

# Pseudokoodista toteutukseen

***COMP.CS.300 Tietorakenteet ja algoritmit 1***

***Matti Rintala (matti.rintala@tuni.fi)***

# Pseudokoodin toteutus

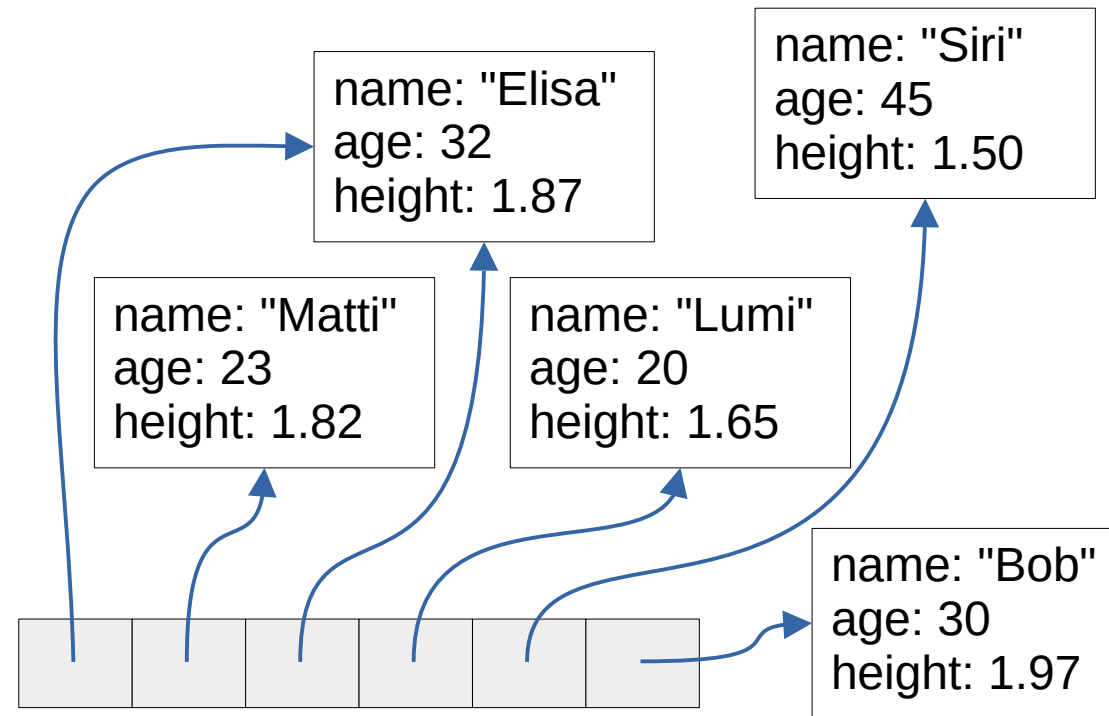
- Sopeuttaminen käyttötilanteeseen
- Syötteiden laillisuuden tarkistukset
- Virhetilanteiden käsittely
- Ohjelmointikielen rajoitukset
- Laitteiston ja kielen aiheuttamat nopeus- ja tarkoituksenmukaisuusnäkökohdat
- Ylläpidettävyys  $\Rightarrow$  modulaarisuus jne.

## Insertion-Sort(*A*)

```
1 for next_elem := 2 to A.length do  
2   key := A[next_elem]  
3   place := next_elem - 1  
4   while place > 0 and A[place] > key do  
5     A[place + 1] := A[place]  
6     place := place - 1  
7   A[place + 1] := key
```

# Ohjelmointikielen vaikutus

- Indeksointi alkaa 0:sta (pseudokoodissa usein 1:stä)
- Käytetäänkö edes indeksointia (tai taulukoita, tai...)
- Kopiodaanko data, vai viitataanko siihen epäsuorasti
- Jos ulkopuoliseen dataan viitataan **epäsuorasti**, tapahtuuko se: *osoittimella*, *älyosoittimella* (esim. `shared_ptr`), *iteraattorilla* (jos data tietorakenteessa), *indeksillä* (jos data vektorissa tms.), *hakuavaimella* (jos data tietorakenteessa, josta haku nopeaa)
- Ovatko algoritmin "parametrit" oikeasti parametreja, vai vain muuttujia tms.



```
1 #include <vector>
2 struct Data { string name; int age; float height; };
3 using Taulukko = std::vector<Data*>;
4 // Täytetään taulukko osoittimilla dataan
5 void insertion_sort(Taulukko& A) {
6     Data* keyp = nullptr; int place = 0;
7     for (Taulukko::size_type next_elem = 1;
8         next_elem < A.size(); ++next_elem) {
9         keyp = Taulukko.at(next_elem);
10        place = next_elem-1;
11        while (place >= 0) {
12            elem = Taulukko.at(place);
13            assert(elem != nullptr);
14            if (keyp->name < elem->name) { break; }
15            Taulukko.at(place+1) = elem; --place;
16        }
17        Taulukko.at(place+1) = keyp;
18    }
```

## Insertion-Sort(*A*)

```
1 for next_elem := 2 to A.length do  
2   key := A[next_elem]  
3   place := next_elem - 1  
4   while place > 0 and A[place] > key do  
5     A[place + 1] := A[place]  
6     place := place - 1  
7   A[place + 1] := key
```