МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Одеський національний політехнічний університет

Інститут Комп’ютерних Систем

Кафедра Інформаційних Систем та Технологій

Протокол Модульной работы №1

з дисципліни об’єктно-орієнтоване програмування

на тему: « РАЗРАБОТКА КОНСОЛЬНЫХ ПРИЛОЖЕНИЙ »

Виконав студент групи

АД-201

Стицковський Н.Ю.

Прийняв

Рудніченко Н.Д.

Одеса, 2021

**Содержание**

Теоретическая часть ….. 1

Практическая часть ……. 2

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

4. Простые и составные специальные символы в Java :

Пример простых символов :

char ch = 'a';

Юникод греческого символа Омега в верхнем регистре

char uniChar = '\u039A';

Массив символов

char[] charArray ={ 'a', 'b', 'c', 'd', 'e' };

Составные символы:  
\t Символ табуляции.

\b Символ возврата в тексте на один шаг назад или удаление одного символа в строке (backspace).

\n Символ перехода на новую строку.

\r Символ возврата каретки.

\f Прогон страницы.

\' Символ одинарной кавычки.

\" Символ двойной кавычки.

\\ Символ обратной косой черты (\).

Методы работы с символами:

1 isLetter()

Определяет, является ли значение указанного типа char буквой.

2 isDigit()

Определяет, является ли величина указанного типа char цифрой.

3 isWhitespace()

Определяет, является ли значение указанного типа char пустым пространством.

4 isUpperCase()

Определяет, является ли значение указанного типа char в верхнем регистре.

5 isLowerCase()

Определяет, является ли величина указанного типа char в нижнем регистре.

6 toUpperCase()

Возвращает значение в верхнем регистре в виде указанного типа char.

7 toLowerCase()

Возвращает значение в нижнем регистре в виде указанного типа char.

8 toString()

Возвращает строковый объект, представляющий указанное символьное значение, string - один символ.

19. Понятие и виды конструкторов в Java. В чем отличие конструктора от обычного метода класса?

Конструктор - создает своеобразный “каркас” класса, которому каждый новый объект класса должен соответствовать.

Конструкторы бывают дефолтные и параметризированные.

Параметризированный

public Cat(String name, int age)

{

this.name = name;

this.age = age;

}

Дефолтный

public Cat()

{

}

Похож на обычный метод, но у него нет типа возвращаемого значения. При этом в конструкторе указывается название класса, тоже с большой буквы.

34. Дайте определение понятию конкатенация строк. Как перевернуть строку? Как сравнить значение двух строк? Как обрезать пробелы в начале и конце строки?

Конкатенация строк в Java означает, в целом, слияние, "склеивание" двух строк в одну.

Перевернуть строку можно разными способами, самый простой – использовать StringBuilder/StringBuffer :  
public static String reverseString(String str)

{

return new StringBuilder(str).reverse().toString();

}

Сравнить значение двух строк можно просто через оператор ==, тут примеры излишни, я считаю.

Чтобы удалить пробелы в начале и конце строки можно использовать метод trim() :

String Str = new String(" Hello world! ");

System.out.println(Str.trim());

# ПРАКТИЧЕСКАЯ ЧАСТЬ

**ЗАДАНИЕ НА МОДУЛЬНУЮ РАБОТУ:**

С клавиатуры вводится целое число. Вывести на экран все числа, на которые введённое число делится без остатка. Например, для 60 такими числами станут 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60.

Код программы:

package mod.pkg1.v4.stytskovskyi;

import java.util.Scanner;

import java.util.ArrayList;

import java.util.Arrays;

/\*\*

\*

\* @author USER

\*/

public class Mod1V4Stytskovskyi {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args)

{

// TODO code application logic here

Scanner sc = new Scanner(System.in);

int num;

ArrayList<Integer> numbers = new ArrayList<Integer>();

System.out.println("Введите число: ");

num = sc.nextInt();

for (int i = 1; i <= num; i++)

{

if (num%i == 0)

numbers.add(i);

}

System.out.print("Введённое число " + num + " делится без остатка на следующие числа:");

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

System.out.print(" " + numbers.get(i));

System.out.println(". \n");

}

}

Результат работы программы:

