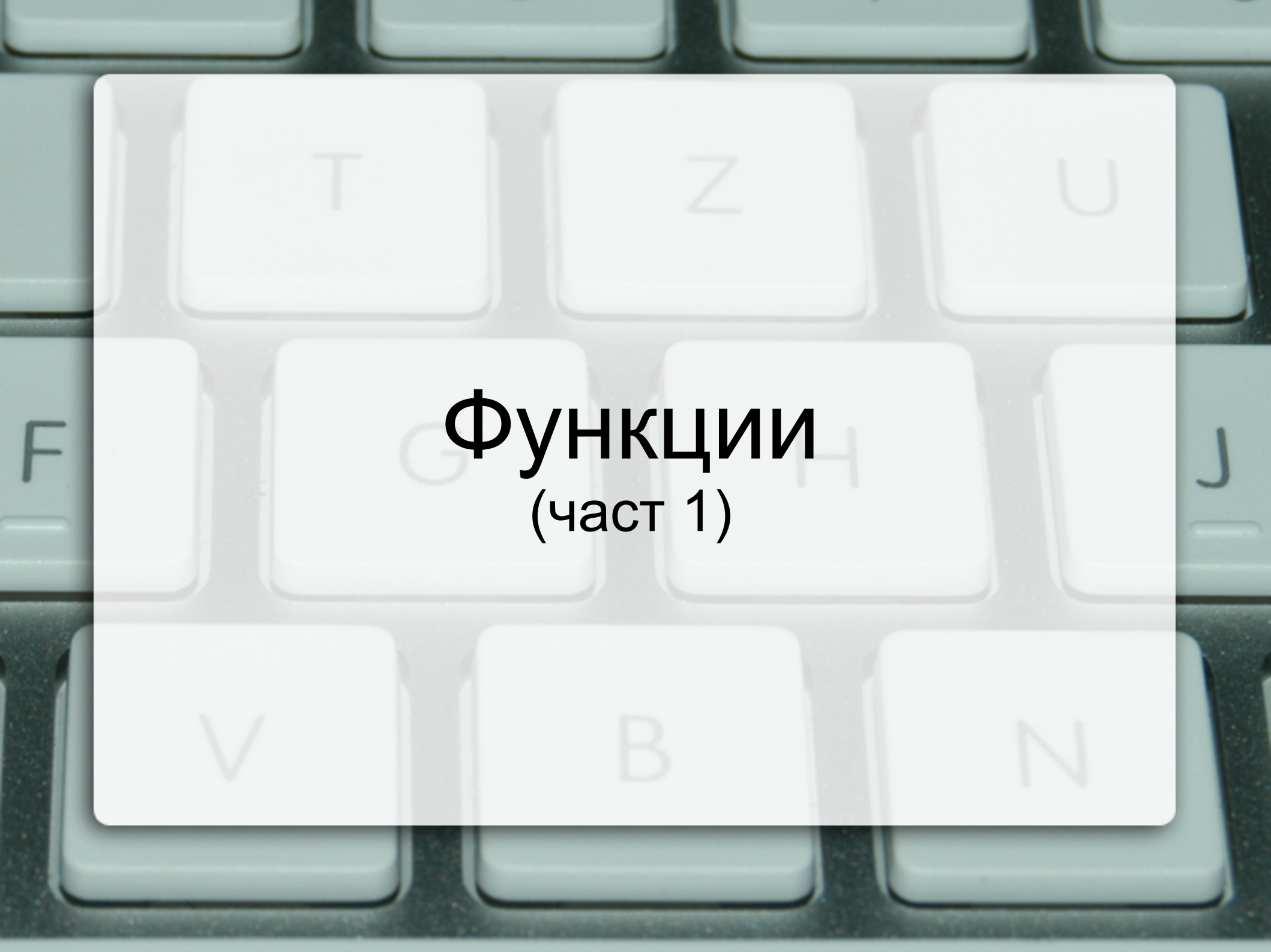


Функции
(част 1)

Функциите в математиката

- Какво е функция в математиката?
- f : Dom → Ran
- Недефинираност на функция
- Dom – дефиниционна област
  - N, { 1, 2, 3, 4 }, [1;100]
- Ran – област от стойности
  - N, {false, true}, { 5, 10, 15, ... }

Какво е програмна функция?

