|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №5-6**

Выполнил студент группы ИНБО-02-20 Судариков Д.A.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

Решить задачи с помощью рекурсии

Целью данной практической работы освоить на практике сортировки различными методами

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.lab.lab5;  
  
import java.util.Scanner;  
public class Recursion {  
  
 String kandk(int k) {  
 int sum = 0;  
 int j = 0;  
 if (k == 1) {  
 System.*out*.print("1");  
 } else {  
 for (int i = 1; sum < k; i++) {  
 sum += i;  
 j = i;  
 }  
 System.*out*.print(kandk(--k) + " " + j);  
 }  
 return "";  
 }  
  
 String from1ton(int k) {  
 if (k == 1) {  
 return "1";  
 }  
 return from1ton(k - 1) + " " + k;  
 }  
  
 String atob(int a, int b) {  
 if (a > b) {  
 if (a == b) {  
 return Integer.*toString*(a);  
 }  
 return a + " " + atob(a - 1, b);  
 } else {  
 if (a == b) {  
 return Integer.*toString*(a);  
 }  
 return a + " " + atob(a + 1, b);  
 }  
 }  
  
 int sumnums(int len, int sum, int k, int s) {  
 if (len == k) {  
 if (sum == s) {  
 return 1;  
 } else {  
 return 0;  
 }  
 }  
 int c = (len == 0 ? 1 : 0);  
 int res = 0;  
 for (int i = c; i < 10; i++) {  
 res += sumnums(len + 1, sum + i, k, s);  
 }  
 return res;  
 }  
  
 int sumnumsdig(int n) {  
 if (n < 10)  
 return n;  
 else {  
 return n % 10 + sumnumsdig(n / 10);  
 }  
 }  
  
 boolean primenum(int n, int i) {  
 if (n < 2)  
 return false;  
 else if (n == 2)  
 return true;  
 else if (n % i == 0)  
 return false;  
 else if (i < n / 2)  
 return primenum(n, ++i);  
 else return true;  
 }  
  
 void factoriz(int n, int k) {  
 if (k > n / 2) {  
 System.*out*.println(n);  
 return;  
 }  
 if (n % k == 0) {  
 System.*out*.println(k);  
 factoriz(n / k, k);  
 } else {  
 factoriz(n, ++k);  
 }  
 }  
  
 String polindrom(String s) {  
 if (s.length() == 1) {  
 return "YES";  
 } else {  
 if (s.substring(0, 1).equals(s.substring(s.length() - 1, s.length()))) {  
 if (s.length() == 2) {  
 return "YES";  
 }  
 return polindrom(s.substring(1, s.length() - 1));  
 } else {  
 return "NO";  
 }  
 }  
 }  
  
 int withouttwozero(int a, int b) {  
 if (a > b + 1)  
 return 0;  
 if (a == 0 || b == 0)  
 return 1;  
 return withouttwozero(a, b - 1) + withouttwozero(a - 1, b - 1);  
 }  
  
 int turnnum(int n, int i) {  
 if (n == 0)  
 return i;  
 else return turnnum(n / 10, i \* 10 + n % 10);  
 }  
  
 int countunit() {  
 Scanner in = new Scanner(System.*in*);  
 int n = in.nextInt();  
 if (n != 0) {  
 int m = in.nextInt();  
 if (m != 0) {  
 return (n == 1 && m == 1) ? countunit() + 2 : (n != 1 && m != 1) ? countunit() : countunit() + 1;  
 } else {  
 int k = in.nextInt();  
 if (k != 0) {  
 return (n == 1 && k == 1) ? countunit() + 2 : (n != 1 && k != 1) ? countunit() : countunit() + 1;  
 } else {  
 return 1;  
 }  
 }  
 } else {  
 int m = in.nextInt();  
 if (m != 0) {  
 return (m == 1) ? countunit() + 1 : countunit();  
 } else {  
 return 0;  
 }  
 }  
 }  
  
 String unevennums() {  
 Scanner in = new Scanner(System.*in*);  
 int n = in.nextInt();  
 if (n != 0) {  
 return (n % 2 == 1) ? Integer.*toString*(n) + " " + unevennums() : unevennums();  
 } else {  
 return "";  
 }  
 }  
  
 String unevennum(int i) {  
 Scanner in = new Scanner(System.*in*);  
 int n = in.nextInt();  
 if (n != 0) {  
 return (i % 2 == 1) ? Integer.*toString*(n) + " " + unevennum(i + 1) : unevennum(i + 1);  
 } else {  
 return "";  
 }  
 }  
  
