenAge();
        studentTeam.getParticipants().forEach(System.out::println);
    }
}
Participant
package com.education.ztu.game;

import java.util.Objects;

public abstract class Participant implements Cloneable, Comparable<Participant> {
    private String name;
    private int age;

    public Participant(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public int compareTo(Participant other) {
        return this.name.compareTo(other.name);
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Participant that = (Participant) obj;
        return age == that.age && Objects.equals(name, that.name);
    }

    @Override
    public int hashCode() {
        return Objects.hash(name, age);
    }
}

```

		<i>Аүсін В.С</i>			<i>ДУ «Житомирська політехніка».25.121.01.000 – Пр1</i>	<i>Арк.</i>
		<i>Плюнгівський В І</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>3</i>

```

    }

    @Override
    public Participant clone() {
        try {
            return (Participant) super.clone();
        } catch (CloneNotSupportedException e) {
            throw new RuntimeException("Clone not supported", e);
        }
    }

    @Override
    public String toString() {
        return getClass().getSimpleName() + "{name='" + name + "', age=" + age + "}";
    }
}

Schoolar
package com.education.ztu.game;

public class Schoolar extends Participant{
    public Schoolar(String name, int age) {
        super(name, age);
    }
}
Student
package com.education.ztu.game;

public class Student extends Participant{
    public Student(String name, int age) {
        super(name, age);
    }
}
Team
package com.education.ztu.game;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Random;

class Team<T extends Participant> {
    private String name;
    private List<T> participants = new ArrayList<>();
    public Team(String name) {
        this.name = name;
    }
    public void addNewParticipant(T participant) {
        participants.add(participant);
        System.out.println("To the team " + name + " was added participant " + participant.getName());
    }
    public void playWith(Team<T> team) {
        String winnerName;
        Random random = new Random();
        winnerName = random.nextBoolean() ? this.name : team.name;
        System.out.println("The team " + winnerName + " is the winner!");
    }
    public String getName() {
        return name;
    }
    public List<T> getParticipants() {
        return participants;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setParticipants(List<T> participants) {
        this.participants = participants;
    }
    public Team<T> deepClone() {
        Team<T> clonedTeam = new Team<>(this.name + " Clone");
        for (T participant : this.participants) {
            clonedTeam.addNewParticipant((T)participant.clone());
        }
    }
}

```

		<i>Аїсін В.С</i>			<i>ДУ «Житомирська політехніка».25.121.01.000 – Пр1</i>	Арк.
		<i>Люнгківський В І</i>				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

        return clonedTeam;
    }
    public void sortByName() {
        Collections.sort(participants);
    }

    public void sortByAge() {
        participants.sort(new AgeComparator());
    }

    public void sortByNameThenAge() {
        participants.sort(new AgeNameComparator());
    }
}

To the team Dragon was added participant Ivan
To the team Dragon was added participant Mariya
To the team Vpered was added participant Mykola
To the team Vpered was added participant Viktoria
To the team Robotyagi was added participant Andriy
To the team Robotyagi was added participant Oksana
To the team Rozumnyky was added participant Sergey
To the team Rozumnyky was added participant Olga
The team Dragon is the winner!

```

Рис1 Результат виконання завдання 2

```

Original: Schoolar{name='Ivan', age=13}
Cloned: Schoolar{name='Ivan', age=13}
Equals? true
Original hashCode: 71029972
Clone hashCode: 71029972
To the team Dragons was added participant Ivan
To the team Dragons was added participant Mariya
To the team Dragons Clone was added participant Ivan
To the team Dragons Clone was added participant Mariya

Original team participants:
Schoolar{name='Ivan', age=13}
Schoolar{name='Mariya', age=15}

Cloned team participants:
Schoolar{name='Ivan', age=13}
Schoolar{name='Mariya', age=15}

```

Рис2 Результат виконання завдання 3

		<i>Аїсін В.С</i>			<i>ДУ «Житомирська політехніка».25.121.01.000 – Пр1</i>	<i>Арк.</i>
		<i>Люнгківський В.І</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>5</i>

```
To the team Vpered was added participant Mykola  
To the team Vpered was added participant Viktoria  
To the team Vpered was added participant Anna
```

Sorting by name:

```
Student{name='Anna', age=22}  
Student{name='Mykola', age=20}  
Student{name='Viktoria', age=21}
```

Sorting by age:

```
Student{name='Mykola', age=20}  
Student{name='Viktoria', age=21}  
Student{name='Anna', age=22}
```

Sorting by name then age:

```
Student{name='Anna', age=22}  
Student{name='Mykola', age=20}  
Student{name='Viktoria', age=21}
```

Рис3 Результат виконання завдання 4

		<i>Aїсін В.С</i>			ДУ «Житомирська політехніка».25.121.01.000 – Пр1	Арк.
		<i>Піонерківський В І</i>				
Змн.	Арк.	№ докум.	Підпис	Дата		6