- Относително независима част от програмата, извършваща дадено пресмятане
- Може да бъде използвана многократно
- Пример 1: double square(double);
- Пример 2: double distance(double, double, double, double);
- Не съвсем същото като в математиката!

Какво е подпрограма?

- Относително независима част от програмата, извършваща определена последователност от действия
- Може да бъде изпълнявана многократно
- Още: процедура, метод
- Пример 1: void printHello();
- Пример 2: void printReverseDigits(int n);

Процедури и функции

- Функцията извършва пресмятане и връща резултат
- Процедурата изпълнява поредица от оператори
- Понякога двете понятия се смесват...
- Пример: int readNumber(int from, int to);
- В C++ се наричат просто “функции”

Дефиниране на функция

- <сигнатура> { <тяло> }
- <сигнатура> ::= [<тип> | void]
  <идентификатор> (<**формални**_параметри>)
- <параметри> ::= <празно> | void |
  <параметър> {, <параметър> }
- <параметър> ::= <тип> [<идентификатор>]
- <тяло> ::= { <оператор> }

Извикване на функция

- <име>(<**фактически**_параметри>)
- <**фактически**_параметри> ::= <празно> | void | <израз> {, <израз> }
- извикването на функция е операция с много висок приоритет
- типът на израза съвпада с типа на функцията

Връщане на резултат

- return [<израз>];
- <израз> е от типа на функцията
- работата на функцията се прекратява незабавно (като цикъл при break)
- стойността на <израз> се замества на мястото на извикване на функцията

Деклариране на функция

- <сигнатура>;
- Обещание за дефиниция на функция
- Декларацията не е задължителна
- Може да има много декларации на една и съща функция
- Но може да има само една дефиниция
- Неизпълнените обещания водят до грешка
  - освен когато не се разчита на тях

Пример

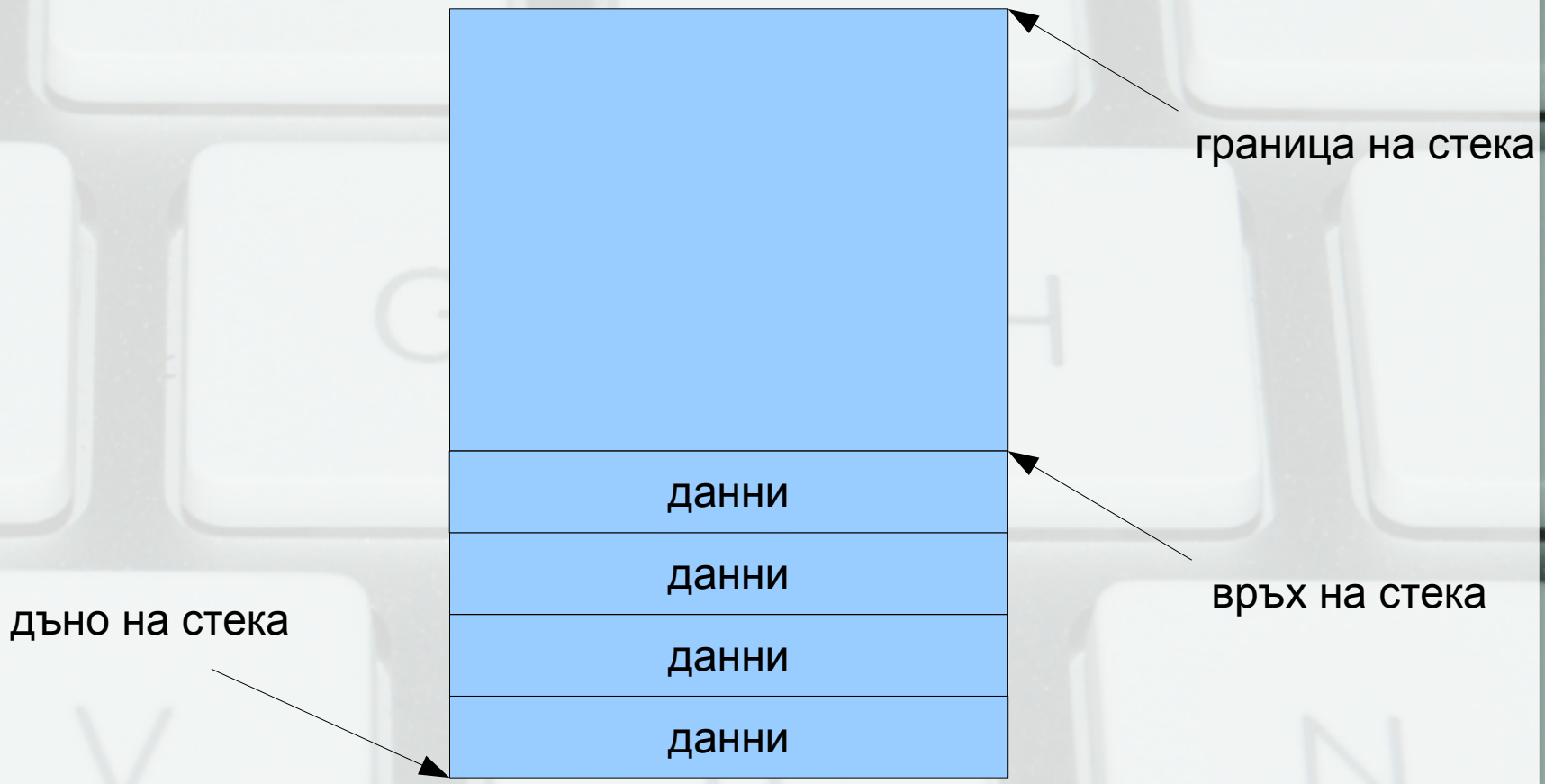
```
double triarea(double x1, double y1, double x2, double y2,
                    double x3, double y3) {
    double a = distance(x1, y1, x2, y2);
    double b = distance(x2, y2, x3, y3);
    double c = distance(x1, y1, x3, y3);
    return heron(a, b, c);
}
double distance(double x1, double y1, double x2, double y2) {
    return sqrt(square(x2-x1)+square(y2-y1));
}
double heron(double a, double b, double c) {
    double p = (a+b+c)/2;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}
double square(double x) { return x*x; }
```

