|  |  |
| --- | --- |
| http://www.pl130.ru/doc/index/%D0%9B%D0%BE%D0%B3%D0%BE%D1%82%D0%B8%D0%BF22.jpg | Санкт-Петербургское государственное бюджетное  профессиональное образовательное учреждение  "Колледж электроники и приборостроения" |

отчет

**по практической работе № 14**

**по дисциплине «Основы алгоритмизации и программирования»**

Тема: Применение рекурсивных функций

|  |  |  |
| --- | --- | --- |
| Студент гр. 13 ИТ |  | Липинский К.С. |
| Преподаватель |  | Счастливцев А.К. |

Санкт-Петербург

2024

**Цель работы.**

Научиться организации рекурсивных функций.

**Выполнение работы.**

**Задание 1.**

using static System.Console;

namespace Practice

{

class Program

{

public static int CalculateSequenceTerm(int n)

{

if (n == 1) return -10;

if (n == 2) return 2;

return Math.Abs(CalculateSequenceTerm(n - 2)) - 6 \* CalculateSequenceTerm(n - 1);

}

static void Main()

{

try

{

Write("Enter the term: ");

int n = Convert.ToInt32(ReadLine());

ArgumentOutOfRangeException.ThrowIfNegativeOrZero(n);

Write(CalculateSequenceTerm(n));

}

catch (ArgumentOutOfRangeException)

{

WriteLine("The number is zero or less");

}

}

}

}

****

**Задание 2.**

using static System.Console;

namespace Practice

{

class Program

{

private static string ToBin(int num)

{

int rest = num % 2;

string res = "";

if (num > 0)

{

res += ToBin(num / 2);

res += Convert.ToString(rest);

}

return res;

}

static void Main()

{

Write("Enter any number: ");

int num = Convert.ToInt32(ReadLine());

WriteLine(ToBin(num));

}

}

}

****

**Задание 3.**

using static System.Console;

namespace Practice

{

class Program

{

private static string Poetry(int count)

{

string poem =@$"{count} лунатиков жили на луне

{count} лунатиков ворочались во сне

Один из лунатиков упал с луны во сне

{count - 1} лунатиков осталось на луне";

if (count == 0)

{

return "И больше лунатиков не стало на луне";

}

return $"{poem}\n{Poetry(count - 1)}";

}

static void Main()

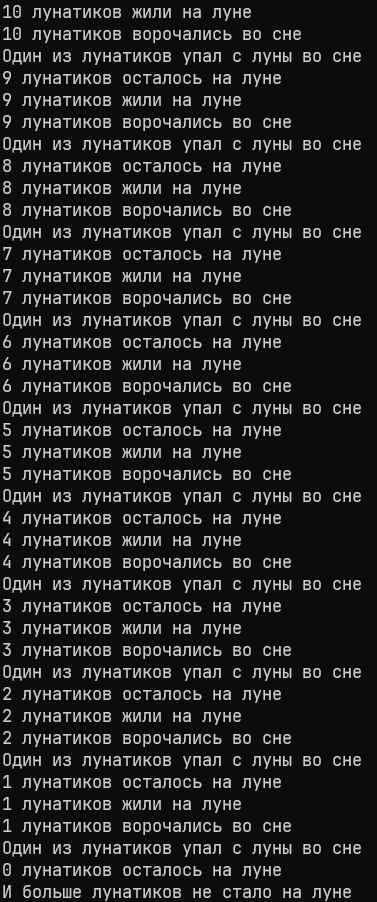
{

Write(Poetry(10));

}

}

}

****

**Вопросы.**

1. Рекурсия – это концепция в программировании, когда функция вызывает саму себя прямо или косвенно
2. Виды рекурсии:
3. Прямая
4. Косвенная
5. Хвостовая
6. Нехвостовая
7. Бесконечная
8. При планировании рекурсии следует учитывать затраты производительности на её выполнение относительно других возможных методов, а также вероятный сценарий бесконечного цикла.
9. Механизмы рекурсивных вызовов и возвратов:
10. Каждый раз при рекурсивном вызове состояние функции сохраняется в стеке функции
11. При возврате из рекурсивной функции состояние функции восстанавливается и продолжается выполнение оставшихся инструкций в функции