 String frltor(int n) {  
 if (n != 0) {  
 return frltor(n / 10) + " " + Integer.*toString*(n % 10);  
 } else {  
 return "";  
 }  
 }  
  
 String frrtol(int n) {  
 if (n != 0) {  
 return Integer.*toString*(n % 10) + " " + frrtol(n / 10);  
 } else {  
 return "";  
 }  
 }  
  
 int countmaxnums(int max, int count) {  
 Scanner in = new Scanner(System.*in*);  
 int n = in.nextInt();  
 if (n != 0) {  
 return (n > max) ? countmaxnums(n, ++count) : (n == max) ? countmaxnums(max, ++count) : countmaxnums(max, count);  
 } else {  
 return count;  
 }  
 }  
  
 int findmax(int max) {  
 Scanner in = new Scanner(System.*in*);  
 int n = in.nextInt();  
 if (n != 0) {  
 return (n > max) ? findmax(n) : findmax(max);  
 } else {  
 return max;  
 }  
 }  
  
 void printChoice() {  
 System.*out*.println("What do u want to use: ");  
 System.*out*.println("1) Треугольная последовательность;\n2) От 1 до n;\n3) От А до В;");  
 System.*out*.println("4) Заданная сумма цифр;\n5) Сумма цифр числа;\n6) Проверка числа на простоту;");  
 System.*out*.println("7) Разложение на множители;\n8) Палиндром\n9) Без двух нулей;");  
 System.*out*.println("10) Разворот числа;\n11) Количество единиц;\n12) Вывести нечётные числа последовательности;");  
 System.*out*.println("13) Вывести члены последовательности с нечётными номерами;\n14) Цифры числа слева направо;");  
 System.*out*.println("15) Цифры числа справо налево;\n16) Количество элементов равных максимуму;");  
 System.*out*.println("17) Максимум последовательности.");  
 }  
  
 boolean choice(int n) {  
 Scanner in = new Scanner(System.*in*);  
 int k, m;  
 String s;  
 switch (n) {  
 case 1:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 System.*out*.print("Ваша последовательность: ");  
 kandk(k);  
 System.*out*.println("");  
 break;  
 case 2:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 System.*out*.println("Ваша последовательность: " + from1ton(k));  
 break;  
 case 3:  
 System.*out*.println("Введите 2 числа: ");  
 k = in.nextInt();  
 m = in.nextInt();  
 System.*out*.println("Ваша последовательность: " + atob(k, m));  
 break;  
 case 4:  
 System.*out*.println("Введите 2 числа: ");  
 k = in.nextInt();  
 m = in.nextInt();  
 System.*out*.println(sumnums(1, 0, k, m));  
 break;  
 case 5:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 System.*out*.println(sumnumsdig(k));  
 break;  
 case 6:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 if (primenum(k, 2))  
 System.*out*.println("YES");  
 else System.*out*.println("NO");  
 break;  
 case 7:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 System.*out*.println("Множители числа: ");  
 factoriz(k, 2);  
 break;  
 case 8:  
 System.*out*.println("Введите слово: ");  
 s = in.next();  
 System.*out*.println(polindrom(s));  
 break;  
 case 9:  
 System.*out*.println("Введите 2 числа: ");  
 k = in.nextInt();  
 m = in.nextInt();  
 System.*out*.println(withouttwozero(k, m));  
 break;  
 case 10:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 System.*out*.println("Ваше число" + turnnum(k, 0));  
 break;  
 case 11:  
 System.*out*.println("Введите последовательность: ");  
 System.*out*.println("Количество единиц: " + countunit());  
 break;  
 case 12:  
 System.*out*.println("Введите последовательность: ");  
 System.*out*.println("Ваша последовательность: " + unevennums());  
 break;  
 case 13:  
 System.*out*.println("Введите последовательность: ");  
 System.*out*.println("Ваша последовательность: " + unevennum(1));  
 break;  
 case 14:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 System.*out*.println("Ваша последовательность: " + frltor(k));  
 break;  
 case 15:  
 System.*out*.println("Введите число: ");  
 k = in.nextInt();  
 System.*out*.println("Ваша последовательность: " + frrtol(k));  
 break;  
 case 16:  
 System.*out*.println("Введите последовательность: ");  
 k = in.nextInt();  
 System.*out*.println("Количество чисел равных максимуму: " + countmaxnums(k, 0));  
 break;  
 case 17:  
 System.*out*.println("Введите последовательность: ");  
 k = in.nextInt();  
 System.*out*.println("Ваш максимум: " + findmax(k));  
 break;  
 default:  
 System.*out*.println("Такой задачи нет!!!\n");  
 break;  
 }  
 System.*out*.println("Хотите ли вы продолжить просмотр задач? 1 - да, 0 - нет");  
 k = in.nextInt();  
 if (k == 0)  
 return false;  
 else return true;  
 }  
  
