Розробник:

Огонькова Наталія Максимівна

Група КІТ 101.8б

Варіант №11

**Лабораторна робота № 2**

Алгоритмічна декомпозиція. Прості алгоритми обробки даних

**Мета**: Розробка простих консольних програм для платформи Java SE.

**Вимоги:**

1. Розробити та продемонструвати програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 10 зменшеного на одиницю номера студента в журналі групи.
2. Для визначення вхідних даних використовувати генератор псевдовипадкових чисел (java.util.Random) та забезпечити циклічне (принаймні десять ітерацій) знаходження результату рішення прикладної задачі.
3. Забезпечити виведення до консолі відповідних значень вхідних даних та результатів обчислень у вигляді таблиці.
4. Застосувати функціональну (процедурну) декомпозицію і забезпечити рішення прикладної задачі за допомогою відповідних методів.
5. Забороняється використання даних типу String та масивів при знаходженні рішення прикладної задачі.
6. ЗАВДАННЯ ДО РОБОТИ

Знайти найбільший спільний дільник двох цілих позитивних чисел.

1. ОПИС ПРОГРАМИ
   1. Опис змінних

Random rnd; // генератор випадкових чисел

int num1; // перше випадкове число

int num2; // друге випадкове число

2.2 Ієрархія та структура класів

**class** Main – точка входу в програму

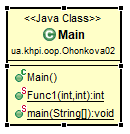


Рисунок 1 – клас Main

* 1. Текст програми

**package** ua.khpi.oop.Ohonkova02;

**import** java.util.Random;

**public** **class** Main {

**public** **static** **int** Func1(**int** a,**int** b) {

**while** (b !=0) {

**int** res = a%b;

a = b;

b = res;

}

**return** a;

}

**public** **static** **void** main(String[]args) {

Random rnd = **new** Random(System.*currentTimeMillis*()); // Инициализируем генератор

**for** (**int** i = 1; i<=10; i++) {

**int** num1 = 1 + rnd.nextInt(20 - 1 + 1); // Получаем случайное число в диапазоне от 1 до 20 (включительно)

**int** num2 = 1 + rnd.nextInt(20 - 1 + 1);

**if**(num1>num2)

{

System.***out***.println(i + ". divider " + num1 + " and " + num2 + " = " + *Func1*(num1, num2));

} **else** **if** (num2>num1)

{

System.***out***.println(i + ". divider " + num1 + " and " + num2 + " = " + *Func1*(num2, num1));

}

}

}

}

1. ВАРІАНТИ ВИКОРИСТАННЯ

Програму можна використовувати для знаходження найбільшого спільного дільника двох цілих позитивних чисел.

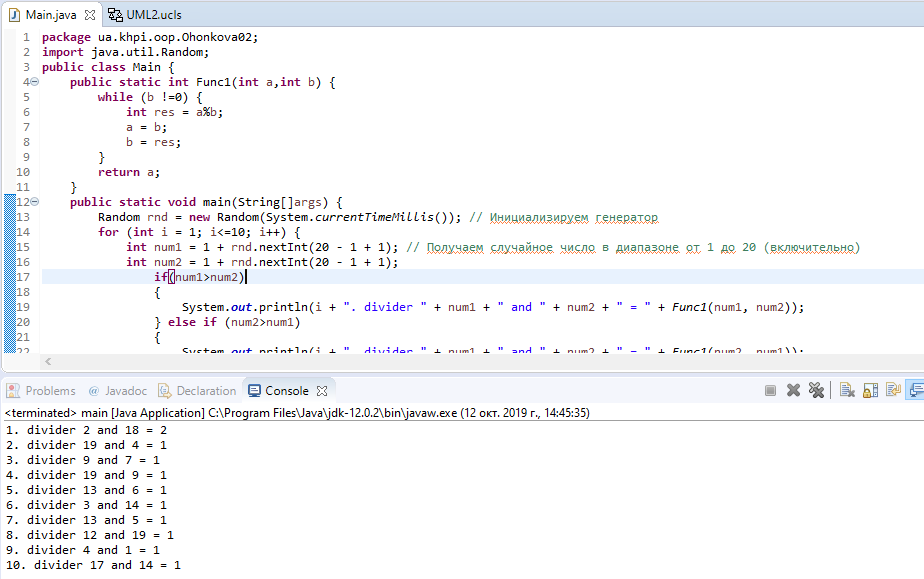


Рисунок 2 – Результат програми

ВИСНОВОК

При виконанні лабораторної роботи набуто практичних навичок, щодо розробки програм за допомогою JDK платформи Java SE. Програма виконується без помилок.