

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.

Хід роботи:

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.

Лістинг програми:

Employee.java:

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

					ДУ «Житомирська політехніка».22.121.16.000 – Лр3						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Кохан Т.О			Звіт з лабораторної роботи №3	Лім.		Арк.		Аркушів	
Перевір.		Піонтківський В. І.						1		8	
Керівник						ФІКТ Гр. ІПЗ-22-3					
Н. контр.											
Зав. каф.		Вакалюк Т.А.									

Participant.java:

```
package com.education.ztu.game;

public abstract class 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;
    }
}
```

Schoolar.java:

```
package com.education.ztu.game;

public class Schoolar extends Participant{
    public Schoolar(String name, int age) {
        super(name, age);
    }
}
```

Student.java:

```
package com.education.ztu.game;

public class Student extends Participant{
    public Student(String name, int age) {
        super(name, age);
    }
}
```

Team.java:

```
package com.education.ztu.game;

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

public class Team<T> extends Participant{
    private String name;
    private List<T> participants = new ArrayList<>();
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Піонтківський В. І.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

    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) {
        System.out.println("Team " + this.name + " is playing with team " +
team.name);
        String winnerName;
        Random random = new Random();
        winnerName = random.nextInt(2) == 0 ? this.name : team.name;
        System.out.println("The team " + winnerName + " is 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;
    }
}

```

Game.java:

```

package com.education.ztu.game;

public class Game {
    public static void main(String[] args) {
        Schoolar scholar1 = new Schoolar("Ivan", 13);
        Schoolar scholar2 = new Schoolar("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<Schoolar> scollarTeam = new Team<>("Dragon");
        scollarTeam.addNewParticipant(scholar1);
        scollarTeam.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<Schoolar> scollarTeam2 = new Team<>("Rozumnyky");
        Schoolar scholar3 = new Schoolar("Sergey", 12);
        Schoolar scholar4 = new Schoolar("Olga", 14);
        scollarTeam2.addNewParticipant(scholar3);
        scollarTeam2.addNewParticipant(scholar4);
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Піонтківський В. І.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        scollarTeam.playWith(scollarTeam2);
    }
}

```

Результат виконання програми:

```

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
Team Dragon is playing with team Rozumnyky
The team Dragon is winner!

```

Рис.1 Завдання 2

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.

Participant.java:

```

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() {

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Піонтківський В. І.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        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 31 * name.hashCode() + age;
    }

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

```

Team.java:

```

package com.education.ztu.game;

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

public class Team<T extends Participant> implements Cloneable{
    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) {
        System.out.println("Team " + this.name + " is playing with team " +
        team.name);
        String winnerName;
        Random random = new Random();
        winnerName = random.nextInt(2) == 0 ? this.name : team.name;
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Піонтківський В. І.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        System.out.println("The team " + winnerName + " is winner!");
    }

    public Team<T> cloneTeam() {
        Team<T> clonedTeam = new Team<>(this.name);
        for (T participant : participants) {
            clonedTeam.addNewParticipant(participant);
        }
        return clonedTeam;
    }

    @Override
    public String toString() {
        return "Team{name='" + name + "', participants=" + participants + '}';
    }
}

```

Main.java:

```

package com.education.ztu;

import com.education.ztu.game.*;

public class Main {

    public static void main(String[] args) {
        Scholar scholar1 = new Scholar("Olivia", 23);
        Scholar scholar2 = new Scholar("Inna", 14);

        Team<Scholar> scholarTeam = new Team<>("Star");
        scholarTeam.addNewParticipant(scholar1);
        scholarTeam.addNewParticipant(scholar2);

        Team<Scholar> clonedTeam = scholarTeam.cloneTeam();
        System.out.println("Original team: " + scholarTeam);
        System.out.println("Cloned team: " + clonedTeam);

        System.out.println("Scholar1 equals scholar2: " +
        scholar1.equals(scholar2));
        System.out.println("Scholar1 hashCode: " + scholar1.hashCode());
        System.out.println("Scholar2 hashCode: " + scholar2.hashCode());
    }
}

```

Результат виконання програми:

```

To the team Star was added participant Olivia
To the team Star was added participant Inna
To the team Star was added participant Olivia
To the team Star was added participant Inna
Original team: Team{name='Star', participants=[Participant{name='Olivia', age=23}, Participant{name='Inna', age=14}]}
Cloned team: Team{name='Star', participants=[Participant{name='Olivia', age=23}, Participant{name='Inna', age=14}]}
Scholar1 equals scholar2: false
Scholar1 hashCode: 290884053
Scholar2 hashCode: 70802774

```

Рис.2 Завдання 3

Завдання 4. Порівняння:

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Піонтківський В. І.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

- 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.

Лістинг програми:

AgeComparator.java:

```
package com.education.ztu.game;

import java.util.Comparator;

public class AgeComparator implements Comparator<Participant> {
    @Override
    public int compare(Participant p1, Participant p2) {
        return Integer.compare(p1.getAge(), p2.getAge());
    }
}
```

Participant.java:

```
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);
    }
}
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Піонтківський В. І.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    @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 31 * name.hashCode() + age;
    }

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

```

Main2.java:

```

package com.education.ztu;

import com.education.ztu.game.AgeComparator;
import com.education.ztu.game.Schoolar;

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

public class Main2 {
    public static void main(String[] args) {
        Schoolar schoolar1 = new Schoolar("Anna", 13);
        Schoolar schoolar2 = new Schoolar("Vika", 15);
        Schoolar schoolar3 = new Schoolar("Oleg", 20);

        List<Schoolar> schoolars = new ArrayList<>();
        schoolars.add(schoolar1);
        schoolars.add(schoolar2);
        schoolars.add(schoolar3);

        Collections.sort(schoolars);
        System.out.println("Sorted by name: " + schoolars);

        Collections.sort(schoolars, new AgeComparator());
        System.out.println("Sorted by age: " + schoolars);
    }
}

```

Результат виконання програми:

```

Sorted by name: [Participant{name='Anna', age=13}, Participant{name='Oleg', age=20}, Participant{name='Vika', age=15}]
Sorted by age: [Participant{name='Anna', age=13}, Participant{name='Vika', age=15}, Participant{name='Oleg', age=20}]

```

Рис.3 Завдання 4

Висновок: під час виконання лабораторної роботи я створила міні проект Game з використанням узагальнень, клонування та порівняння об'єктів.

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Піонтківський В. І.				8
Змн.	Арк.	№ докум.	Підпис	Дата		