Предимства на функциите

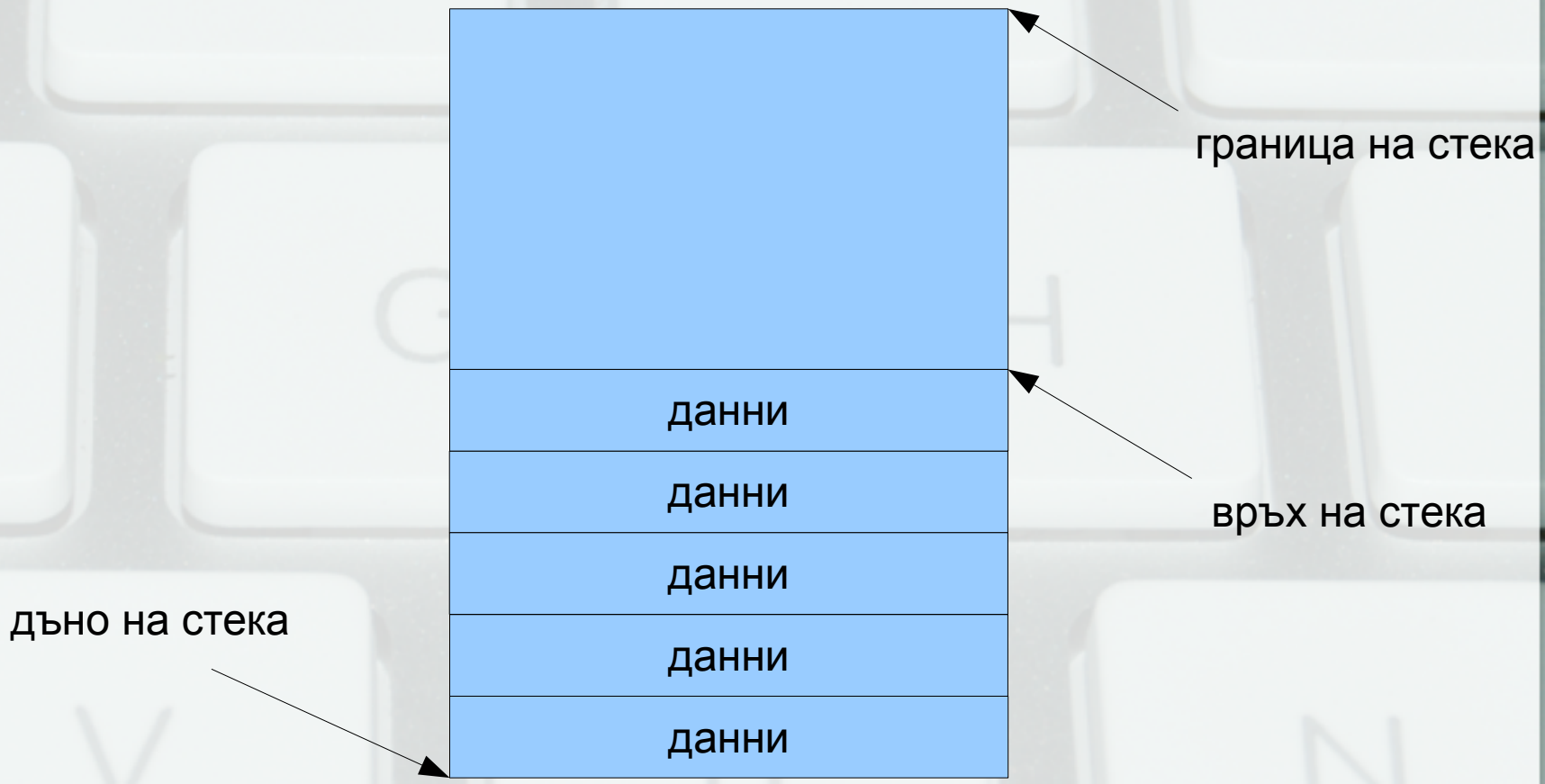
- Разбиване на една сложна задача на няколко по-лесни
- Избягване на повторение на код
- Възможност за преизползване на код
- Разпределение на работата
- Абстрахиране от реализацията
- Улеснява се поддръжката и тестването

