

# Agregatna inicijalizacija

❖ Agregatna inicijalizacija radi sledeće:

- Inicijalizuje kopiranjem svaki element niza, odnosno podobjekat osnovne klase i podobjekat član, po redosledu indeksa, odnosno deklarisanja, odgovarajućim elementom liste
- Ako je element liste ugnežđena lista, odgovarajući element / podobjekat se inicijalizuje tom listom (što opet može biti agregatna inicijalizacija, koja je stoga rekurzivna)
- Ako je tip objekta koji se inicijalizuje niz sa nepoznatim brojem elemenata, taj broj elemenata određen je brojem elemenata u listi:

```
Coord points[] = {{0.,0.}, {3.,0.}, {3.,4.}};
```

Niz *points* ima tri elementa

- Ako elemenata u listi ima više nego elemenata niza / podobjekata koje treba inicijalizovati, prijavljuje se greška
- Ako elemenata u listi ima manje nego elemenata niza / podobjekata koje treba inicijalizovati, preostali se inicijalizuju praznim listama, po pravilima inicijalizacije praznom listom (inicijalizacija vrednošću):

```
Coord points[3] = {{3.,0.}, {3.,4.}};
```

Treći element niza biće inicijalizovan nulom

---

# Inicijalizacija reference

---

- ❖ Reference obe vrste (i na lvrrednosti i na dvrednosti) mogu se inicijalizovati na sve navedene načine (u svim slučajevima umesto jednog znaka & za referencu na lvrrednost, može stajati i dva znaka && za referencu na dvrednost):

- Kada se imenovana referenca deklariše sa inicijalizatorom:

```
T& ref = expression;  
T& ref(expression);  
T& ref = {expression, expression, ...};  
T& ref{expression, expression, ...};
```

- Kada se argument prenosi u pozvanu funkciju, a parametar je referenca:

```
void f (T& t);  
  
f(expression);  
f({expression, expression, ...});
```

- Kada je povratna vrednost funkcije referenca:

```
T& f () {  
    ...  
    return expression;  
}
```

- Kada se nestatički podatak član inicijalizuje u listi inicijalizatora članova u konstruktoru klase:

```
X::X (...) : ref(expression) {...}
```