МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

***Циклова комісія програмування***

**Робочий план**

з курсу «Java - програмування»

Солом’яного Ярослава Сергійовича

*ПІБ студента*

студента групи **2П-18**

|  |  |  |
| --- | --- | --- |
| **Вид роботи** | **Дата** | **Оцінка** |
| *Пром. контроль* |  |  |
| *Залік* |  |  |

Викладач Марченко С. В.

Черкаси-2021

Завдання 1. Створіть список (спробуйте як ArrayList, так і LinkedList) та заповніть його назвами країн, введених з клавіатури. Відсортуйте список та виведіть його на екран. Далі кілька разів застосуйте до списку метод Collections.shuffle() та після кожного застосування виведіть результат.

Лістинг 1. Код виконання програми.

package ex\_1;  
import java.text.SimpleDateFormat;  
import java.util.\*;  
  
public class Test{  
 public static void main(String[] args)  
 {  
 //ArrayList = city  
 ArrayList city = new ArrayList();  
 city.add("Harkiv");  
 city.add("Cherkassy");  
 city.add("Kiev");  
 //Sort.  
 city.sort(Comparator.naturalOrder());  
 System.out.println("City після сортування.");  
 System.out.println(city);  
 System.out.println("Використання Collections.shuffle.");  
 Collections.shuffle(city);  
 System.out.println(city);  
 Collections.shuffle(city);  
 System.out.println(city);  
 Collections.shuffle(city);  
 System.out.println(city);  
 Collections.shuffle(city);  
 System.out.println(city);  
 //Linked List  
 LinkedList city\_linked = new LinkedList();  
 city\_linked.add("Житомир");  
 city\_linked.add("Донецьк");  
 city\_linked.add("Дніпро");  
 city\_linked.add("Одеса");  
 Collections.sort(city\_linked);  
 System.out.println("City після сортування.");  
 System.out.println(city\_linked);  
 System.out.println("Використання Collections.shuffle.");  
 Collections.shuffle(city\_linked);  
 System.out.println(city\_linked);  
 Collections.shuffle(city\_linked);  
 System.out.println(city\_linked);  
 Collections.shuffle(city\_linked);  
 System.out.println(city\_linked);  
 Collections.shuffle(city\_linked);  
 System.out.println(city\_linked);  
 printTimeStamp();  
 }  
 public static void printTimeStamp() {  
 final String sname = "Yaroslav Solomianyi";  
 Date currentDate = new Date();  
 SimpleDateFormat dateFormat = null;  
 dateFormat = new SimpleDateFormat();  
 System.out.println("Made by: " + sname + " on " + dateFormat.format(currentDate));  
 }  
}

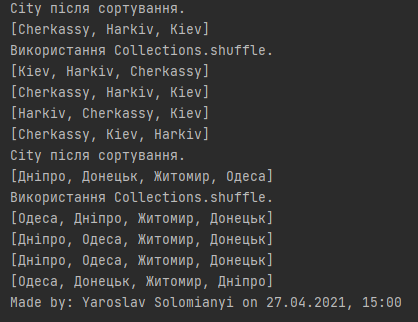


Рис 1. Результат роботи програми №1.

Завдання 2. Напишіть статичний метод reverseCopy(), який має аргумент List і повертає власноруч створену копію List з елементами в зворотному порядку (без використання вбудованого методу).

Лістинг 2. Код для виконання завдання

package Ex\_2;  
import java.lang.reflect.Array;  
import java.text.SimpleDateFormat;  
import java.util.\*;  
import java.lang.String;  
  
