

---

# Inicijalizacija listom

---

- ❖ Inicijalizacija listom omogućava da se varijable inicijalizuju na kompaktan način, čitavim listama izraza unutar velikih zagrada (*braced-init-list*) proizvoljne dužine. Na primer:

```
Task t1, t2, ...;
```

```
TaskList myTasks {&t1, &t2, &t3};
```

```
TaskList yourTasks {&t4, &t5};
```

- ❖ Slično važi i za pozive funkcija kojima se mogu prosleđivati liste kao argumenti; tada se njihovi parametri inicijalizuju listom; na primer:

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

```
myTasks.add({&t7});
```

- ❖ Liste izraza unutar velikih zagrada nisu izrazi pa nemaju svoj tip; osim toga, ovakva inicijalizacija ne dozvoljava neke implicitne konverzije (uključujući i neke standardne)

---

# 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});
}
```