|  |
| --- |
|  |
| 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 Самойленко М. А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2020г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2020г.

Отметка о выполнении

**Москва – 2020 г.**

**Практическое занятие №5**

**Цель работы:** Изучение работы с рекурсией.

**Задание.**

Дано слово, состоящее только из строчных латинских букв. Проверьте, является

ли это слово палиндромом. Выведите YES или NO.

При решении этой задачи нельзя пользоваться циклами, в решениях на питоне

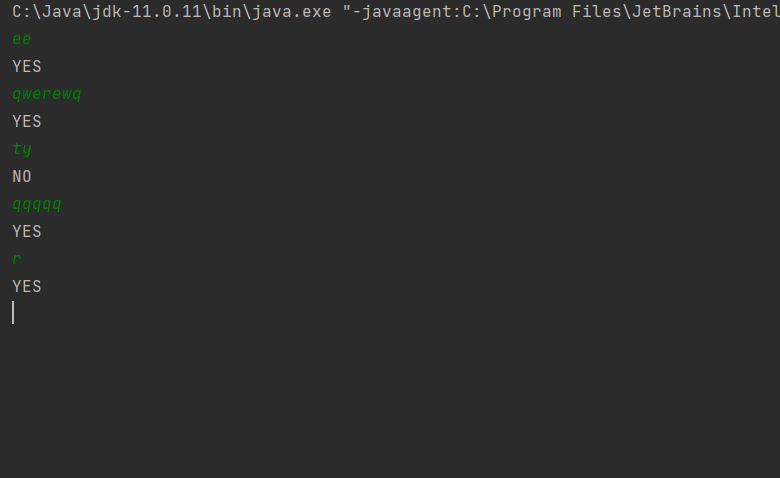
нельзя использовать срезы с шагом, отличным от 1.

**Код программы:**

**Класс Main**

public class Main {  
  
 public static void rekursiya(String wrld) {  
 if (wrld.length() == 1 || wrld.length() == 2 && wrld.charAt(0) == wrld.charAt(wrld.length()-1) ) {  
 System.*out*.println("YES");  
 }  
 else if (wrld.charAt(0) == wrld.charAt(wrld.length() - 1)) {  
 *rekursiya*(wrld.substring(1,wrld.length()-1));  
 }  
 else System.*out*.println("NO");  
 }  
 public static void main(String[] args) {  
 String k = " ";  
 Scanner in = new Scanner(System.*in*);  
 while (k != "0") {  
 k = in.next();  
 *rekursiya*(k);  
 }  
 }  
}

**Тест**



**Практическое занятие №6**

**Цель работы:**

Целью данной практической работы освоить на практике сортировки различными методами.

**Задание 1.**

Написать тестовый класс, который создает массив класса Student и сортирует

массив iDNumber.

**Код программы:**

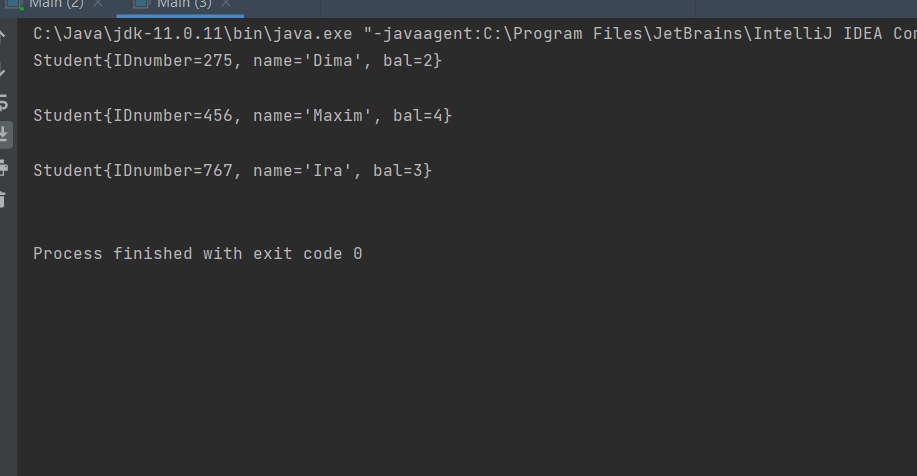
**Класс Main**

public class Main {  
 public static void insertionSort(Student[] arr) {  
 for (int i = 1; i < arr.length; i++) {  
 Student currElem = arr[i];  
 int prevKey = i - 1;  
 while (prevKey >= 0 && arr[prevKey].getIDnumber() > currElem.getIDnumber()) {  
 arr[prevKey + 1] = arr[prevKey];  
 prevKey--;  
 }  
 arr[prevKey + 1] = currElem;  
 }  
 }  
  
 public static void main(String[] args) {  
 Student[] students = new Student[3];  
 students[0] = new Student(456, "Maxim",4);  
 students[1] = new Student(275, "Dima",2);  
 students[2] = new Student(767, "Ira",3);  
 *insertionSort*(students);  
 for (Student s : students) {  
 System.*out*.println(s.toString());  
 }

}

}

**Тест**



**Задание 2 и задание 3.**

Напишите класс SortingStudentsByGPA который реализует интерфейс

Comparator таким образом, что она сортирует студентов с их итоговым баллом в

порядке убывания.