public class Test {  
 public static void main(String[] args)  
 {  
 List<String> city = new ArrayList<>();  
 city.add("Умань");  
 city.add("Ужгород");  
 city.add("Прип'ять");  
 city.add("Мадрид");  
 System.out.println("Список до сортування:");  
 for (String s: city) {  
 System.out.println(s);  
 }  
 System.out.println("Список після сортування:");  
 List<String> ReverseList = reverseCopy(city);  
 for (String s: ReverseList ) {  
 System.out.println(s);  
 }  
 printTimeStamp();  
 }  
 public static <T> List<T> reverseCopy(List<T> list)  
 {  
 List<T> tempList = new ArrayList<T>();  
 for(int i = list.size()-1;i >= 0;i--)  
 {  
 tempList.add(list.get(i));  
 }  
 return tempList;  
 }  
 public static void printTimeStamp() {  
 final String sname = "Yaroslav Solomianyi";  
 Date currentDate = new Date();  
 SimpleDateFormat dateFormat = null;  
 dateFormat = new SimpleDateFormat();  
 System.out.println("Made by: " + sname + " on " + dateFormat.format(currentDate));  
 }  
}

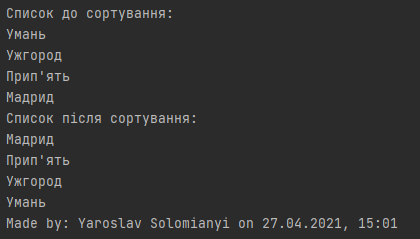


Рис 2. Результат роботи програми №2.

Завдання 3. (Сортування слів за допомогою TreeSet) Напишіть програму, яка використовує метод String.split() для відокремлення слів з введеного користувачем тексту, а потім розміщує їх у TreeSet. Виведіть елементи TreeSet на екран.

Лістинг 3. Код для виконання завдання №3.

package Ex\_3;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Scanner;  
import java.util.TreeSet;  
  
public class Test {  
 public static void main(String[] args)  
 {  
 System.out.println("Введіть текст:");  
 Scanner sc = new Scanner(System.in);  
 String text = sc.nextLine();  
 TreeSet<String> tree = new TreeSet<String>();  
 String[] splitText = text.split(" ");  
  
 for(int i = 0;i < splitText.length;i++)  
 {  
 tree.add(splitText[i]);  
 }  
 System.out.println(tree);  
 printTimeStamp();  
 }  
 public static void printTimeStamp() {  
 final String sname = "Yaroslav Solomianyi";  
 Date currentDate = new Date();  
 SimpleDateFormat dateFormat = null;  
 dateFormat = new SimpleDateFormat();  
 System.out.println("Made by: " + sname + " on " + dateFormat.format(currentDate));  
 }  
}

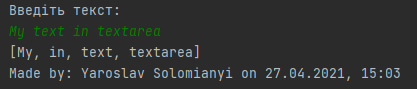


Рис 3. Результат роботи програми №3.

Завдання 4. (Хешовані структури) Реалізуйте додаток, який буде виконувати роль телефонної книги. Кожний запис телефонної книги містить номер телефону та назву контакту. Користувачеві мають бути доступними наступні можливості: додати контакт, видалити контакт, редагувати контакт, вивести список усіх контактів та вийти з додатку. Телефонна книга повинна мати при запуску контакти екстрених служб: «Пожежна безпека» - 101, «Поліція» - 102, «Швидка допомога» - 103. Книгу рекомендується тримати у вигляді мепу, а також в процесі реалізації видалення контакту додати перевірку на непорожність книги. Видалення контакту слід виконувати за назвою контакту.

Лістинг 4. Код для виконання завдання №4.

package Ex\_4;  
  
import java.text.SimpleDateFormat;  
import java.util.\*;  
  
public class Test  
{  
 public static void main(String[] args)  
 {  
 Map<Long,String> contacts = new HashMap<Long,String>();  
 contacts.put(101L,"Пожежна безпека");  
 contacts.put(102L,"Поліція");  
 contacts.put(103L,"Швидка допомога");  
 Scanner sc = new Scanner(System.in);  
 boolean point = true;  
 while (point)  
 {  
  
