

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №4**

**з дисципліни «Алгоритми та структури даних»**

**на тему: «Робота з шаблонами (generic java).»**

Виконав:

студент гр. ПЗ2011 Савенко Я. О.

Прийняла:

Куроп’ятник О. С.

Дніпро, 2022

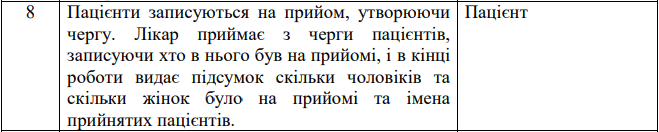
**Тема:** Робота з шаблонами (generic java).

**Мета:** Ознайомитися з поняттям узагальнення, варіантами його застосування. Отримати практичні навички роботи з колекціями як різновидом узагальнень.

**Завдання**

Виконайте моделювання ситуації, описаної в індивідуальному завданні, з використанням структур Java Collection Framework. Обґрунтуйте вибір структури даних, виконайте аналіз можливих альтернатив.

**Індивідуальне завдання**



**Вибір та обгрунтування коллекції**

Для моделювання ситуації із завдання була обрана структура даних «Черга». Так як лікар приймає пацієнтів в черзі, то і при виконанні завдання була обрана саме така структура даних.

**Текст програми**

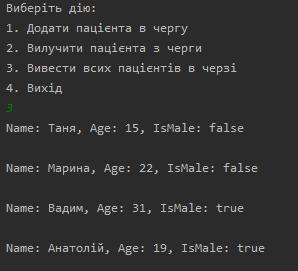
*Файл Patient.java*

package com.company;  
  
public class Patient  
{  
 private String name;  
 private int age;  
 private boolean isMale;  
  
 public Patient(String name, int age, boolean isMale)  
 {  
 this.name = name;  
 this.age = age;  
 this.isMale = isMale;  
 }  
  
 public String getName()  
 {  
 return name;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
  
 public int getAge()  
 {  
 return age;  
 }  
  
 public void setAge(int age)  
 {  
 this.age = age;  
 }  
  
 public boolean IsMale()  
 {  
 return isMale;  
 }  
  
 @Override  
 public String toString()  
 {  
 return "Name: " + name + ", Age: " + age + ", IsMale: " + isMale + '\n';  
 }  
}

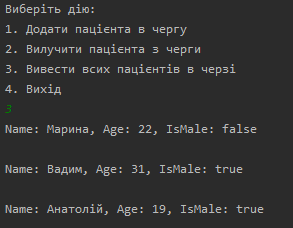
*Файл main.java*

package com.company;  
import java.util.LinkedList;  
import java.util.Queue;  
import java.util.Scanner;  
public class Main {  
  
 public static void main(String[] args) {  
 Queue<Patient> queue = new LinkedList<>();  
 LinkedList<String> namesPatients = new LinkedList<String>();  
 int maleCount = 0;  
 int femaleCount = 0;  
  
 while (true) {  
 System.out.println("Виберіть дію: ");  
 System.out.println("1. Додати пацієнта в чергу");  
 System.out.println("2. Вилучити пацієнта з черги");  
 System.out.println("3. Вивести всих пацієнтів в черзі");  
 System.out.println("4. Вихід");  
  
 int choice = Integer.parseInt(new Scanner(System.in).nextLine());  
 switch (choice) {  
 case 1:  
 System.out.println("Введіть ім'я пацієнта: ");  
 String name = new Scanner(System.in).nextLine();  
 System.out.println("Введіть вік пацієнта: ");  
 int age = Integer.parseInt(new Scanner(System.in).nextLine());  
 System.out.println("Це чоловік? (true or false): ");  
 boolean isMale = Boolean.parseBoolean(new Scanner(System.in).nextLine());  
 queue.add(new Patient(name, age, isMale));  
 if (isMale) {  
 maleCount++;  
 } else {  
 femaleCount++;  
 }  
 namesPatients.add(name);  
 break;  
 case 2:  
 if (queue.isEmpty()) {  
 System.out.println("Черга пуста");  
 break;  
 }  
 queue.remove();  
 break;  
 case 3:  
 for (Patient patient : queue) {  
 System.out.println(patient);  
 }  
 break;  
 case 4:  
 System.out.println("Кількість чоловіків: " + maleCount);  
 System.out.println("Кількість жінок: " + femaleCount);  
 System.out.println("Список пацієнтів: " + namesPatients);  
  
 System.exit(0);  
 default:  
 System.out.println("Введіть коректне значення");  
 break;  
 }  
 }  
 }  
}

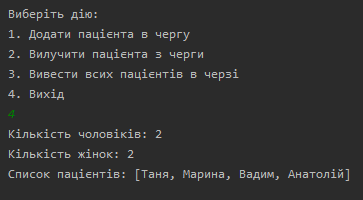
**Результат роботи**

****

*Рис. 1 – Початкова черга*

****

*Рис. 2 – Черга після вилучення пацієнта*

****

*Рис. 3 – Кінець роботи програми*

**Висновок**

Для повторного використання коду, що може бути досягнуто завдяки використання структур та алгоритмів, які не прив’язані до типів даних, використовують узагальнення – параметризовані типи. Вони застосовуються до класів, інтерфейсів, методів.

Для максимально ефективної реалізації поставленого завдання велику роль грає коректність обраної структури, адже правильно обрана структура даних допомагає пришвидшити написання програми, використовувати всі переваги обраної структури.