Схема на програмната памет

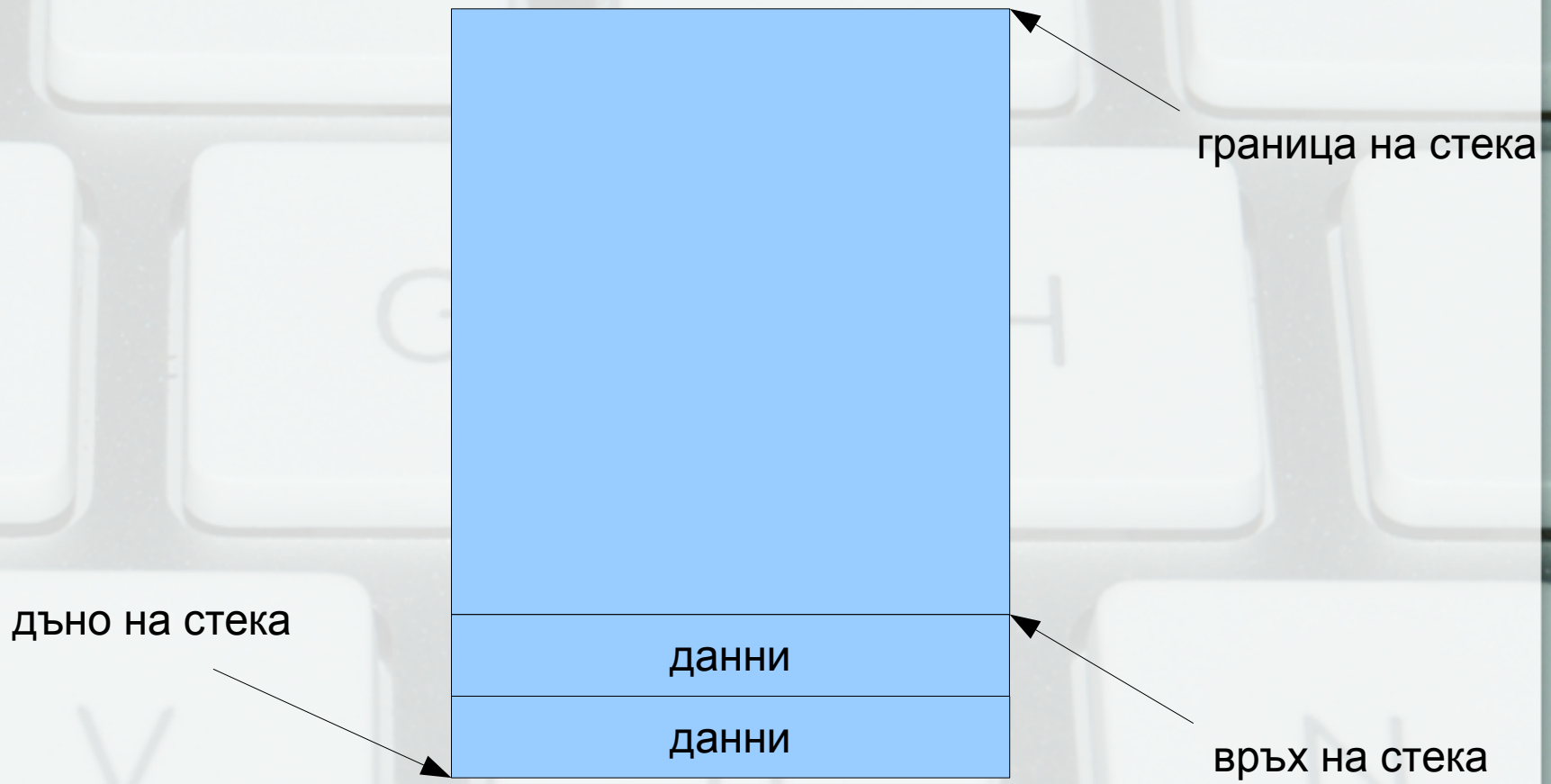
Програмен стек



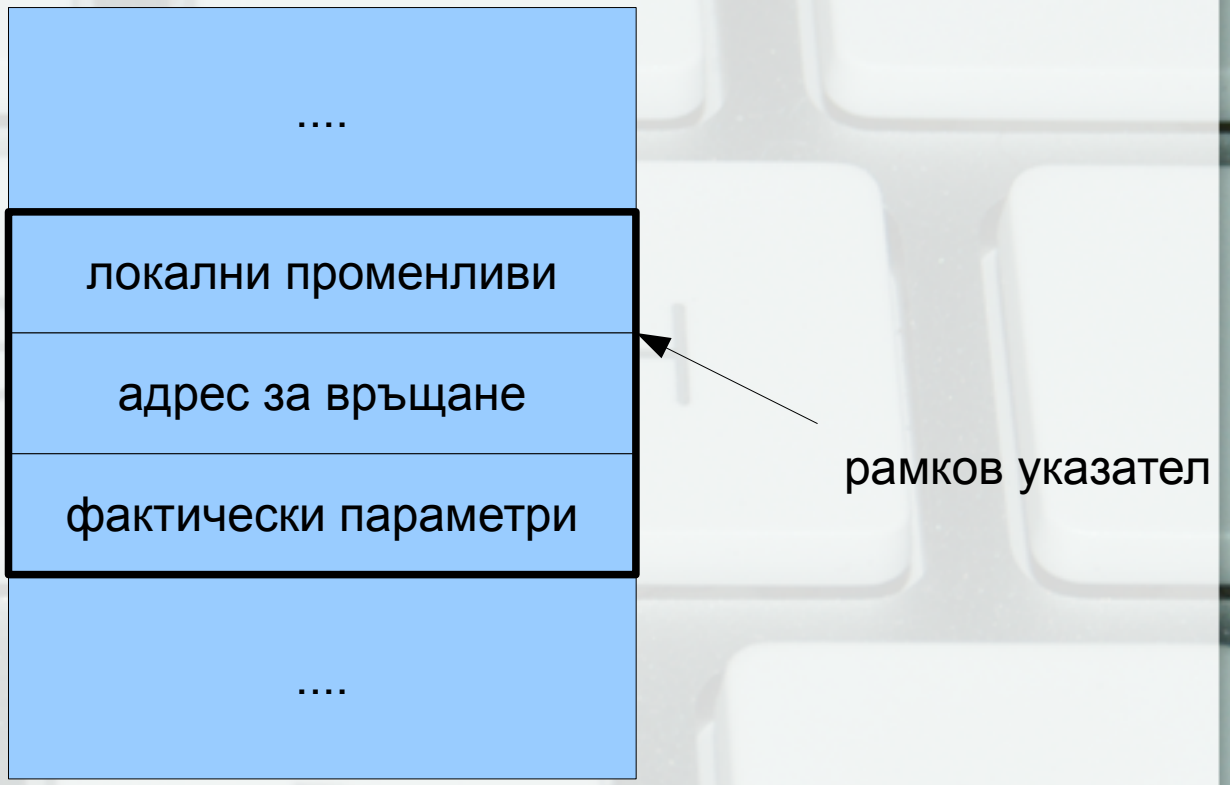
Програмен стек



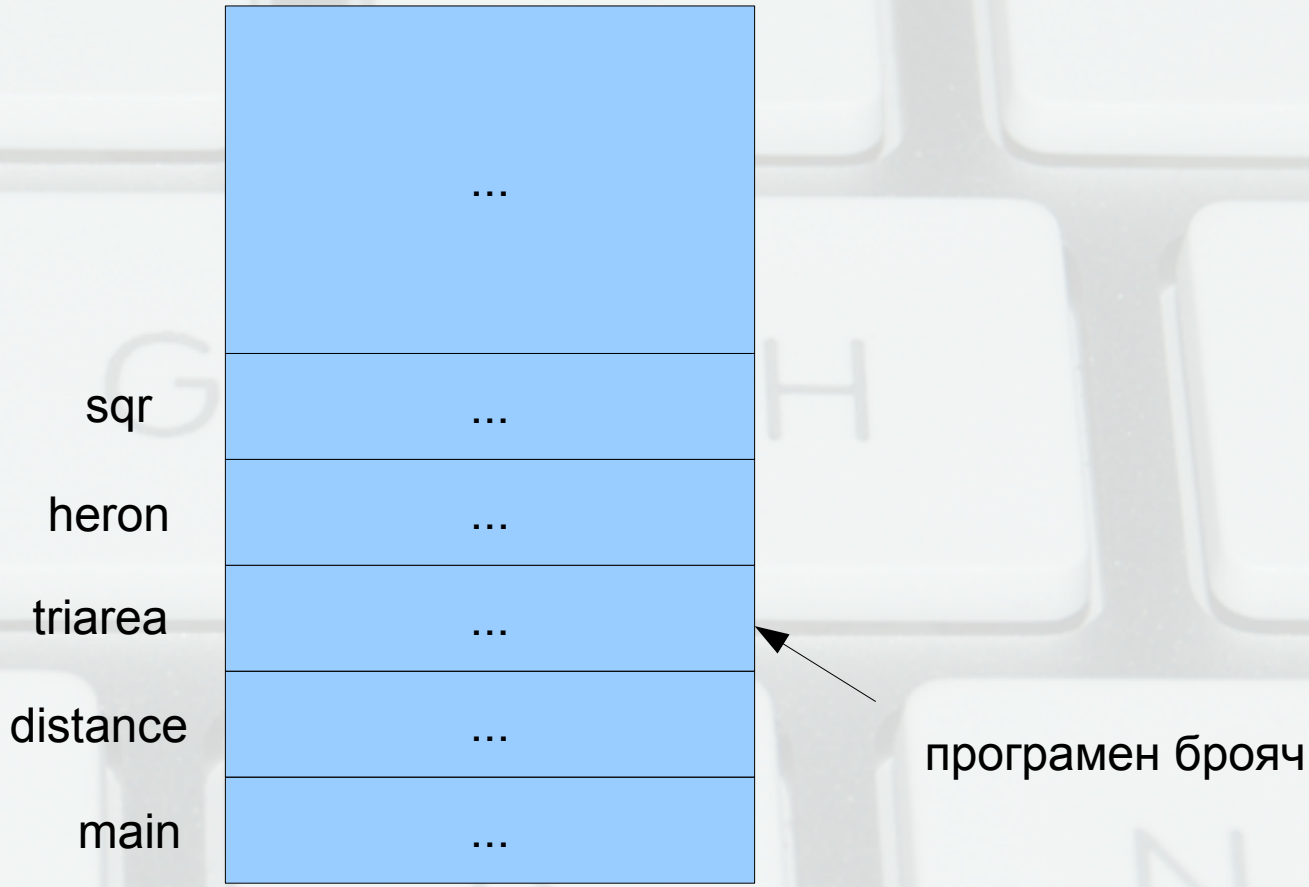
Програмен стек



Стекова рамка на функция



Област за програмен код



Предаване по стойност (call by value)

- пресмята се стойността на фактическия параметър
- в стековата рамка на функцията се създава копие на стойността
- всяка промяна на стойността остава локална за функцията
- при завършване на функцията, предадената стойност и всички нейни промени изчезват