 String appText = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\* \n"+  
 "| \*\* | 1.Додати контакт в книгу \n" +  
 "| \*\* | 2.Видалити контакт з книги \n" +  
 "| \*\* | 3.Редагувати контакт \n" +  
 "| \*\* | 4.Показати всі контакти \n" +  
 "| \*\* | 5.Вийти з додатку \n" ;  
 System.out.println(appText);  
 int cmd = sc.nextInt();  
 switch (cmd)  
 {  
 case 1:{  
 System.out.println("Введіть номер телефону, який хочете добавити: ");  
 long number = sc.nextLong();  
 System.out.println("Введіть назву контакту");  
 String name = sc.useDelimiter("\n").next();  
 System.out.println("Введені дані: | Номер: " + number + " | Назва контакту: " + name);  
 contacts.put(number,name);  
 System.out.println("Введені дані збережено в телефону книгу!");  
 break;  
 }  
  
 case 2:{  
 if (contacts.size() > 0) {  
 System.out.println("Введіть назву контакту для видалення:");  
 String tempName = sc.useDelimiter("\n").next();  
 for (Map.Entry<Long, String> entry : contacts.entrySet()) {  
 if (entry.getValue().equals(tempName)) {  
 contacts.remove(entry.getKey());  
 System.out.println("Контакт видалено з ім'ям: " + tempName);  
 }  
 }  
 }  
 else  
 {  
 System.out.println("Телефонна книга пуста");  
 }  
 break;  
 }  
 case 3:{  
 System.out.println("Редагування контакту");  
 System.out.println("Введіть номер телефону для редагування контакту:");  
 long tempKey = sc.nextLong();  
 if(contacts.containsKey(tempKey))  
 {  
 System.out.println("Введіть нове значення для номеру телефону: " + tempKey);  
 String newName = sc.useDelimiter("\n").next();  
 contacts.replace(tempKey, newName);  
 System.out.print("Інформація користувача після редагування:");  
 System.out.println(contacts.get(tempKey));  
 break;  
 }  
 }  
 case 4:{  
 System.out.println("\*\*\*\*\*\*\*\* Список контактів вашої телефонної книги: \*\*\*\*\*\*\*\*");  
 for (Map.Entry entry : contacts.entrySet()) {  
 System.out.println("| \*\* |" + entry.getKey() + " - "  
 + entry.getValue());}  
 System.out.println("\n");  
 break;  
 }  
 case 5:{  
 System.out.println("Вихід з додатку");  
 point = false;  
 printTimeStamp();  
 break;  
 }  
 }  
 }  
 }  
 public static void printTimeStamp() {  
 final String sname = "Yaroslav Solomianyi";  
 Date currentDate = new Date();  
 SimpleDateFormat dateFormat = null;  
 dateFormat = new SimpleDateFormat();  
 System.out.println("Made by: " + sname + " on " + dateFormat.format(currentDate));  
 }  
}

