|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ |  |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |  |

**Институт информационных технологий**

КАФЕДРА ИНСТРУМЕТНАЛЬНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИППО)

ПРАКТИЧЕСКАЯ РАБОТА № 5

по дисциплине «Объектно-ориентированное программирование»

«РЕКУРСИЯ»

ВАРИАНТ 2

Выполнил студент группы ИНБО-01-18 *Трофимов С. М.*

Принял Батанов А.О.

Практическая работа выполнена «\_\_\_»\_\_\_\_\_\_\_2019г.

«Зачтено» «\_\_\_»\_\_\_\_\_\_\_2019г.

Москва 2019

**Цель работы**

Изучение работы с рекурсией.

**Теоретическое введение**

В первую очередь надо понимать, что рекурсия — это своего рода перебор. Вообще говоря, всё то, что решается итеративно можно решить рекурсивно, то есть с использованием рекурсивной функции.

Так же как и у перебора (цикла) у рекурсии должно быть условие остановки — Базовый случай (иначе также как и цикл рекурсия будет работать вечно — infinite). Это условие и является тем случаем к которому рекурсия идет (шаг рекурсии). При каждом шаге вызывается рекурсивная функция до тех пор пока при следующем вызове не сработает базовое условие и произойдет остановка рекурсии(а точнее возврат к последнему вызову функции). Всё решение сводится к решению базового случая. В случае, когда рекурсивная функция вызывается для решения сложной задачи (не базового случая) выполняется некоторое количество рекурсивных вызовов или шагов, с целью сведения задачи к более простой. И так до тех пор пока не получим базовое решение.

Итак рекурсивная функция состоит из:

* Условие остановки или же Базовый случай
* Условие продолжения или Шаг рекурсии — способ сведения задачи к более простым.

**Выполнение практической работы**

*Задание:*

Дано натуральное число n. Выведите все числа от 1 до n.

*Решение:*

Файл Main.java

**package** com.gradess.company;  
  
**import** java.util.Scanner;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
 Scanner scanner = **new** Scanner(System.***in***);  
  
 **int** border = scanner.nextInt();  
  
 *printN*(1, border);  
 }  
  
 **private static void** printN(**int** number, **int** border) {  
 **if** (number < border) {  
 System.***out***.print(number + **"\t"**);  
 *printN*(++number, border);  
 }  
 }  
}

**Результат выполнения программы**



**Выводы по работе**

Я изучил на практике работу с рекурсией.

**Используемая литература:**

1. К. Сьерра, Б. Бэйтс. Изучаем Java, 2-е издание.
2. С. Макконнелл. Совершенный код, 3-е издание.