****Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра автоматизації проектування енергетичних процесів і систем

ЗВІТ   
про виконання лабораторної роботи №4  
з дисципліни «Технології конструювання програмного забезпечення»

**«Разработка и тестирование программ в среде IntelliJ IDEA»**

Виконав: студент групи TI-92

Черноусов Денис Ігорович

Перевірив: доцент, к.ф.-м.н.  
Тарнавський Ю.А.

Київ - 2020

# Ход выполнения работы

1. В IntelliJ IDEA создайте проект **Maven**.
2. Создайте классы программ в соответствии с вариантом индивидуального задания. При этом в заданиях №1-15 не использовать массивы (если возможно), а воспользоваться подходящими классами коллекций!
3. Создайте unit-тесты для тестирования созданной программы.
4. Выполните тестирование программ.
5. Оформите отчет о проделанной работе.

**16 варіант**

5. Создайте программу для шифрования\расшифровки текста методом Цезаря. В нем ключом является целое число, а шифрование\расшифровка заключается в суммировании\ вычитании кодов символов открытого текста\криптотекста с ключом.

13.Создайте программу, которая осуществляет шифрование / расшифровка методом простой замены. В нем каждый символ незашифрованного текста из исходного алфавита заменяется другим из алфавита, символы в каком представлены в соответствии с ключом шифрования.

25. Создайте приложение для поиска изображения в списке изображений. Список (5 изображений) создается в программе и является объектом класса HashMap, где ключом является наименование изображения (типа String), а значением – изображение (объект класса Image). Предусмотреть возможность чтобы изображения добавлялись с уникальными именами с помощью TreeSet.

31. Создайте приложение для просмотра списка товаров и изменения цены товара в электронном магазине. Список товаров (5 записей) создается в программе и является объектом класса TreeMap. Ключом записи является артикул товара marking (типа Integer), а значением – объект Article, содержащий наименование товара (типа String) и цену товара (типа float). Предусмотреть возможность проверки уникальности наименования товара с помощью HashSet.

Завдання 5

import java.util.Stack;  
  
public class task5 {  
  
 public static Stack<Character> fillStack(Stack<Character> stk1 ){  
 for ( char a:"Hello, i'm 18 years old".toCharArray()) {  
 stk1.push(a);  
 }  
 return stk1;  
 }  
  
 public static void main(String[] args){  
  
 Stack<Character> stk1 = new Stack<>();  
 *fillStack*( stk1 );  
  
 Stack<Character> stk2 = new Stack<>();  
  
 int key = 15 ;  
  
 System.*out*.printf("Encrypted : %n");  
  
 *Caesar*( key , stk1 , stk2);  
  
 System.*out*.printf("%nDecrypted : %n");  
  
 *Caesar*( -key , stk2 , stk1);  
 }  
  
 public static Stack<Character> Caesar(int key , Stack<Character> stk1 , Stack<Character> stk2 ){  
  
  
 while(!stk1.empty()) System.*out*.print(stk2.push((char)(stk1.pop() + key)));  
  
  
 System.*out*.println();  
  
 return stk2 ;  
 }  
}

Вивід:

Encrypted :

s{~/pt/G@/|6x/;~{{tW

Decrypted :

Hello, i'm 18 years old

Завдання 13

import java.util.ArrayList;  
import java.util.TreeMap;  
  
import static java.lang.Character.\*;  
  
public class task13 {  
  
 public static void create(ArrayList<Character> abc , ArrayList<Character> key){  
 char[] abcStr = "abcdefghijklmnopqrstuvwxyz".toCharArray();  
 char[] keyStr = "zbyxvwutsrqponmlkjihgfedca".toCharArray();  
  
 for (int i = 0 ; i < 26 ; i++) {  
  
 abc.add(abcStr[i]);  
 key.add(keyStr[i]);  
  
