

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №3**

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

**на тему: «Динамічні лінійними структури даних.»**

Виконав:

студент гр. ПЗ2011 Кулик С. В.

Прийняла:

Демидович І. М.

Дніпро, 2022

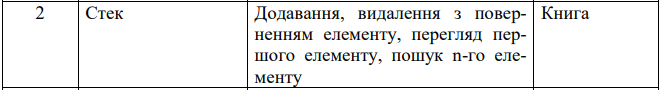
**Тема.** Динамічні лінійними структури даних.

**Мета.** Ознайомитися з поняттям структури даних та основними динамічними лінійними структурами даних. Отримати практичні навички роботи з динамічними лінійними структурами даних.

**Завдання**

Написати програму мовою java для роботи зі структурою даних, що має зв’язне представлення. Вид структури, операції над структурою, зміст інформаційної частини визначається індивідуальним завданням. Програма має містити текстове меню, за допомогою якого можна викликати усі операції над структурою.

**Індивідуальне завдання**

****

**Текст програми**

Main.java

package com.company;  
import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
  
 Stack<Student> stack = new Stack<>(5);  
  
 int choice = 0;  
 Scanner input = new Scanner(System.in);  
 while (true) {  
 System.out.println("Виберіть дію:");  
 System.out.println("1. ввести элемент в стек");  
 System.out.println("2. вивести елемент із стеку");  
 System.out.println("3. вивести верхній елемент стеку");  
 System.out.println("4. вивести всі элементи стеку");  
 System.out.println("5. выход");  
 choice = input.nextInt();  
 switch (choice) {  
 case 1: {  
 System.out.print("Введіть ім'я:");  
 String name = input.next();  
 System.out.print("Введіть вік:");  
 int age = input.nextInt();  
 stack.push(new Student(name, age));  
 break;  
 }  
 case 2: {  
 Student student = stack.pop();  
 System.out.println(student);  
 break;  
 }  
 case 3: {  
 Student student = stack.peek();  
 System.out.println(student);  
 break;  
 }  
 case 4: {  
 stack.print();  
 break;  
 }  
 case 5: {  
 System.exit(0);  
 }  
 }  
 }  
 }  
}

Stack.java

package com.company;  
  
//клас для реалізації стеку  
public class Stack<T> {  
 private int size;  
 private int capacity;  
 private T[] array;  
  
 //конструктор  
 public Stack(int capacity) {  
 this.capacity = capacity;  
 array = (T[]) new Object[capacity];  
 }  
  
 //додавання елемента в стек  
 public void push(T value) {  
 if (size == capacity) {  
 System.out.println("Стек переповнений!");  
 return;  
 }  
 array[size++] = value;  
 }  
  
 //витягування елемента зі стеку  
 public T pop() {  
 if (size == 0) {  
 System.out.println("Стек пустий!");  
 return null;  
 }  
 return array[--size];  
 }  
  
 //взяти верхній елемент стеку  
 public T peek() {  
 if (size == 0) {  
 System.out.println("Стек пустий!");  
 return null;  
 }  
 return array[size - 1];  
 }  
  
 //друк стеку  
 public void print() {  
 for (int i = 0; i < size; i++) {  
 System.out.print(array[i] + " \n");  
 }  
 }  
  
 public int getSize() {  
 return size;  
 }  
  
 public int getCapacity() {  
 return capacity;  
 }  
  
 public boolean isEmpty() {  
 return size == 0;  
 }  
  
 public boolean isFull() {  
 return size == capacity;  
 }  
}

Student.java

package com.company;  
  
public class Student  
{  
 private String name;  
 private int age;  
  
 public Student(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;  
 }  
  
 public String toString()  
 {  
 return "Name: " + name + " Age: " + age;  
 }  
}

**Висновок**

Під час виконання лабораторної роботи займався реалізацією стеку на мові java. Стек структура даних, яка має доступ лише до своєї верхівки. Він нагадує собою млинці які лежать один на одному. Щоб дістати один елемент, треба прибрати елементи які знаходяться вище нього.