

Алгоритми върху линейни структури от данни. Стек и опашка



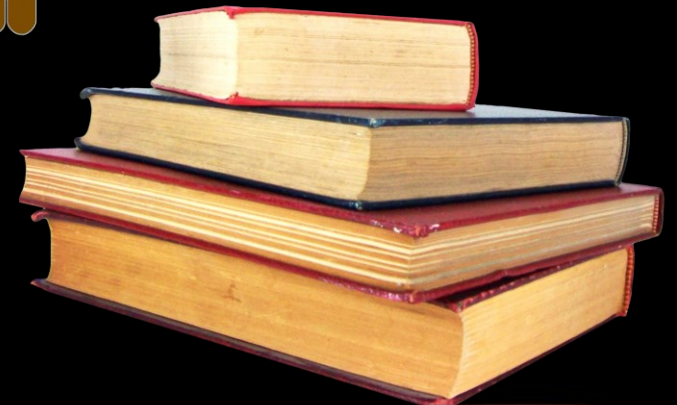
Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Увод в алгоритмите



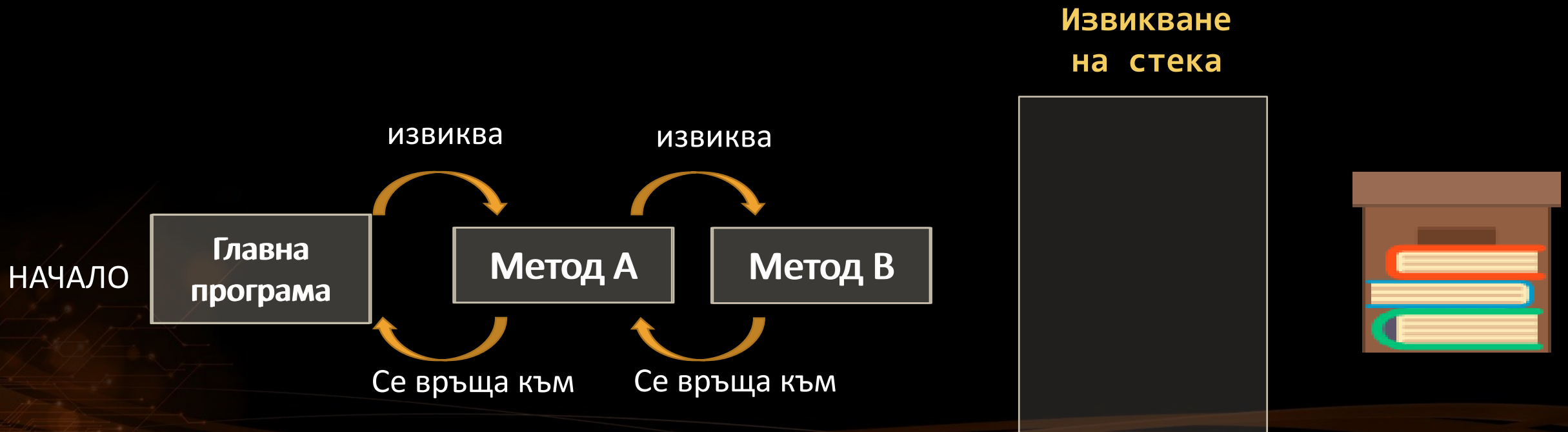
Съдържание

1. Що е стек?
2. Задачи със стек
3. Задачи с опашка



Що е стек?

