

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА - Российский технологический университет»**

**РТУ МИРЭА**

Институт Информационных технологий

Кафедра математического обеспечения и стандартизации

информационных технологий

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

по дисциплине «Объектно-ориентированное программирование»

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-12-17 | А.А. Лисовой |

|  |  |  |
| --- | --- | --- |
| Принял | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность, звание, ученая степень) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (И.О. Фамилия) |

|  |  |  |
| --- | --- | --- |
| Работа выполнена | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Подпись студента) | «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 г. |

|  |  |  |
| --- | --- | --- |
| Работа зачтена | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Подпись преподавателя) | «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 г. |

Москва 2018

**Постановка задачи**

Дан код основного класса приложения на языке Java. Считать, что в коде нет синтаксических ошибок. Определить, сколько раз в программе создавались объекты. Сформировать список объектов (строку), указав название класса этого объекта. В какие переменные кода осуществлялся ввод с клавиатуры. Сколько в программе использовалось констант.

**Программная реализация**

|  |
| --- |
| package Lab\_2;  // Импорт класса  import java.util.Scanner;  // Основной класс  public class Lab\_2\_1  {  // Поля  public static String code = ""; // Код для обработки  public static String new\_string = " = new"; // Текст для поиска  объектов  public static String str\_string = "String"; //Текст для поиска строк  public static String final\_string = " final "; // Строка, содержащая  оператор объявления константы  public static String sc\_next\_string = ".next"; // Строка, содержащая  часть метода ввода  // Главный метод  public static void main(String[] args)  {  int command = 0;  Scanner input = new Scanner(System.in);  // Вывод списка опций  outputMenu();  do  {  System.out.print("Введите команду: ");  // Проверка ввода переменной выбора  while (!input.hasNextInt())  {  System.out.println("Неверно введена команда");  System.out.println();  System.out.print("Введите команду: ");  input.next();  }  command = input.nextInt();  // Обработка выбора  switch (command)  {  case 1:  // Ввести текст  input\_code();  break;  case 2:  // Определить, сколько раз в программе создавались  объекты  new\_count();  break;  case 3:  // Сформировать список объектов (строку), указав  название класса этого объекта  objects\_string();  break;  case 4:  // В какие переменные кода осуществлялся ввод с  клавиатуры  input\_values();  break;  case 5:  // Сколько в программе использовалось констант  final\_count();  break;  case 6:  // Повторный вывод меню  outputMenu();  break;  case 0:  System.out.println("Программа завершена");  break;  default:  System.out.println("Неверно введена команда");  System.out.println();  break;  }  }  while (command != 0);  }  // Метод вывода меню  public static void outputMenu()  {  System.out.println();  System.out.println("Меню программы:");  System.out.println("1. Ввести код");  System.out.println("2. Определить, сколько раз в программе  создавались объекты");  System.out.println("3. Сформировать список объектов (строку),  указав название класса этого объекта");  System.out.println("4. В какие переменные кода осуществлялся ввод с  клавиатуры");  System.out.println("5. Сколько в программе использовалось  констант");  System.out.println("6. Повторный вывод меню");  System.out.println("0. Завершение программы");  System.out.println();  }  // Метод ввода кода  public static void input\_code()  {  System.out.println("Введите код:");  Scanner input = new Scanner(System.in);  code = input.nextLine(); // Ввод кода  code = " " + code + " ";  System.out.println("Код введен");  System.out.println();  }  // Метод подсчёта количества созданий объектов:  public static void new\_count()  {  // Проверка введенности кода  if (code == "")  {  System.out.println("Ошибка: не был введен код");  System.out.println();  }  else  {  int count\_new = 0; // Счётчик созданных объектов  int count\_str = 0; // Счётчик созднных строк  // Опеределение индекса первого вхождения указанной подстроки  int position\_new = code.indexOf(new\_string);  int position\_str = code.indexOf(str\_string);  //Проверка на наличие хотябы одного вхождения  if (position\_new != -1 || position\_str != -1)  {  // Цикл поиска вхождений new в строку  while (position\_new != -1)  {  position\_new = code.indexOf(new\_string, position\_new +  new\_string.length());  count\_new++;  }  // Цикл поиска вхождений String в строку  while( position\_str != -1)  {  // Проверка на пробел после объявления строки  if (code.charAt(position\_str + str\_string.length()) ==  ' ')  {  count\_str++;  }  position\_str = code.indexOf(str\_string, position\_str +  str\_string.length());  }  System.out.println("В исходном коде было создано " +  (count\_new + count\_str) + " объектов");  System.out.println();  }  else  {  System.out.println("В исходном коде не было встречено ни  одного объекта");  System.out.println();  }  }  }  // Метод подсчёта количества констант:  public static void final\_count()  {  // Проверка введенности текста  if (code == "")  {  System.out.println("Ошибка: не был введен код");  System.out.println();  }  else  {  int count\_final = 0; // Счётчик созднных объектов  // Опеределение индекса первого вхождения указанной подстроки  int position\_final = code.indexOf(final\_string);  //Проверка на наличие хотябы одного вхождения  if (position\_final != -1)  {  // Цикл поиска вхождений констант в строку  while (position\_final != -1)  {  position\_final = code.indexOf(final\_string,  position\_final + final\_string.length());  count\_final++;  }  System.out.println("В исходном коде было создано " +  (count\_final) + " констант");  }  else  {  System.out.println("В исходном коде не было встречено ни  одной константы");  }  }  }  // Метод составления списка(строки) используемых объектов:  public static void objects\_string()  {  // Проверка введенности кода  if (code == "")  {  System.out.println("Ошибка: не был введен код");  System.out.println();  }  else  {  int current\_position = 0; // Текущая