**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

Факультет «Информационных технологий и программной инженерии»

Кафедра «Систем обработки данных»

|  |  |
| --- | --- |
| Направление подготовки: | 09.03.02 - Информационные системы и технологии |
| Направленность (профиль): | Технологии проектирования защищенных систем обработки данных |

**Лабораторная работа № 3**

по дисциплине:

|  |
| --- |
| Кроссплатформенное программирование |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Выполнили студенты группы ИСТ-321 | |  |
|  |  | Чаушьянц Р.В., Нариц А.А. | | |
|  |  | *Фамилия И. О.* | | |
|  |  | Руководитель | Заведующий кафедры СОД | |
| *оценка* |  |  | *уч. степень, уч. звание* | |
|  |  | Бородянский Юрий Михайлович | | |
| *дата, подпись* |  | *Фамилия И. О.* | | |

**Цель работы.**

Изучение общих принципов программирования в среде java. Написание приложения, осуществляющего поиск в случайно созданном тексте "слов", являющихся палиндромами, заполнение найденными "словами" массива и распечатка полученного массива на экране.

**Задание.**

Необходимо создать консольное приложение, осуществляющее генерацию случайного текста в размере не менее чем 100 строк. Все "слова" текста составляются программой автоматически из случайно выбранных букв. После создания текста программа осуществляет поиск в тексте слов – палиндромов (то есть таких слов, которые одинаково читаются слева – направо и справа – налево), заполняет этими словами массив и распечатывает на экране исходный текст и найденные палиндромы. Для простоты размер всех "слов" взять одинаковым. Размер "слова" задается пользователем путем ввода с клавиатуры. Так же желательно задавать с клавиатуры число строк и число слов в строке.

**Теоретическая часть.**

Java – это объектно-ориентированный язык программирования, широко используемый для разработки приложений различного уровня сложности. Для успешного освоения Java необходимо понимать синтаксис языка, принципы объявления переменных, а также разницу между простыми (примитивными) и ссылочными типами данных. Кроме того, важно уметь работать с циклами и массивами.

**Общая структура программы**

Программа на Java состоит из классов и методов. Точка входа в программу – это метод main, который имеет следующий вид:

public class Main {

public static void main(String[] args) {

System.out.println("Hello, world!");

}

}

Здесь:

* public class Main – объявление класса Main.
* public static void main(String[] args) – главный метод, с которого начинается выполнение программы.
* System.out.println("Hello, world!"); – вывод текста в консоль.

**Объявление переменных**

В Java переменные объявляются с указанием типа данных. Примитивные типы данных включают:

* int (целые числа)
* double (числа с плавающей запятой)
* boolean (логический тип)
* char (символы)

Пример объявления переменных:

int age = 25;

double price = 99.99;

boolean isActive = true;

char grade = 'A';

Ссылочные типы данных, такие как String и массивы, хранят ссылку на объект в памяти:

String name = "Java";

int[] numbers = {1, 2, 3, 4, 5};

**Разница между примитивными и ссылочными типами**

Примитивные типы хранят конкретное значение в памяти, тогда как ссылочные типы содержат ссылку на объект в куче (heap). Например:

int a = 10;

int b = a;

b = 20; // a остается 10

А при использовании ссылочных типов:

int[] arr1 = {1, 2, 3};

int[] arr2 = arr1;

arr2[0] = 10; // arr1 также изменится

**Циклы и массивы**

Циклы позволяют повторять выполнение блока кода. В Java есть три основных типа циклов:

* for – используется, когда известное количество итераций.
* while – выполняется, пока истинно заданное условие.
* do-while – выполняется хотя бы один раз.

Пример использования for для работы с массивом:

int[] numbers = {1, 2, 3, 4, 5};

for (int i = 0; i < numbers.length; i++) {

System.out.println(numbers[i]);

}

Альтернативный способ – цикл for-each:

for (int num : numbers) {

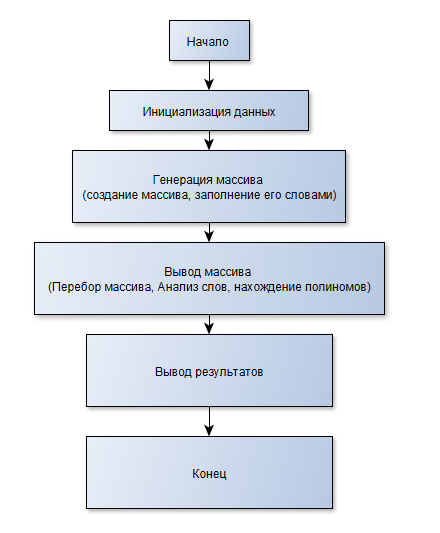
System.out.println(num);