 }  
 }  
  
  
 public static char[] changer ( char[] str , ArrayList<Character> abc , ArrayList<Character> key ) {  
  
  
 for ( int index, i = 0 ; i < str.length ; i++) {  
  
 if(!*isLetter*(str[i])) ;  
 else if(*isLowerCase*(str[i])) {  
  
 index = abc.indexOf(str[i]);  
 str[i] = key.get(index);  
  
 }else if(*isUpperCase*(str[i])){  
 index = abc.indexOf(*toLowerCase*(str[i]));  
 str[i] = *toUpperCase*(key.get(index));  
  
 }  
 System.*out*.print(str[i]);  
 }  
  
 return str;  
 }  
  
 public static void main(String[] args){  
  
 ArrayList<Character> abc = new ArrayList<>();  
 ArrayList<Character> key = new ArrayList<>();  
  
 *create*(abc, key);  
  
 char[] str = "I'm going to feel".toCharArray();  
  
 System.*out*.print("Encrypted : \n");  
 *changer*( str , abc , key );  
  
 System.*out*.print("\nDecrypted : \n");  
 *changer*( str , key , abc );  
  
 }  
}

Вивід :

Encrypted :

S'o umsnu hm wvvp

Decrypted :

I'm going to feel

Завдання 25

import javax.imageio.ImageIO;  
import javax.swing.\*;  
import java.awt.image.BufferedImage;  
import java.io.File;  
import java.io.IOException;  
import java.lang.reflect.Array;  
import java.util.\*;  
import java.awt.\*;  
  
import static javax.swing.WindowConstants.*DISPOSE\_ON\_CLOSE*;  
  
  
  
public class task25 {  
  
 public static ArrayList<String> toAlbum(HashMap<String,Image> album , TreeSet<File> treeAlbum )throws IOException{  
  
 Iterator i = treeAlbum.descendingIterator();  
 ArrayList<String> newStr = new ArrayList<>();  
  
 while(i.hasNext()){  
  
 File f = (File) i.next();  
  
 String picName = f.getName();  
  
 picName = picName.replace(".jpg", "");  
  
 boolean uniqKey = !album.containsKey(picName);  
  
 if (uniqKey) {  
  
 album.put(picName, ImageIO.*read*(f));  
 System.*out*.println(picName + " is new in the Gallery");  
  
 } else {  
 newStr.add(picName);  
 System.*out*.println(picName + " already is in the Gallery ");  
 }  
  
 }  
  
 return newStr;  
 }  
  
 public static HashMap<String,Image> Gallery(HashMap<String,Image> album)throws IOException {  
  
 BufferedImage LizaPic = ImageIO.*read*(new File("D:\\Objects\\JavaLab\\lab4\\src\\main\\25taskPicture\\Mona\_Lisa.jpg"));  
 album.put("Liza", LizaPic);  
  
 BufferedImage WalliPic = ImageIO.*read*(new File("D:\\Objects\\JavaLab\\lab4\\src\\main\\25taskPicture\\Walli.jpg"));  
 album.put("Walli", WalliPic);  
  
 BufferedImage ElonPic = ImageIO.*read*(new File("D:\\Objects\\JavaLab\\lab4\\src\\main\\25taskPicture\\Musk.jpg"));  
 album.put("Elon", ElonPic);  
  
 BufferedImage ScreamPic = ImageIO.*read*(new File("D:\\Objects\\JavaLab\\lab4\\src\\main\\25taskPicture\\Screamjpg.jpg"));  
 album.put("Scream", ScreamPic);  
  
 BufferedImage DaliPic = ImageIO.*read*(new File( "D:\\Objects\\JavaLab\\lab4\\src\\main\\25taskPicture\\Dali.jpg" ));  
 album.put( "Dali", DaliPic );  
  
 return album;  
 }  
  
 public static void FindPic(HashMap <String,Image> album) {  
  
  
 System.*out*.println("\n(type \"out\" to quit) Check our Gallery : ");  
  
 for (String art : album.keySet()) System.*out*.print(art + " ");  
 System.*out*.println();  
  
 Scanner scan = new Scanner(System.*in*);  
 String name;  
 name = scan.nextLine();  
  
 for(;;){  
  
 if(name.equals("out")) break ;  
  
