Inicijalizacija listom

- Tip *std::initializer_list* iz standardne biblioteke predstavlja tip objekata posrednika koji prenose vrednosti iz inicijalizatorske liste u konstruktore klasa i druge funkcije čiji se parametri inicijalizuju listom izraza unutar velikih zagrada. Na primer:

```
class TaskList {
public:
  TaskList (std::initializer_list<Task*> lst) { this->add(lst); }
  TaskList& add (std::initializer_list<Task*>);
  …
};

TaskList& TaskList::add (std::initializer_list<Task*> lst) {
  for (auto t : lst) {
    //… add t to the list
  }
  return *this;
}

int main () {
  Task t1, t2, t3, t4, t5, t6;

  TaskList tlst{&t1, &t2, &t3};

  tlst.add({&t4, &t5}).add({&t6});
}
```

Inicijalizacija listom

- *Inizijalizacija kopiranjem-listom* (*copy-list-initialization*) se obavlja kada se varijabla tipa *T* inicijalizuje listom izraza unutar velikih zagrada u sledećim slučajevima:
  - Kada se imenovana varijabla inicijalizuje listom iza znaka =:

```
T t = {expression, expression, ...};
```

  - Kada se argument prenosi u pozvanu funciju, a parametar inicijalizuje listom:

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

  - Kada se povratna vrednost funkcije inicijalizuje listom:

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

  - U izrazu za indeks kod korisnički definisanog preklopljenog operatora *[]*, kada se listom inicijalizuje parametar tog operatora:

```
object[{expression, expression, ...}]
```

  - U desnom operandu korisnički definisanog preklopljenog operatora dodele, kada se listom inicijalizuje parametar tog operatora:

```
object = {expression, expression, ...}
```

  - U izrazu eksplicitne konverzije u obliku funkcionalnog poziva i u eksplicitnim pozivima konstruktora, kada se parametar odgovarajuće konverzione funkcije ili konstruktora inicijalizuje listom (njome se ne inicijalizuje rezultat, nego parametar):

```
X({expression, expression, ...})
```

  - Kada se nestatički podatak član inicijalizuje listom iza znaka =:

```
class X { T t = {expression, expression, ...}; };
```