



УВОД В ПРОГРАМИРАНЕТО – УПРАЖНЕНИЕ №13

03.01.2023

СОФТУЕРНО ИНЖЕНЕРСТВО, ГРУПА 6

АСИСТЕНТ: ЕЛЕНА ТУПАРОВА



ЗА КАКВО ЩЕ СИ ГОВОРИМ ДНЕС?

- Кратък преглед от миналия път
- Функции от по-висок ред (function pointers)
- Ламбда функции

ЗАДАЧА 2 – ОТ МИНАЛИЯ ПЪТ

- Да се напише функция със сигнатура **void addNumberToArray(int* &array, int& currentNumbersCount, int& arraySize, int number)**, която добавя числото *number* в масива *array*. Ако в масива няма място, да се увеличи размерът му двойно. (*currentNumbersCount* е текущият брой на заетите позиции в масива, а *arraySize* – размерът му.)

FUNCTION POINTERS – УКАЗАТЕЛИ КЪМ ФУНКЦИИ

- Указателят към функция има за стойност адреса на изпълнимия код на функцията
- Указателите към функции могат да се използват, за да се извикват функции и да се подават функции като параметри на други функции (функции от по-висок ред)
- НЕ може да се извършва указателна аритметика върху указатели към функции

ЗАДАЧА 1

- Напишете функция, която пресмята следната сума:

$$\text{sqrt}(1) + \text{sqrt}(2) + \text{sqrt}(3) + \text{sqrt}(4) + \dots + \text{sqrt}(n).$$

- Как ще промените функцията, така че да може да пресмята и сумата:

$$\text{sqrt}(1) + \text{sqrt}(2) + \text{sqrt}(4) + \text{sqrt}(8) + \dots + \text{sqrt}(n) ?$$

- Може ли да използвате същата функция, ако искате да пресметнете сумата:

$$\sin(1) + \sin(2) + \sin(4) + \sin(8) + \dots + \sin(n) ?$$

ОПЕРАТОР TYPEDEF

- Чрез него можем да задаваме имена (синоними) на типове
- Може да е удобен, когато работим с указатели към функции

ЗАДАЧА 2

- Напишете функция, която сортира масив от цели числа в нарастващ ред на последната им цифра. Ако две числа имат една и съща последна цифра, по-малкото по стойност да бъде преди по-голямото.

Пример:

95 8 13 123 4567 -> 13 123 95 4567 8

ЗАДАЧА 3

- Напишете функция, която сортира масив от цели числа в намаляващ ред на абсолютната им стойност.

ЛАМБДА ФУНКЦИИ

- Т.нар. анонимни функции – дефинират се еднократно на мястото, на което се използват (извикват или подават като параметър на функция)

The diagram shows a C++ lambda function definition with numbered annotations:

- 1: Points to the opening square bracket `[=]`.
- 2: Points to the opening parenthesis `()`.
- 3: Points to the `mutable` keyword.
- 4: Points to the `throw()` exception specifier.
- 5: Points to the `-> int` return type.
- 6: Points to the opening curly brace `{` of the function body.

```
[=] () mutable throw() -> int
{
    int n = x + y;

    x = y;
    y = n;

    return n;
}
```