 Image Im = album.get(name);  
  
 JLabel picLabel = new JLabel(new ImageIcon(Im));  
 JPanel jPanel = new JPanel();  
 jPanel.add(picLabel);  
  
 JFrame f = new JFrame();  
 f.setSize(new Dimension(Im.getWidth(null), Im.getHeight(null)));  
 f.add(jPanel);  
  
 f.setDefaultCloseOperation(*DISPOSE\_ON\_CLOSE*);  
 f.setVisible(true);  
 name = scan.nextLine();  
  
 }  
  
 }  
  
 public static TreeSet<File> FillTree(TreeSet<File> treeAlbum){  
  
 treeAlbum.add( new File ( "D:\\Objects\\JavaLab\\lab4\\src\\main\\25taskPicture\\Dali.jpg" ));  
 treeAlbum.add( new File ("D:\\Objects\\JavaLab\\lab4\\src\\main\\25taskPicture\\Sodom.jpg" ));  
  
 return treeAlbum ;  
 }  
  
  
 public static void main(String[] args)throws IOException{  
  
 HashMap <String,Image> album = new HashMap<>(5);  
 TreeSet<File> treeAlbum = new TreeSet<>();  
  
 *Gallery*(album);  
  
 *FillTree*(treeAlbum);  
  
 *toAlbum*(album , treeAlbum);  
  
 *FindPic*(album);  
  
 }  
}

Вивід:

Sodom is new in the Gallery

Dali already is in the Gallery

(type "out" to quit) Check our Gallery :

Liza Walli Dali Sodom Scream Elon

Liza

Out

Завдання 31

import java.lang.reflect.Array;  
import java.util.Collections;  
import java.util.HashSet;  
import java.util.Scanner;  
import java.util.TreeMap;  
  
public class task31 {  
  
 public static class Article{  
 public String name ;  
 public float price ;  
  
 public Article( String n , double p ){  
 name = n ;  
 price = (float)p ;  
 }  
  
 }  
  
 public static TreeMap <Integer, Article> Shop(TreeMap <Integer, Article> list){  
  
 list.put(1 , new Article( "Milk" , 10.2 ));  
 list.put(2 , new Article( "Latte" , 15.7 ));  
 list.put(3 , new Article( "Ring" , 100.5 ));  
 list.put(4 , new Article( "Bread" , 5.1 ));  
 list.put(5 , new Article( "Milk" , 500.2 ));  
  
 return list ;  
 }  
  
 public static void ShowList(TreeMap <Integer, Article> list){  
  
 for (int index : list.keySet() ) {  
 System.*out*.println( index + "." + list.get(index).name + " - " +list.get(index).price);  
 }  
 }  
  
 public static void ChangePrice(TreeMap <Integer, Article> list){  
  
 System.*out*.println("\n('0' to quit )Type index of good you want to change its price : ");  
  
 Scanner scan2 = new Scanner(System.*in*);  
 Scanner scan = new Scanner(System.*in*);  
  
 int index = scan.nextInt();  
  
 for ( ;index >= Collections.*min*(list.keySet()) && index <= Collections.*max*(list.keySet()) ; ){  
  
 System.*out*.println(index + "."+ list.get(index).name + " - " + list.get(index).price);  
 System.*out*.println("Type new price : ");  
  
 list.get(index).price = scan2.nextFloat(); // use ',' not '.'  
 System.*out*.println(index + "."+ list.get(index).name + " - " + list.get(index).price + "\n");  
  
  
 System.*out*.println("('0' to quit )Type index of good you want to change its price : ");  
 index = scan.nextInt();  
  
 }  
 }  
  
 public static TreeMap <Integer, Article> heshCheck(TreeMap <Integer, Article> list){  
  
 HashSet<String> hesh = new HashSet<>();  
  
 boolean isUnique ;  
  
 for (int index:list.keySet()) {  
 isUnique = hesh.add(list.get(index).name);  
 if(!isUnique){  
 System.*out*.println("\"" + list.remove(index).name + "\"" + " is not an unique good.\n" );  
 }  
  