}

package ru.mirea.lab.lab5;  
  
import java.util.Scanner;  
  
public class TestRecursion {  
 public static void main(String[] args){  
 Recursion rec = new Recursion();  
 rec.printChoice();  
 Scanner in = new Scanner(System.*in*);  
 int k = in.nextInt();  
 while (rec.choice(k)){  
 rec.printChoice();  
 k = in.nextInt();  
 }  
 }  
}

**6 практика.** В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.lab.lab6.ex1;  
  
public class Student {  
 public int idNumber[];  
  
 public Student() {  
 idNumber = new int[6];  
 int a=0;  
 int b=100;  
 for (int i =0; i <idNumber.length;i++){  
 idNumber[i] = a+ (int) (Math.*random*()\*b);  
 }  
 }  
 public void selectionSort() {  
 for (int j = 1; j < idNumber.length; j++) {  
 int current = idNumber[j];  
 int i = j - 1;  
 while ((i > -1) && (idNumber[i] > current)) {  
 idNumber[i + 1] = idNumber[i];  
 i--;  
 }  
 idNumber[i + 1] = current;  
  
 } }  
 public void PrintScr() {  
 for (int aIdNumber : idNumber) {  
 System.*out*.println(aIdNumber);  
 }  
 }  
  
  
}

package ru.mirea.lab.lab6.ex1;  
  
import ru.mirea.lab.lab6.ex1.Student;  
  
public class TestStudent {  
 public static void main(String[] args) {  
 Student student = new Student();  
 student.selectionSort();  
 student.PrintScr();  
 }  
}

package ru.mirea.lab.lab6.ex2;  
  
import java.util.Arrays;  
import java.util.Comparator;  
  
  
public class SortingStudentsByGPA extends Student1 {  
 public static void main(String[] args) {  
 Student1[] students = new Student1[5];  
 Comparator<Student1> scomp = new StudentGPAComparator();  
 students[0] = new Student1("Pasha ", 245);  
 students[1] = new Student1("Masha ", 223);  
 students[2] = new Student1("Kolya ", 221);  
 students[3] = new Student1("Petya ", 290);  
 students[4] = new Student1("Andrey ", 255);  
 Arrays.*sort*(students, scomp);  
 for (int i = 0; i < 5; i++) {  
 System.*out*.println(students[i].getName() + " " + students[i].getGPA());  
 }  
 }  
  
 static class StudentGPAComparator implements Comparator<Student1> {  
 public int compare(Student1 a, Student1 b) {  
 if (a.getGPA() > b.getGPA())  
 return 1;  
 else if (a.getGPA() < b.getGPA())  
 return -1;  
 else return 0;  
 }  
 }  
}

package ru.mirea.lab.lab6.ex2;  
  
import java.util.Arrays;  
public class Student1 {  
 private String name;  
 private int GPA;  
  
 public Student1(String name, int GPA) {  
 this.name=name;  
 this.GPA=GPA;  
 }  
  
 public Student1() {  
  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setname(String name) {  
 this.name = name;  
 }  
  
 public int getGPA() {  
 return GPA;  
 }  
  
 public void setGPA(int GPA) {  
 this.GPA = GPA;  
 }  
}

package ru.mirea.lab.lab6.ex3;  
  
public class MergeSort {  
 public static void mergeSort(int[] a,int left,int right){  
 if(right<left)  
 return;  
 int mid=(left+right)/2;  
 *mergeSort*(a,left,mid);  
 *mergeSort*(a,mid+1,right);  
 *merge*(a,left,mid,right);  
 }  
 public static void merge(int[]a,int left,int mid,int right){  
 int lenghtleft=mid-left+1;  
 int lenghtright=right-mid;  
  
 int leftAr[]=new int [lenghtleft];  
 int rightAr[]=new int [lenghtright];  
  
 for(int i=0;i<lenghtleft;i++){  
 leftAr[i]=a[left+i];  
 }  
 for(int i=0;i<lenghtright;i++){  
 rightAr[i]=a[mid+i+1];  
 }  
 }  
}

## **Вывод**

В данной практической работе, мной были получены знания по работе с рекурсией, я познакомился с ней и выполнил практику. Также мной были получены знания по работе с сортировками различными методами.