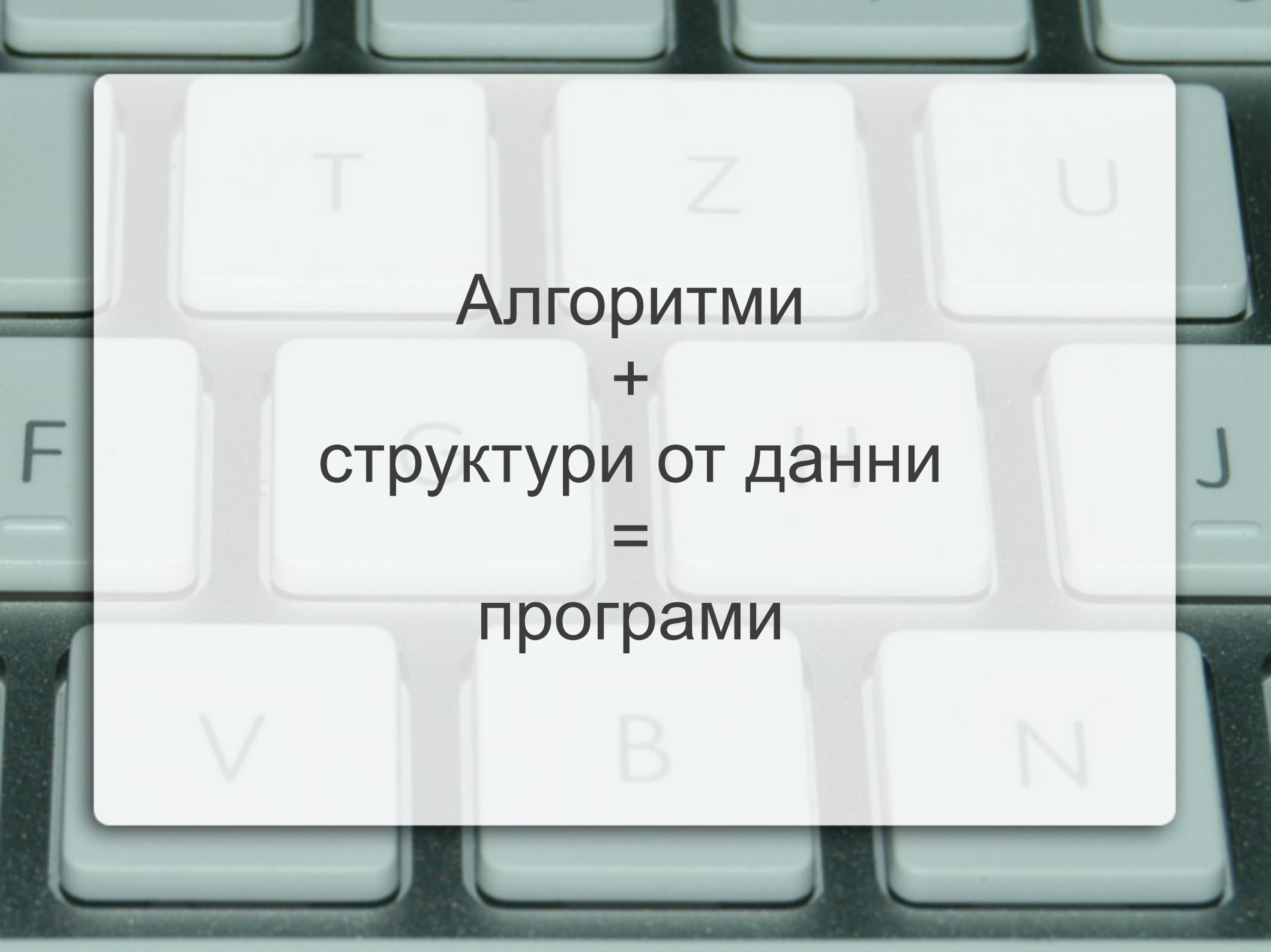
Алгоритми
+
структури от данни
=
програми

Типове данни

- Множество от стойности
- Операции
- За какво служат типовете?

Примитивни типове данни

- булев (bool)
- целочислен (short, int, long, unsigned)
- числа с плаваща запетая (float, double)
- символен (char)
- изброен (enum)
- указател (*)
- псевдоним (&)

Съставни типове данни

- масив ([])
- структура / запис (struct)
- обединение (union)

Структури от данни

- Организация на данни за ефективност
- Всеки тип данни може да се разглежда като структура от данни
- Всяка структура от данни се реализира чрез тип от данни

Абстрактен тип данни

- Математически модел на тип данни или структура от данни
- Не налага конкретна реализация
- Дефинира се чрез операциите си
- Допуска една или повече реализации
- Предимства

Описание на СД

- Логическо описание
  - състав, компоненти, операции, свойства
- Физическо описание
  - представяне в паметта, реализация на операциите

Класификация

- Хомогенни и хетерогенни СД
- Статични и динамични СД
- Линейни и нелинейни СД

Какво е алгоритъм?

Добре дефиниран набор от инструкции за извършване на дадено пресмятане

Сложност на алгоритми

- Алгоритъмът като решение на масова задача
- Оценка за ефективност
- Времева сложност — оценка на време за изпълнение
- Пространствена сложност — оценка на използвана памет

O-нотация

- Нека $f, g : \mathbb{N} \to \mathbb{N}$
- $f \in O(g) \Leftrightarrow$
  $\exists C\ \exists k\ \forall n \geq k\ (\ 0 \leq f(n) \leq C.g(n)\ )$
- $f \in \Omega(g) \Leftrightarrow$
  $\exists C\ \exists k\ \forall n \geq k\ (\ 0 \leq C.g(n) \leq f(n)\ )$
- $f \in O(g) \Leftrightarrow g \in \Omega(f)$
- $\theta(g) = O(g) \cap \Omega(g)$

O-нотация: примери

- $n^2 \in O(n^3)$, $3n^3 + 4n \in \Theta(n^3)$
- $f \in O(f)$, $f \in \Theta(f)$
- $2^n \in O(3^n)$, $\log_2(n) \in \Theta(\log_{10}(n))$
- $\log n \in O(n^{0.001})$, $n^{1000} \in O(1.001^n)$
- $1000000 \in O(1)$, $1 \in O(1000000)$

Видове сложност

- В най-добрия случай (оптимистична)
- В средния случай (средна)
- В най-лошия случай (песимистична)

Пример за времева сложност

```
for(int i = 0; i < n-1 ; i++)
    for(int j = n-2; j >= i; j--)
        if (a[j] > a[j+1]) {
            double x = a[j];
            a[j] = a[j+1];
            a[j+1]= x;
        }
```

- Оценете сложността в най-добрия, средния и най-лошия случай