

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**

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | Стек | Додавання, видалення, перевірка наявності хоча б одного елементу, перегляд першого елементу, визначення глибини стеку (після елементи мають лишитися в початковому стані) | Книга |

**Текст програми**

Main.java

package com.company;  
import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
  
 //стек для прикладу  
 Stack<Book> stack = new Stack<>(new Book("Книга 1", "Автор 1", 2000, 100, 100));  
 stack.push(new Book("Книга 2", "Автор 2", 2001, 200, 200));  
 stack.push(new Book("Книга 3", "Автор 3", 2002, 300, 300));  
 stack.push(new Book("Книга 4", "Автор 4", 2003, 400, 400));  
  
  
 int choice = 0;  
 Scanner input = new Scanner(System.in);  
 while (true) {  
 System.out.println("Виберіть дію:");  
 System.out.println("1. додати элемент в стек");  
 System.out.println("2. видалити елемент із стеку");  
 System.out.println("3. отримати верхній елемент стеку");  
 System.out.println("4. вивести всі элементи стеку");  
 System.out.println("5. вивести кількість елементів у стеку");  
 System.out.println("6. выход");  
 choice = input.nextInt();  
 switch (choice) {  
 case 1: {  
 System.out.print("Введіть назву книги: ");  
 String name = input.next();  
 System.out.print("Введіть автора книги: ");  
 String author = input.next();  
 System.out.print("Введіть рік видання книги: ");  
 int year = input.nextInt();  
 System.out.print("Введіть кількість сторінок книги: ");  
 int pages = input.nextInt();  
 System.out.print("Введіть ціну книги: ");  
 int price = input.nextInt();  
 stack.push(new Book(name, author, year, pages, price));  
 break;  
 }  
 case 2: {  
 Book book = stack.pop();  
 System.out.println(book);  
 break;  
 }  
 case 3: {  
 Book book = stack.peek();  
 System.out.println(book);  
 break;  
 }  
 case 4: {  
 stack.print();  
 break;  
 }  
 case 5: {  
 System.out.println(stack.getSize());  
 break;  
 }  
 case 6: {  
 System.exit(0);  
 }  
 }  
 }  
 }  
}

Stack.java

package com.company;  
  
public class Stack<T>  
{  
 private T value;  
 private Stack<T> next;  
  
 public Stack(T value)  
 {  
 this.value = value;  
 }  
  
 public void push(T value)  
 {  
 Stack<T> newNode = new Stack<T>(value);  
 newNode.next = this.next;  
 this.next = newNode;  
 }  
  
 public T pop()  
 {  
 if (this.next == null)  
 {  
 return null;  
 }  
 T value = this.next.value;  
 this.next = this.next.next;  
 return value;  
 }  
  
 public T peek()  
 {  
 if (this.next == null)  
 {  
 return null;  
 }  
 return this.next.value;  
 }  
 public void print()  
 {  
 Stack<T> current = this;  
 while (current != null)  
 {  
 System.out.println(current.value);  
 current = current.next;  
 }  
 }  
 public boolean isEmpty()  
 {  
 return this.next == null;  
 }  
  
 public int getSize()  
 {  
 int size = 0;  
 Stack<T> current = this;  
 while (current != null)  
 {  
 size++;  
 current = current.next;  
 }  
 return size;  
 }  
}

Book.java

package com.company;  
  
public class Book  
{  
 private String name;  
 private String author;  
 private int year;  
 private int pages;  
 private int price;  
 private int count;  
  
 public Book(String name, String author, int year, int pages, int price)  
 {  
 this.name = name;  
 this.author = author;  
 this.year = year;  
 this.pages = pages;  
 this.price = price;  
 }  
  
 public String getName()  
 {  
 return name;  
 }  
  
 public String getAuthor()  
 {  
 return author;  
 }  
  
 public int getYear()  
 {  
 return year;  
 }  
  
 public int getPages()  
 {  
 return pages;  
 }  
  
 public int getPrice()  
 {  
 return price;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
  
 public void setAuthor(String author)  
 {  
 this.author = author;  
 }  
  
 public void setYear(int year)  
 {  
 this.year = year;  
 }  
  
 public void setPages(int pages)  
 {  
 this.pages = pages;  
 }  
  
 public void setPrice(int price)  
 {  
 this.price = price;  
 }  
  
 public String toString()  
 {  
 return "Name: " + name + " Author: " + author + " Year: " + year + " Pages: " + pages + " Price: " + price + " Count: " + count;  
 }  
}

**Висновок**

Під час виконання лабораторної роботи реалізовув стек. Стек – це структура даних, в якій є доступ лише до верхнього елементу. Стек нагадує собою млинці або стопку з тарілками, для того щоб дістати елемент з середини потрібно прибрати всі елементи, які знаходяться вище нього.