позиция искомого элемента  int buffer\_position = 0; // Буферная позиция  String current\_object = ""; // Текущее имя объекта  String current\_object\_type = ""; // Текущий тип объекта  String objects\_string = "Список объектов:"; // Строка с списком  найденных объектов  int start\_length = objects\_string.length(); // Изначальная  длина строки объектов  // Цикл поиска объектов класса Строка  while (current\_position != - 1)  {  // Текущая позиция первого объекта Cтрока  current\_position = code.indexOf(str\_string,  current\_position);  if (current\_position != - 1 )  {  current\_position = current\_position +  str\_string.length() + 1;  // Проверка на пробел после объявления строки  if (code.charAt(current\_position - 1) == ' ')  {  // Запись имени объекта Строки  while (code.charAt(current\_position) != ' ')  {  //Запоминаем текущее имя объекта  if (code.charAt(current\_position) != ';' &&  code.charAt(current\_position) != ')')  {  current\_object = current\_object +  code.charAt(current\_position);  }  current\_position++;  }  // Добавление в строку объектов с проверкой на  уникальность и символы  if (current\_object.indexOf('(') == - 1 &&  objects\_string.indexOf(current\_object) == - 1 &&  current\_object.indexOf('\n') == - 1)  {  objects\_string = objects\_string + " " +  str\_string + " " + current\_object + ",";  }  }  }  current\_object = "";  current\_object\_type = "";  }  // Возвращение текущей позиции в начало строки  current\_position = 0;  // Текущая позиция первого вхождения new  current\_position = code.indexOf(new\_string, current\_position);  // Цикл поиска объектов общего вида  while (current\_position != - 1)  {  buffer\_position = current\_position - 1;  // Запись имени объекта  while (code.charAt(buffer\_position) != ' ')  {  current\_object = current\_object +  code.charAt(buffer\_position);  buffer\_position--;  }  current\_object = reverse(current\_object);  // Переход на начало типа объекта  buffer\_position = current\_position + new\_string.length() +  1;  // Запись типа объекта с проверкой на лишние символы  while (code.charAt(buffer\_position) != ' ' &&  code.charAt(buffer\_position) != ';' &&  code.charAt(buffer\_position) != '"')  {  current\_object\_type = current\_object\_type +  code.charAt(buffer\_position);  buffer\_position++;  }  // Редактирование типа объекта  // Случай массива  if (current\_object\_type.indexOf('[') != - 1)  {  int border = current\_object\_type.indexOf('[');  current\_object\_type = current\_object\_type.substring(0,  border);  current\_object\_type = current\_object\_type + "[]";  }  // Случай STL контейнера  if (current\_object\_type.indexOf('<') != - 1)  {  int border = current\_object\_type.indexOf('<');  current\_object\_type = current\_object\_type.substring(0,  border);  }  // Общий случай  if (current\_object\_type.indexOf('(') != - 1)  {  int border = current\_object\_type.indexOf('(');  current\_object\_type = current\_object\_type.substring(0,  border);  }  // Добавление в строку объектов с проверкой на уникальность  и пустоту  if ( objects\_string.indexOf(current\_object) == - 1 &&  current\_object\_type != "")  {  objects\_string = objects\_string + " " +  current\_object\_type + " " + current\_object + ",";  }  // Переход к новому вхождению new  current\_position = code.indexOf(new\_string,  current\_position + new\_string.length());  current\_object = "";  current\_object\_type = "";  }  // Редактирование конца строки  if (objects\_string != "" && (objects\_string.length() >  start\_length))  {  objects\_string = objects\_string.substring(0,  objects\_string.length() - 1);  System.out.println(objects\_string);  System.out.println();  }  else  {  System.out.println("В исходном коде объекты не были  обнаружены");  System.out.println();  }  }  }  // Метод инвертирования строки  public static String reverse(String string)  {  String reversed\_string = ""; // Инвертированная строка  char[] array = string.toCharArray(); // Массив символов полученной  строки  int last\_index = array.length - 1; // Индекс последнего элемента  массива  for (int index = last\_index; index >= 0; index--)  {  reversed\_string += array[index];  }  return reversed\_string;  }  // Метод составления списка вводимых с клавиатуры переменных  public static void input\_values()  {  // Проверка введенности кода  if (code == "")  {  System.out.println("Ошибка: не был введен код");  System.out.println();  }  else  {  int current\_position = 0; // Текущая позиция искомой переменной  int buffer\_position = 0; // Буферная позиция  String current\_value = ""; // Текущее имя переменной  String values\_string = "Список вводимых переменных:"; // Строка  с списком найденных переменных  int start\_length = values\_string.length(); // Изначальная длина  строки объектов  // Поиск первого вхождения введенной переменной  current\_position = code.indexOf(sc\_next\_string);  // Цикл поиска вводиых переменных  while (current\_position != -1)  {  buffer\_position = current\_position - 1;  // Перевод текущей позиции на конец функции ввода  переменной  while (code.charAt(buffer\_position) != ' ' &&  code.charAt(buffer\_position) != '=')  {  buffer\_position--;  }  // Перевод текушей позиции на первый символ имени вводимой  переменной  while (code.charAt(buffer\_position) == '=' ||  code.charAt(buffer\_position) == ' ')  {  buffer\_position--;  }  // Цикл записи имени переменной  while (code.charAt(buffer\_position) != ' ')  {  current\_value = current\_value +  code.charAt(buffer\_position);  buffer\_position--;  }  current\_value = reverse(current\_value);  // Добавление в строку переменных с проверкой на  уникальность и пустоту  if ( values\_string.indexOf(current\_value) == -1 &&  current\_value.indexOf("") != -1 &&  current\_value.indexOf('"') == -1)  {  values\_string = values\_string + " " + current\_value +  ",";  }  // Переход к новому вхождению подстроки  current\_position = code.indexOf(sc\_next\_string,  current\_position + sc\_next\_string.length());  current\_value = "";  }  // Редактирование конца строки  if (values\_string != "" && (values\_string.length() >  start\_length))  {  values\_string = values\_string.substring(0,  values\_string.length() - 1);  System.out.println(values\_string);  }  else  {  System.out.println("В исходном коде вводимые переменные не  были обнаружены");  }  }  }  } |

