Лабораторная работа №1: Основы синтаксиса Java

В данной лабораторной работе вы изучите основы синтаксиса Java с помощью нескольких простых задач программирования. Далее вы узнаете, как использовать компилятор Java и виртуальную машину Java для запуска программы. От вас потребуется решить следующие задачи:

Простые числа

Создайте программу, которая находит и выводит все простые числа меньше 100.

1. Создайте файл с именем Primes.java, в этом файле опишите следующий класс:

```
public class Primes {
    public static void main(String[] args) {
    }
}
```

Воспользовавшись данным классом, соберите и запустите программу. Так как в данной программе нет конкретной реализации, результата выполнения ее вы не увидите.

2. Внутри созданного класса, после метода main(), опишите функцию IsPrime (Int n), которая определяет, является ли аргумент простым числом или нет. Можно предположить, что входное значение n всегда будет больше 2. Полное описание функции будет выглядеть так:

```
public static boolean isPrime(int n)
{
}
```

Данный метод вы можете реализовать по вашему усмотрению, однако простой подход заключается в использовании цикла for. Данный цикл должен перебирать числа, начиная с 2 до (но не включая) n, проверяя существует ли

какое-либо значение, делящееся на n без остатка. Для этого можно использовать оператора остатка “%”. Например, 17%7 равняется 3, и 16%4 равно 0. Если какая-либо переменная полностью делится на аргумент, сработает оператор return false. Если же значение не делится на аргумент без остатка, то это простое число, и оператор покажет return true. (Оператор return в Java используется для возврата данных из функции, таким способом закрывается метод.)

3. После того, как этот участок будет реализован, приступайте к заполнению основного метода main() другим циклом, который перебирает числа в диапазоне от 2 до 100 включительно. Необходимо вывести на печать те значения, которые ваш помощник IsPrime () посчитал простыми.

4. После завершения вашей программы скомпилируйте и протестируйте её. Убедитесь, что результаты правильные. В интернете вы сможете найти списки простых чисел.

Кроме того, как видно из примера, не следует забывать об использовании комментариев: перед классом с его назначением и перед методом с его целью. Когда вы пишете программы, крайне важно писать подобные комментарии.

Палиндромы

Вторая программа, которую вам необходимо будет написать, показывает, является ли строка палиндромом.

1. Для этой программы, создайте класс с именем Palindrome в файле под названием Palindrome.java. На этот раз вы можете воспользоваться следующим кодом:

```
public class Palindrome {
    public static void main(String[] args) {
        for (int i = 0; i < args.length; i++) {
            String s = args[i];
} }
}
```

Скомпилируйте и запустите эту программу в таком виде, результат работы не будет отображен.

2. Ваша первая задача состоит в том, чтобы создать метод, позволяющий полностью изменить символы в строке. Сигнатура (последовательность) метода должна быть следующей:

public static String reverseString(Strings)

Вы можете реализовать этот метод путем создания локальной переменной, которая начинается со строки "", а затем добавлять символы из входной строки в выходные данные, в обратном порядке. Используйте метод length(), который возвращает длину строки, и метод charAt(int index), который возвращает символ по указанному индексу. Индексы начинаются с 0 и увеличиваются на 1. Например:

```
String s = "pizzeria";
System.out.println(s.length());   //Выводим 8
System.out.println(s.charAt(5));  //Выводим r
```

Вы можете использовать оператор конкатенации (соединения) строк + или оператор +=, на ваше усмотрение.

3. После того, как вы применили метод reverseString (), создайте еще один метод public static boolean isPalindrome(String s). Этот метод должен перевернуть слово s, а затем сравнить с первоначальными данными. Используйте метод Equals (Object) для проверки значения равенства. Например:

```
String s1 = "hello";
String s2 = "Hello";
String s3 = "hello";
s1.equals(s2);  // Истина
s1.equals(s3);  // Ложь
```

Не используйте == для проверки равенства строк. Этим занимается другой тест в Java, который будет рассмотрен далее.

4. Скомпилируйте и протестируйте программу! На этот раз входными данными будут аргументы командной строки, например:

java Palindrome madam racecar apple kayak song noon

Ваша программа должна вывести ответ, является ли каждое слово палиндром.

5. Убедитесь в наличии комментариев, где указаны назначения вашей программы и используемых методов.