}

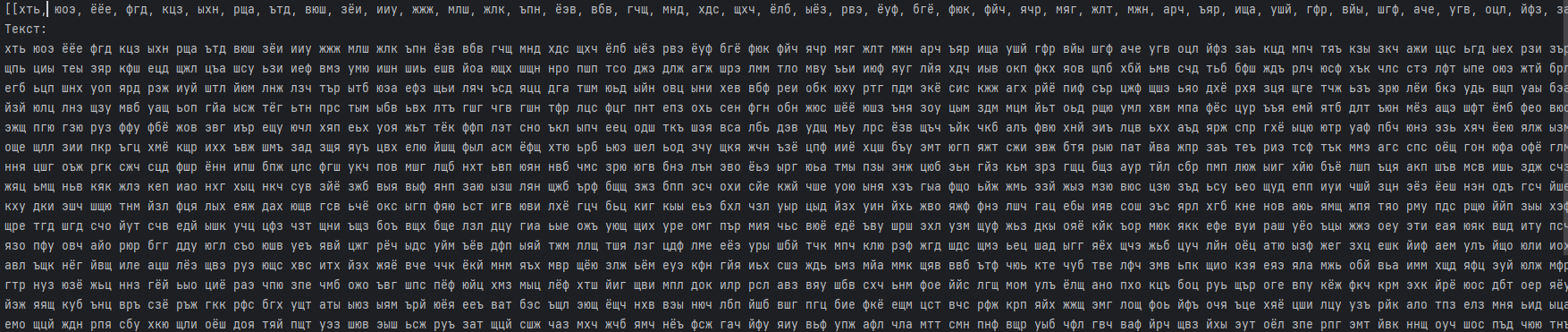
Использование циклов и массивов в Java позволяет эффективно обрабатывать данные и автоматизировать повторяющиеся операции.

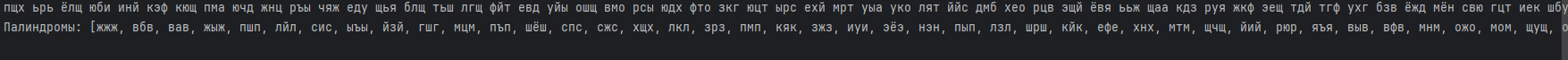
**Ход работы.**

В ходе работы была написана программа, осуществляющее генерацию случайного текста в размере не менее чем 100 строк. Все "слова" текста составляются программой автоматически из случайно выбранных букв. После создания текста программа осуществляет поиск в тексте слов – палиндромов (то есть таких слов, которые одинаково читаются слева – направо и справа – налево), заполняет этими словами массив и распечатывает на экране исходный текст и найденные палиндромы. Для простоты размер всех "слов" взят одинаковым. Размер "слова" задается пользователем путем ввода с клавиатуры. Так же желательно задавать с клавиатуры число строк и число слов в строке.



import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Random;  
  
public class Main {  
 static final String *ALPHABET* = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";  
  
 public static ArrayList<StringBuilder> get\_palindromes(StringBuilder[][] text) {  
 ArrayList<StringBuilder> palindromes = new ArrayList<>();  
 for (StringBuilder[] strings : text) {  
 for (StringBuilder string : strings) {  
// System.out.println(string + " " + string.reverse());  
  
 if (string.toString().contentEquals(string.reverse())){  
 palindromes.add(string);  
 }  
 }  
 }  
 return palindromes;  
 }  
  
 public static void print\_text(StringBuilder[][] text) {  
 System.*out*.print("Текст: \n");  
 for (StringBuilder[] strings : text) {  
 for (StringBuilder string : strings) {  
 System.*out*.print(string + " ");  
 }  
 System.*out*.println();  
 }  
 }  
  
 public static StringBuilder[][] generate\_text(int size\_word, int count\_lines, int size\_line) {  
 Random random = new Random();  
 StringBuilder[][] text = new StringBuilder[count\_lines][size\_line];  
  
  
 for (int i= 0; i < count\_lines; i++) {  
 for (int j = 0; j < size\_line; j++) {  
 StringBuilder word = new StringBuilder();  
 for (int h = 0; h < size\_word; h++) {  
 int randomNumber = random.nextInt(*ALPHABET*.length());  
 char randomChar = *ALPHABET*.charAt(randomNumber);  
 word.append(randomChar);  
 }  
 text[i][j] = word;  
 }  
 }  
  
 System.*out*.println(Arrays.*deepToString*(text));  
  
 return text;  
 }  
  
 public static void main(String[] args) {  
 StringBuilder[][] text = *generate\_text*(3, 100, 100);  
 *print\_text*(text);  
 ArrayList<StringBuilder> palindromes = *get\_palindromes*(text);  
 System.*out*.println("Палиндромы: " + palindromes);  
 }  
}

****

****

**Выводы.**

Входе выполнения лабораторной работы, мы научились нескольким важным навыкам, которые пригодятся нам в будущем:

1. **Работа с многомерными массивами** – мы научились создавать, заполнять и обрабатывать массивы строк (StringBuilder[][]), что полезно для организации данных в сложных структурах.
2. **Использование StringBuilder для работы со строками** – в отличие от String, StringBuilder позволяет эффективно изменять строки, что важно для оптимизации работы с текстом.
3. **Генерация случайных строк** – мы использовали Random для создания случайных слов, что применяется в тестировании, моделировании данных и генерации контента.
4. **Поиск палиндромов** – мы реализовали алгоритм проверки строк на палиндромность, что помогает в обработке текста, анализе данных и разработке алгоритмов на естественном языке.
5. **Форматированный вывод данных** – научились красиво представлять текст и результаты работы программы, что делает код более удобным для пользователя.

Эти навыки пригодятся при разработке приложений, анализе данных и создании алгоритмов обработки текста.