Основы алгоритмизации и программирования. Рекурсия

Лекция 8

Привезенцев Д.Г.

Муромский институт Владимирского государственного университета

Очная форма обучения

28 октября 2021 г.

Рекурсия

Оределение

Рекурсией называется ситуация, когда в описании функции вызывается та же самая функция, но с другими значениями параметров.

Замечание! Рекурсивная функция должна включать в себя условие окончания рекурсии. Иначе рекурсия будет происходить бесконечно.

Рекурсия

Механизм реализации рекурсивного решения состоит из:

1. подпрограмма явно или косвенно вызывает саму себя (и называется рекурсивная подпрограмма)
2. при каждом вызове подпрограммы создаются экземпляры локальных переменных подпрограммы, которые доступны только на том уровне рекурсии (вызове подпрограммы), на котором были созданы.

Этим обеспечивается передача данных с одного уровня рекурсии на другой.

Пример

Пример рекурсивного решения задачи вычисления $n!$ для натурального n:

$$n! = \begin{cases} 1 & \text{для } n = 1, \\ n(n-1)! & \text{иначе.} \end{cases}$$

Определение рекурсивной функции для вычисления n!

```
int fact(int n)
{
    if(n == 1)
        return 1;
    else
        return fact(n - 1) * n;
}
int main()
{
    int a, r;
    printf("a = ");
    scanf("%d", &a);
    r = fact(a);
    printf("%d! = %d", a, r);
    getchar(); getchar();
    return 0;
}
```

Пример

Пример рекурсивного решения задачи вычисления x^n для вещественного x и натурального n:

$$x^n = \begin{cases} x & \text{для } n = 1 \\ x^{n-1} \cdot x & \text{для } n > 1 \text{ и } n \text{ - нечетное,} \\ (x^{n/2})^2 & \text{для } n > 1 \text{ и } n \text{ - четное.} \end{cases}$$

Определение рекурсивной функции для вычисления x^n:

```c
float step (float x, int n) {
    float r;
    if (n == 1) return x;
    if (n % 2)  return step(x, n - 1) * x;
    r = step (x, n / 2);
    return r * r;
}
int main() {
    int a, b, r;
    printf("Input a and b: ");
    scanf("%d %d", &a, &b);
    r = step(a, b);
    printf("%d^%d = %d", a, b, r);
    getchar(); getchar();
    return 0;
}
```

Пример

Задача о вычислении суммы нечетных натуральных чисел
$n^2 = 1 + 3 + 5 + \cdots + 2n + 1$

$$\text{f(n)=} \begin{cases} \text{x} & \text{для } n = 1 \\ 2 \cdot n - 1 + f(n-1) & \text{иначе.} \end{cases}$$

Определение рекурсивной функции для вычисления n^2:

```c
int quad(int n)
{
  if(n == 1)
    return 1;
  else
    return 2 * n - 1 + quad(n - 1);
}
int main()
{
  int a, r;
    printf("a = ");
    scanf("%d", &a);
    r = quad(a);
    printf("%d^2 = %d", a, r);
    getchar(); getchar();
  return 0;
}
```

Пример

Задача о вычислении чисел Фибоначи $f(n) = f(n-1) + f(n-2)$

$$\text{f(n)=}\begin{cases} 1 & \text{для } n = 1 \text{ или } n = 2 \\ \text{f(n-1)+f(n-2)} & \text{иначе.} \end{cases}$$

Вычисление чисел Фибоначи

```
int fib(int n){
  if(n == 1 || n == 2)
    return 1;
  else
    return fib(n-2) + fib(n-1);
}
int main(){
  int a, r;
    printf("a = ");
    scanf("%d", &a);
    for(int i = 1; i <= a; i++){
      r = fib(i);
      printf("%d\n", r);
    }
    getchar(); getchar();
  return 0;
}
```

Задачи

1. Написать рекурсивную функцию вычисления суммы цифр натурального числа.
2. Написать рекурсивную функцию вычисления количества цифр натурального числа.