Листинг 1. Исходный код класса Lab2\_1.

**Тестирование**

Меню программы:

1. Ввести код

2. Определить, сколько раз в программе создавались объекты

3. Сформировать список объектов (строку), указав название класса этого объекта

4. В какие переменные кода осуществлялся ввод с клавиатуры

5. Сколько в программе использовалось констант

6. Повторный вывод меню

0. Завершение программы

Введите команду: 666

Неверно введена команда

Введите команду: мям

Неверно введена команда

Введите команду: 3

Ошибка: не был введен код

Введите команду: 1

Введите код:

public class Exmp { int a = 55; int b; final int const = 42; String str1; Scanner sc = new Scanner(System.in); String str2; public static void main () { b = sc.nextInt(); str2 = sc.nextLine(); while (a > 50) { a--;} System.out.print (a + b);}}

Код введен

Введите команду: 2

В исходном коде было создано 3 объектов

Введите команду: 3

Список объектов: String str1, String str2, Scanner sc

Введите команду: 4

Список вводимых переменных: b, str2

Введите команду: 5

В исходном коде было создано 1 констант

Введите команду: 6

Меню программы:

1. Ввести код

2. Определить, сколько раз в программе создавались объекты

3. Сформировать список объектов (строку), указав название класса этого объекта

4. В какие переменные кода осуществлялся ввод с клавиатуры

5. Сколько в программе использовалось констант

6. Повторный вывод меню

0. Завершение программы

Введите команду: 9

Неверно введена команда

Введите команду: 0

Программа завершена

**Выводы**

В ходе выполнения лабораторной работы были получены практические навыки по работе с классами String, StringBuilder и регулярными выражениями. Для удобства работы с программой было разработано консольное меню.