Phuong Ho

011056693

CECS277 – Lab 4

**public** **class** Employee **implements** Comparable

{

**private** String fName;

**private** String lName;

**private** **int** ID;

**private** String Scale;

Employee(){

fName = "";

lName = "";

ID = 0;

Scale = "";}

Employee(String first, String last, **int** id, String scale){

fName = first;

lName = last;

ID = id;

Scale = scale;}

**public** **void** setFirst(String first){

fName = first;}

**public** **void** setLast(String last){

lName = last;}

**public** **void** setID(**int** id){

ID = id;}

**public** **void** setScale(String scale){

Scale = scale;}

**public** String getFirst(){

**return** fName;}

**public** String getLast(){

**return** lName;}

**public** **int** getID(){

**return** ID;}

**public** String getScale(){

**return** Scale;}

**public** String toString(){

**return** (lName + ", " + fName + " - ID: " + ID + " - Scale: " + Scale);}

@Override

**public** **int** compareTo(Object o) {

Employee e = (Employee) o;

**if** (**this**.lName.equals(((Employee) o).lName)){

**if** (**this**.fName.equals(((Employee) o).fName)){

**return** (**this**.ID - e.getID());}

**else** **return** (**this**.fName.compareTo(((Employee) o).fName));}

**else** **return** (**this**.lName.compareTo(((Employee) o).lName));}

@Override

**public** **boolean** equals(Object o) {

Employee e = (Employee) o;

**if** ((**this**.lName == e.lName) && (**this**.fName == e.fName) && (**this**.ID == e.ID))

**return** **false**;

**return** **true**;}

**public** **int** hashCode(){

**int** h1 = **new** Double(ID).hashCode();

**int** h2 = fName.hashCode();

**int** h3 = lName.hashCode();

**return** h1 \* 12 + h2 \* 26 + h3;}

}

import java.util.Map;

import java.util.Set;

import java.util.TreeMap;

import java.util.TreeSet;

import java.util.Scanner;

public class Main

{

public static Set<Integer> generateWinningNumbers(){

int n = 0;

Set<Integer> winningNumbers = new TreeSet<Integer>();

for (int i = 0; i < 6; i++){

n = 1 + (int)(Math.random() \* ((40 - 1) + 1));

winningNumbers.add(n);}

return winningNumbers;}

public static Set<Integer> getTicket(){

Set<Integer> ticket = new TreeSet<Integer>();

Scanner input = new Scanner( System.in );

int t1;

int t2;

int t3;

int t4;

int t5;

int t6;

System.out.print("Type 6 lotto numbers: ");

t1 = input.nextInt();

t2 = input.nextInt();

t3 = input.nextInt();

t4 = input.nextInt();

t5 = input.nextInt();

t6 = input.nextInt();

ticket.add(t1);

ticket.add(t2);

ticket.add(t3);

ticket.add(t4);

ticket.add(t5);

ticket.add(t6);

return ticket;}

public static Map<Integer, Employee> employeesMap(){

Map<Integer, Employee> employees = new TreeMap<Integer, Employee>();

Employee e1 = new Employee("Puppy", "Nguyen", new Integer(345), "4");

employees.put(e1.getID(), e1);

Employee e2 = new Employee("Kitty", "Thompsons", new Integer(123), "2");

employees.put(e2.getID(), e2);

Employee e3 = new Employee("Cubby", "Gonzalez", new Integer(234), "5");

employees.put(e3.getID(), e3);

Employee e4 = new Employee("Monkey", "Smith", new Integer(456), "1");

employees.put(e4.getID(), e4);

return employees;}

public static Map<Employee, String> newEmployeeMap(){

Map<Employee, String> p = new TreeMap<Employee, String>();

Employee e1 = new Employee("Puppy", "Nguyen", new Integer(345), "4");

p.put(e1, e1.getScale());

Employee e2 = new Employee("Kitty", "Thompsons", new Integer(123), "2");

p.put(e2, e2.getScale());

Employee e3 = new Employee("Cubby", "Gonzalez", new Integer(234), "5");

p.put(e3, e3.getScale());