Лістинг 5. Результат роботи програми програми №4.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
4  
\*\*\*\*\*\*\*\* Список контактів вашої телефонної книги: \*\*\*\*\*\*\*\*  
| \*\* |101 - Пожежна безпека  
| \*\* |102 - Поліція  
| \*\* |103 - Швидка допомога  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
1  
Введіть номер телефону, який хочете добавити:   
380956432785  
Введіть назву контакту  
АВТОМИЙКА ЧЕРКАСИ  
Введені дані: | Номер: 380956432785 | Назва контакту: АВТОМИЙКА ЧЕРКАСИ  
Введені дані збережено в телефону книгу!  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
1  
Введіть номер телефону, який хочете добавити:   
3804473263042  
Введіть назву контакту  
ЧДБК  
Введені дані: | Номер: 380473263042 | Назва контакту: ЧДБК  
Введені дані збережено в телефону книгу!  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
1  
Введіть номер телефону, який хочете добавити:   
380943214342  
Введіть назву контакту  
АЗС WOG  
Введені дані: | Номер: 380943214342 | Назва контакту: АЗС WOG  
Введені дані збережено в телефону книгу!  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
4  
\*\*\*\*\*\*\*\* Список контактів вашої телефонної книги: \*\*\*\*\*\*\*\*  
| \*\* |380943214342 - АЗС WOG  
| \*\* |380473263042 - ЧДБК  
| \*\* |101 - Пожежна безпека  
| \*\* |102 - Поліція  
| \*\* |103 - Швидка допомога  
| \*\* |380956432785 - АВТОМИЙКА ЧЕРКАСИ  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
3  
Редагування контакту  
Введіть номер телефону для редагування контакту:  
380956432785  
Введіть нове значення для номеру телефону: 380956432785  
АВТОМИЙКА БІЛЯ ЧЕРКАС  
Інформація користувача після редагування:АВТОМИЙКА БІЛЯ ЧЕРКАС  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
4  
\*\*\*\*\*\*\*\* Список контактів вашої телефонної книги: \*\*\*\*\*\*\*\*  
| \*\* |380943214342 - АЗС WOG  
| \*\* |380473263042 - ЧДБК  
| \*\* |101 - Пожежна безпека  
| \*\* |102 - Поліція  
| \*\* |103 - Швидка допомога  
| \*\* |380956432785 - АВТОМИЙКА БІЛЯ ЧЕРКАС  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
2  
Введіть назву контакту для видалення:  
АВТОМИЙКА ЧЕРКАСИ  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
2  
Введіть назву контакту для видалення:  
АВТОМИЙКА БІЛЯ ЧЕРКАС  
Контакт видалено з ім'ям: АВТОМИЙКА БІЛЯ ЧЕРКАС  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
4  
\*\*\*\*\*\*\*\* Список контактів вашої телефонної книги: \*\*\*\*\*\*\*\*  
| \*\* |380943214342 - АЗС WOG  
| \*\* |380473263042 - ЧДБК  
| \*\* |101 - Пожежна безпека  
| \*\* |102 - Поліція  
| \*\* |103 - Швидка допомога  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ТЕЛЕФОННА КНИГА \*\*\*\*\*\*\*\*\*\*\*\*\*   
| \*\* | 1.Додати контакт в книгу   
| \*\* | 2.Видалити контакт з книги   
| \*\* | 3.Редагувати контакт   
| \*\* | 4.Показати всі контакти   
| \*\* | 5.Вийти з додатку   
  
5  
Вихід з додатку  
Made by: Yaroslav Solomianyi on 26.04.2021, 16:59

Завдання 5. (Створення плейлісту на базі списку) Створіть плейліст, кожна пісня якого має назву, виконавця, альбом та тривалість. Плейліст дозволяє додавати та видаляти пісню, отримувати розмір та очищати його, а також виконувати форматований вивід композицій і повну тривалість його відтворення.

Лістинг 6. Class Song

package Ex\_5;  
  
public class Song {  
 String name;  
 String author;  
 String album;  
 double duration;  
  
 public Song(String name, String author, String album, double duration) {  
 this.name = name;  
 this.author = author;  
 this.album = album;  
 this.duration = duration;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public String getAuthor() {  
 return author;  
 }  
  
 public String getAlbum() {  
 return album;  
 }  
  
 public double getDuration() {  
 return duration;  
 }  
  
 @Override  
 public String toString() {  
 return "Song{" +  
 "name='" + name + '\'' +  
 ", author='" + author + '\'' +  
 ", album='" + album + '\'' +  
 ", duration=" + duration +  
 '}';  
 }  
}

Лістинг 7.Class Test

package Ex\_5;  
  
import java.text.SimpleDateFormat;  
import java.util.\*;  
  