 }  
 return list ;  
 }  
  
 public static void main(String [] args){  
  
 TreeMap <Integer, Article> list = new TreeMap<>();  
  
 *Shop*(list);  
  
 *ShowList*(list);  
  
 *ChangePrice*(list);  
  
 *heshCheck*(list);  
  
 *ShowList*(list);  
  
 }  
}

Вивід :

1.Milk - 10.2

2.Latte - 15.7

3.Ring - 100.5

4.Bread - 5.1

5.Milk - 500.2

('0' to quit )Type index of good you want to change its price :

1

1.Milk - 10.2

Type new price :

10,5

1.Milk - 10.5

('0' to quit )Type index of good you want to change its price :

0

"Milk" is not an unique good.

1.Milk - 10.5

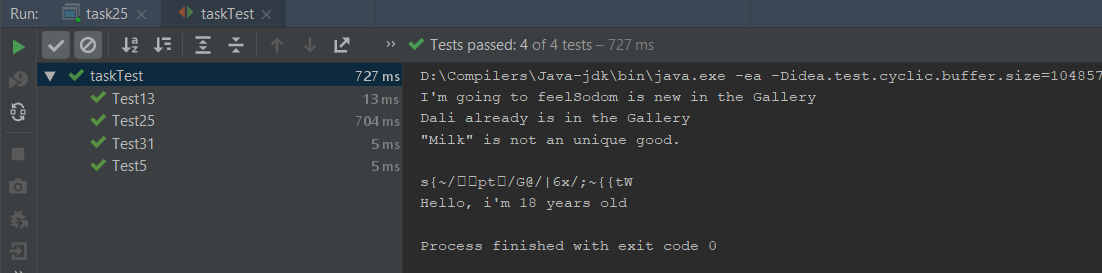
2.Latte - 15.7

3.Ring - 100.5

4.Bread - 5.1

**Unit тести**

import org.junit.Test;  
import java.io.IOException;  
import java.util.\*;  
  
import static org.junit.Assert.\*;  
  
public class taskTest {  
  
 @Test  
 public void Test25() throws IOException {  
 task25 test = new task25();  
  
 ArrayList<String> actuall = test.*toAlbum*(test.*Gallery*(new HashMap<>(5)) , test.*FillTree*(new TreeSet<>()));  
  
 ArrayList<String> expected = new ArrayList<>();  
 expected.add("Dali");  
  
 *assertEquals*( expected.toString() , actuall.toString() );  
  
 }  
  
 @Test  
 public void Test31() {  
 task31 test = new task31();  
  
 TreeMap<Integer, task31.Article> actuall = test.*heshCheck*( test.*Shop*(new TreeMap<>()) );  
  
 TreeMap<Integer, task31.Article> expected = test.*Shop*(new TreeMap<>()) ;  
 expected.remove(5);  
  
 *assertEquals*( expected.get(5) , actuall.get(5) );  
  
 }  
 @Test  
 public void Test5() {  
 task5 test = new task5();  
  
  
 String actuall = test.*Caesar*( -15 ,  
 test.*Caesar*( 15 , test.*fillStack*(new Stack<>()) , new Stack<>() ),  
 new Stack<>() ).toString();  
  
 String expected = test.*fillStack*(new Stack<>()).toString();  
  
 *assertEquals*( expected , actuall );  
  
 }  
  
 @Test  
 public void Test13() {  
 task13 test = new task13();  
  
 ArrayList<Character> abc = new ArrayList<>();  
 ArrayList<Character> key = new ArrayList<>();  
  
 test.*create*(abc, key);  
  
 char[] actuall = test.*changer*("S'o umsnu hm wvvp".toCharArray() , key , abc);  
 char[] expected = "I'm going to feel".toCharArray() ;  
  
 *assertArrayEquals*( expected , actuall );  
  
 }  
}

Вивід :