Employee e4 = new Employee("Monkey", "Smith", new Integer(456), "1");

p.put(e4, e4.getScale());

Employee e5 = new Employee("Yuppy", "Gonzalez", new Integer(789), "4");

p.put(e5, e5.getScale());

Employee e6 = new Employee("Cubby", "Gonzalez", new Integer(890), "3");

p.put(e6, e6.getScale());

return p;}

public static void main(String[] args) throws java.io.IOException{

System.out.println("PART 1\n");

Set<Integer> myTicket = getTicket();

Set<Integer> matches = new TreeSet<Integer>(myTicket);

Set<Integer> winning = generateWinningNumbers();

matches.retainAll(winning);

System.out.println("your ticket was: " + myTicket);

System.out.println("Winning numbers: " + winning);

System.out.println("Match numbers: " + matches);

if (matches.size() == 0){

System.out.println("You don't win this time. Good Luck for next time!");}

else if (matches.size() == 1){

System.out.println("Your prize is $100");}

else if (matches.size() == 2){

System.out.println("Your prize is $500");}

else if (matches.size() == 3){

System.out.println("Your prize is $700");}

else if (matches.size() == 4){

System.out.println("Your prize is $900");}

else if (matches.size() == 5){

System.out.println("Your prize is $1000");}

else System.out.println("Your prize is $1,000,000");

System.out.println("\n==========================================");

System.out.println("PART 2\n");

String userInput;

java.io.BufferedReader br = new java.io.BufferedReader(new java.io.InputStreamReader(System.in));

Scanner input = new Scanner( System.in );

Map<Integer, Employee> e = employeesMap();

Map<Employee, String> ne = newEmployeeMap();

do{

System.out.print("(a). Print the TreeMap<Integer, Employee> -- \n(b). Add an Employee into the TreeMap \n(c). Remove an Employee " + "\n(d). Edit Performance Scale of an Employee \n(e). Print the sorted TreeMap<Employee, String> \n==>Enter: " );

userInput = br.readLine();

switch (userInput.charAt(0)){

case 'a':

System.out.println("\nPRINTING ORIGINAL <Integer, Employee> TREEMAP: ");

for(Map.Entry<Integer, Employee> entry : e.entrySet())

System.out.println(entry.getValue());

System.out.println("-------\n");

break;

case 'b' :

String ln;

String fn;

int id;

String sc;

System.out.print("\nADDING -- Enter in order Last Name, First Name, ID and Performance Scale of an Employee you want to add: ");

ln = input.next();

fn = input.next();

id = input.nextInt();

sc = input.next();

Employee addE = new Employee(fn, ln, id, sc);

e.put(addE.getID(), addE);

System.out.println("\n....");

for(Map.Entry<Integer, Employee> entry : e.entrySet())

System.out.println(entry.getValue());

System.out.println("-------\n");

break;

case 'c' :

System.out.print("\n REMOVING -- Enter ID of the Employee you want to remove: ");

id = input.nextInt();

e.remove(id);

System.out.println("\n....");

for(Map.Entry<Integer, Employee> entry : e.entrySet())

System.out.println(entry.getValue());

System.out.println("-------\n");

break;

case 'd' :

System.out.print("\nEDITTING SCALE -- Enter ID of the Employee you want to have scale editted: ");

id = input.nextInt();

System.out.print("Enter the new scale: ");

sc = input.next();

Employee scaleEdit = e.get(id);

e.remove(id);

scaleEdit.setScale(sc);

e.put(scaleEdit.getID(), scaleEdit);

System.out.println("\n....");

for(Map.Entry<Integer, Employee> entry : e.entrySet()){

System.out.println(entry.getValue());

System.out.println("-------\n");

break;

case 'e' :

System.out.println("\nPRINTING THE SORTED <Employee, String>TREEMAP: ");

for(Map.Entry<Employee, String> entry : ne.entrySet())

System.out.println(entry.getKey());

System.out.println("-------\n");

break;

default :

break;}

} while (userInput.charAt(0) != 'q');

}

}