public class Test {  
 public static void main(String[] args) {  
 Set<Song> myPlayer = new HashSet<Song>();  
 Scanner sc = new Scanner(System.in);  
 boolean point = true;  
 while (point = true)  
 {  
 System.out.println("========== ПЛЕЙЛИСТ =============\n" +  
 "1.Добавити пісню\n" +  
 "2.Видалити пісню\n" +  
 "3.Розмір плейлисту\n" +  
 "4.Очистка плейлисту\n" +  
 "5.Вивід плейлисту\n" +  
 "6.Тривалість відтворення плейлисту\n" +  
 "7. Вихід");  
 int cmd = sc.nextInt();  
 switch (cmd)  
 {  
 case 1:{  
 System.out.print("Введіть назву пісні:");  
 String newName = sc.useDelimiter("\n").next();  
 System.out.print("Введіть автора пісні:");  
 String newAuthor = sc.useDelimiter("\n").next();  
 System.out.print("Введіть альбом пісні:");  
 String newAlbum = sc.useDelimiter("\n").next();  
 System.out.print("Введіть тривалість пісні(Наприклад '2,45'):");  
 double newDuration = sc.nextDouble();  
 myPlayer.add(new Song(newName,newAuthor,newAlbum,newDuration));  
 break;  
 }  
 case 2:{  
 System.out.println("Введіть назву пісні для видалення:");  
 String newName = sc.useDelimiter("\n").next();  
 for (Song song : myPlayer) {  
 if (newName.equals(song.name))  
 {  
 myPlayer.remove(song);  
 System.out.println("Пісню з назвою: | "+song.getName()+ " | Успішно видалено!");  
 }  
 else  
 {  
 System.out.println("Пісню з таким іменем не знайдено в плейлисті!");  
 }  
 }  
 break;  
 }  
 case 3:{  
 System.out.println("Розмір плейлисту: " + myPlayer.size());  
 break;  
 }  
 case 4:{  
 myPlayer.clear();  
 System.out.println("Плейлист очищено.");  
 break;  
 }  
 case 5:{  
 if(myPlayer.isEmpty())  
 {  
 System.out.println("Плейлист пустий!");}  
 else {  
 for (Song s : myPlayer) {  
 System.out.println("| Назва: " + s.getName() + "| Aвтор: " + s.getAuthor() + "| Альбом: " + s.getAlbum() + "| Тривалість: " + s.getDuration());  
 }  
 break;  
 }  
 }  
 case 6:{  
 double time = 0;  
 for (Song s : myPlayer)  
 {  
 time += s.getDuration();  
 }  
 time = Math.round(time\*100)/100D;  
 String str = String.format("Повна тривалість відтворення плейлисту: " + time + " Минут.");  
 System.out.println(str);  
 break;  
 }  
 case 7:{  
 System.out.println("Виконується вихід з програми.");  
 point = false;  
 printTimeStamp();  
 break;  
 }  
 }  
 }  
 }  
 public static void printTimeStamp() {  
 final String sname = "Yaroslav Solomianyi";  
 Date currentDate = new Date();  
 SimpleDateFormat dateFormat = null;  
 dateFormat = new SimpleDateFormat();  
 System.out.println("Made by: " + sname + " on " + dateFormat.format(currentDate));  
 }  
}

Лістинг 8. Результат роботи програми №5.

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

1

Введіть назву пісні:Пьяная Ночь

Введіть автора пісні:Гудзон

Введіть альбом пісні:Ремикс

Введіть тривалість пісні(Наприклад '2,45'):2,11

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

5

| Назва: Пьяная Ночь| Aвтор: Гудзон| Альбом: Ремикс| Тривалість: 2.11

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

2

Введіть назву пісні для видалення:

Пьяная Ночь

Пісню з назвою: | Пьяная Ночь | Успішно видалено!

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

1

Введіть назву пісні:Да Да Да

Введіть автора пісні:Jarico

Введіть альбом пісні:Remix

Введіть тривалість пісні(Наприклад '2,45'):3,12

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

3

Розмір плейлисту: 1

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

1

Введіть назву пісні:Можно быть первым

Введіть автора пісні:Homie

Введіть альбом пісні:

Введіть тривалість пісні(Наприклад '2,45'):2,01

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

5

| Назва: Можно быть первым| Aвтор: Homie| Альбом: | Тривалість: 2.01

| Назва: Да Да Да| Aвтор: Jarico| Альбом: Remix| Тривалість: 3.12

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

3

Розмір плейлисту: 2

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

4

Плейлист очищено.

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

5

Плейлист пустий!

Повна тривалість відтворення плейлисту: 0.0 Минут.

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

7

Виконується вихід з програми.