- Стекът е малко парче памет с фиксиран размер (напр. 1МВ)
- Пази **точката** в която всяка активна подпрограма трябва да **върне контрола**, когато **завърши изпълнението си**



Задача: „Undo“ списък от адреси

- Запазете историята на браузъра. Ще получите възможни команди:
 - URL – отваря дадената страница
 - back – връща към предната страница
 - exit – спира програмата

```
www.softuni.bg  
www.judge.softuni.bg  
www.kids.softuni.bg  
back  
back  
exit
```



```
www.judge.softuni.bg  
www.softuni.bg
```


Решение: „Undo“ списък от адреси

```
if (command == "back")
{
    if (stack.Count != 0) {
        Console.WriteLine(stack.Pop());
    }
    previous = null;
}
else
{
    if (previous != null) {
        stack.Push(previous);
    }
    previous = command;
}
```

Задача: Съответстващи си квадратни скоби

- Даден е аритметичен израз със скоби (може и вложени)
- Цел: извличане на всички **под-изрази** в скоби

$$1 + (2 - (2 + 3) * 4 / (3 + 1)) * 5$$

$$(2 + 3)$$
$$(3 + 1)$$
$$(2 - (2 + 3) * 4 / (3 + 1))$$

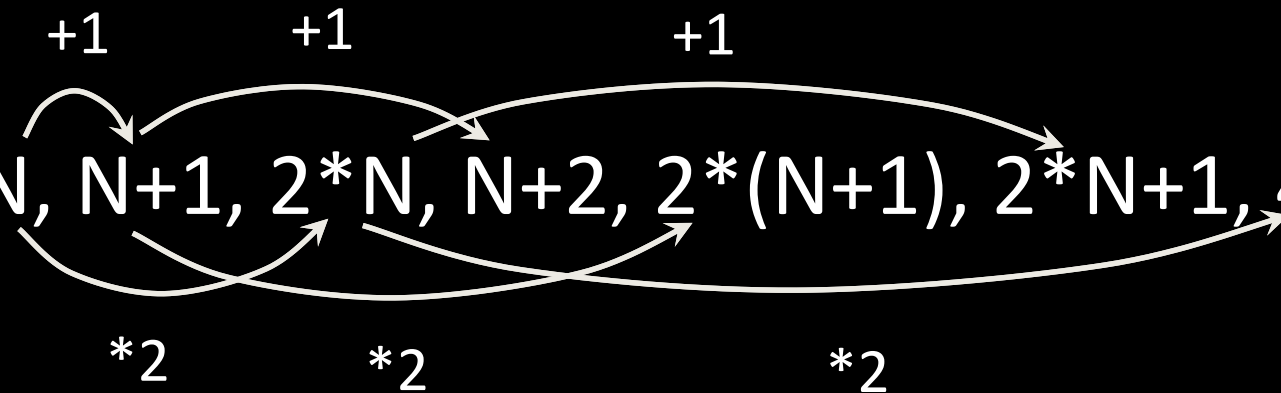
Решение: Съответстващи си квадратни скоби

```
for (int index = 0; index < expression.Length; index++)
{
    char ch = expression[index];
    if (ch == '(')
        stack.Push(index);
    else if (ch == ')')
    {
        int startIndex = stack.Pop();
        int length = index - startIndex + 1;
        string contents = expression.Substring(startIndex, length);
        Console.WriteLine(contents);
    }
}
```

Задача: Редица $N, N+1, 2*N \dots$

- За дадено число N , намерете членът на редицата с индекс P . Номерацията започва от 1. Редицата изглежда така:

- $S = N, N+1, 2*N, N+2, 2*(N+1), 2*N+1, 4*N, \dots$



- $S = 3, 4, 6, 5, 8, 7, 12, 6, 10, 9, 16, 8, 14, \dots$



Решение: Редица N, N+1, 2*N ... (с опашка)

```
int n = 3, p = 16;
Queue<int> queue = new Queue<int>();
queue.Enqueue(n);
int index = 0;
while (queue.Count > 0)
{
    int current = queue.Dequeue();
    index++;
    if (current == p) {
        Console.WriteLine("Index = {0}", index);
        break;
    }
    queue.Enqueue(current + 1);
    queue.Enqueue(2 * current);
}
```

Незавършено: този код се чупи в случай че **p** е недостъпно.

Алгоритми върху линейни структури от данни. Стек и опашка



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