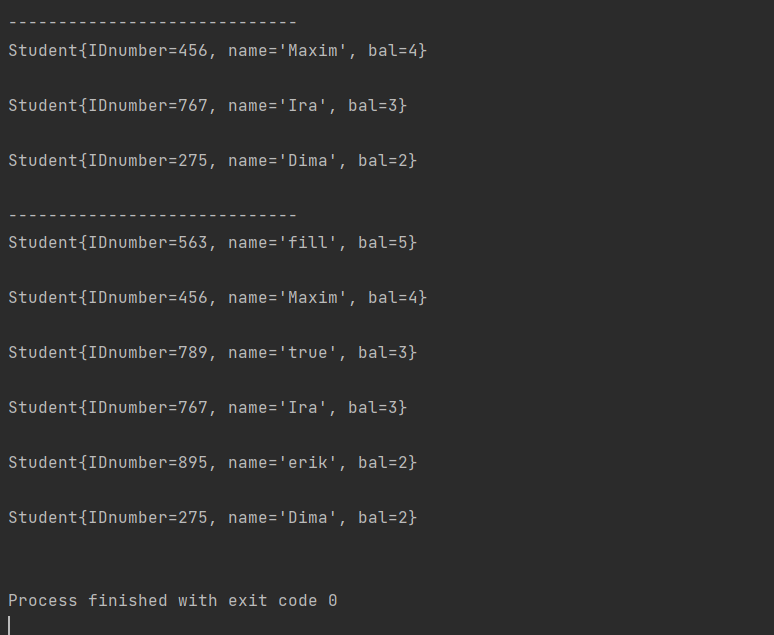
Напишите программу, которая объединяет два списка данных о студентах в один отсортированный списках.

**Код Программы:**

**Класс Main**

public class Main {  
 public static void insertionSort(Student[] arr) {  
 for (int i = 1; i < arr.length; i++) {  
 Student currElem = arr[i];  
 int prevKey = i - 1;  
 while (prevKey >= 0 && arr[prevKey].getIDnumber() > currElem.getIDnumber()) {  
 arr[prevKey + 1] = arr[prevKey];  
 prevKey--;  
 }  
 arr[prevKey + 1] = currElem;  
 }  
 }  
  
 public static void main(String[] args) {  
 Student[] students = new Student[3];  
 students[0] = new Student(456, "Maxim",4);  
 students[1] = new Student(275, "Dima",2);  
 students[2] = new Student(767, "Ira",3);  
 *insertionSort*(students);  
 for (Student s : students) {  
 System.*out*.println(s.toString());  
 }  
 System.*out*.println("-----------------------------");  
 List<Student> one = new ArrayList<>();  
 one.add(students[0]);  
 one.add(students[1]);  
 one.add(students[2]);  
 Comparator<Student> k = new SortingStudentsByGPA();  
 one.sort(k);  
 for (Student s : one) {  
 System.*out*.println(s.toString());  
 }  
 System.*out*.println("-----------------------------");  
 List<Student> two = new ArrayList<>();  
 two.add(new Student(789,"true",3));  
 two.add(new Student(563,"fill",5));  
 two.add(new Student(895,"erik",2));  
 two.sort(k);  
 List<Student> three = new ArrayList<>();  
 int posA = 0, posB = 0;  
 for (int i = 0; i < one.size() + two.size(); i++){  
 if (posA == one.size()){  
 three.add(two.get(posB));  
 posB++;  
 } else if (posB == two.size()) {  
 three.add(one.get(posA));  
 posA++;  
 } else if (one.get(posA).getBal() > two.get(posB).getBal()) {  
 three.add(one.get(posA));  
 posA++;  
 } else {  
 three.add(two.get(posB));  
 posB++;  
 }  
 }  
 for(Student s: three) {  
 System.*out*.println(s.toString());  
 }  
  
 }  
}

**Тест**



**Выводы**

Я научился работать с рекурсией, также освоил на практике различные методы сортировки.