Made by: Yaroslav Solomianyi on 27.04.2021, 15:19

========== ПЛЕЙЛИСТ =============

1.Добавити пісню

2.Видалити пісню

3.Розмір плейлисту

4.Очистка плейлисту

5.Вивід плейлисту

6.Тривалість відтворення плейлисту

7. Вихід

Завдання 6. У даному завданні потрібно сформувати частоти зустрічності кожного символу з тексту, створити деревовидну структуру, а потім виконати обхід дерева. Побудова дерева відбувається на основі черги з пріоритетами. Спочатку матимемо ліс із листків (для кожного символу з тексту). У результаті роботи програми відповідно до введеного тексту потрібно вивести таблицю символів, що зустрічаються в цьому тексті, їх частоти зустрічності (кількість згадувань у тексті), а також сформовані бінарні коди.

Лістинг 9. Код для виконання завдання №6.

package Ex\_6;  
  
import java.text.SimpleDateFormat;  
import java.util.\*;  
  
abstract class HuffmanTree implements Comparable<HuffmanTree> {  
 public final int frequency;  
 public HuffmanTree(int freq) { frequency = freq; }  
  
 public int compareTo(HuffmanTree tree) {  
 return frequency - tree.frequency;  
 }  
}  
  
class HuffmanLeaf extends HuffmanTree {  
 public final char value;  
 public HuffmanLeaf(int freq, char val) {  
 super(freq);  
 value = val;  
 }  
}  
  
class HuffmanNode extends HuffmanTree {  
 public final HuffmanTree left, right;  
 public HuffmanNode(HuffmanTree l, HuffmanTree r) {  
 super(l.frequency + r.frequency);  
 left = l;  
 right = r;  
 }  
}  
public class HuffmanCode {  
 public static HuffmanTree buildTree(int[] charFreqs) {  
 PriorityQueue<HuffmanTree> trees = new PriorityQueue<HuffmanTree>();  
 for (int i = 0; i < charFreqs.length; i++)  
 if (charFreqs[i] > 0)  
 trees.offer(new HuffmanLeaf(charFreqs[i], (char)i));  
  
 assert trees.size() > 0;  
 while (trees.size() > 1) {  
 HuffmanTree a = trees.poll();  
 HuffmanTree b = trees.poll();  
 trees.offer(new HuffmanNode(a, b));  
 }  
 return trees.poll();  
 }  
  
 public static void printCodes(HuffmanTree tree, StringBuffer prefix) {  
 assert tree != null;  
 if (tree instanceof HuffmanLeaf) {  
 HuffmanLeaf leaf = (HuffmanLeaf)tree;  
 System.out.println(leaf.value + "\t\t" + leaf.frequency + "\t\t\t" + prefix);  
 } else if (tree instanceof HuffmanNode) {  
 HuffmanNode node = (HuffmanNode)tree;  
 prefix.append('0');  
 printCodes(node.left, prefix);  
 prefix.deleteCharAt(prefix.length()-1);  
 prefix.append('1');  
 printCodes(node.right, prefix);  
 prefix.deleteCharAt(prefix.length()-1);  
 }  
 }  
  
 public static void main(String[] args) {  
 String test = "ABRAKADABRASIMSALABIM";  
 int[] charFreqs = new int[256];  
 for (char c : test.toCharArray())  
 charFreqs[c]++;  
 HuffmanTree tree = buildTree(charFreqs);  
 System.out.println("SYMBOL\tWEIGHT\tHUFFMAN CODE");  
 printCodes(tree, new StringBuffer());  
 printTimeStamp();  
 }  
 public static void printTimeStamp() {  
 final String sname = "Yaroslav Solomianyi";  
 Date currentDate = new Date();  
 SimpleDateFormat dateFormat = null;  
 dateFormat = new SimpleDateFormat();  
 System.out.println("Made by: " + sname + " on " + dateFormat.format(currentDate));  
 }  
}

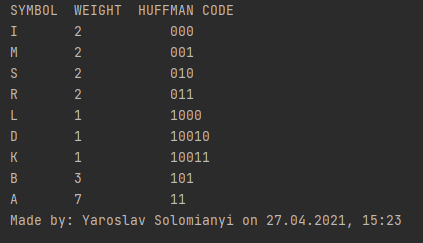


Рис 4. Результат роботи програми №6.