

# Tipovi

## ❖ Složeni tipovi (*compound types*):

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• tipovi referenci (<i>reference types</i>):<ul style="list-style-type: none"><li>• tipovi referenci na l-vrednosti (<i>lvalue reference types</i>):<ul style="list-style-type: none"><li>• tipovi <i>lvalue</i> referenci na objekte (<i>lvalue reference to object types</i>)</li><li>• tipovi <i>lvalue</i> referenci na funkcije (<i>lvalue reference to function types</i>)</li></ul></li><li>• tipovi referenci na d-vrednosti (<i>rvalue reference types</i>):<ul style="list-style-type: none"><li>• tipovi <i>rvalue</i> referenci na objekte (<i>rvalue reference to object types</i>)</li><li>• tipovi <i>rvalue</i> referenci na funkcije (<i>rvalue reference to function types</i>)</li></ul></li></ul></li><li>• tipovi pokazivača (<i>pointer types</i>):<ul style="list-style-type: none"><li>• tipovi pokazivača na objekte (<i>pointer-to-object types</i>)</li><li>• tipovi pokazivača na funkcije (<i>pointer-to-function types</i>)</li></ul></li></ul> | <ul style="list-style-type: none"><li>• tipovi pokazivača na članove (<i>pointer-to-member types</i>):<ul style="list-style-type: none"><li>• tipovi pokazivača na podatke članove (<i>pointer-to-data-member types</i>)</li><li>• tipovi pokazivača na funkcije članice (<i>pointer-to-member-function types</i>)</li></ul></li><li>• tipovi nizova (<i>array types</i>)</li><li>• tipovi funkcija (<i>function types</i>)</li><li>• tipovi nabiranja (<i>enumeration types</i>)</li><li>• klasni tipovi (<i>class types</i>):<ul style="list-style-type: none"><li>• ne-unije (<i>non-union types</i>): strukture (<i>struct</i>) i klase (<i>class</i>)</li><li>• tipovi unija (<i>union types</i>)</li></ul></li></ul> |
|---|--|

# Tipovi

- ❖ Za svaki od tipova, osim tipova funkcija ili referenci, postoje tzv. *cv-kvalifikovane* (*cv-qualified*) varijacije: *const* (konstantni), *volatile* (nestalni) i *const volatile*
- ❖ Tipovi su grupisani i u različite kategorije po nekim svojim svojstvima:
  - *objektni tipovi* (*object types*) su (potencijano cv-kvalifikovani) tipovi koji nisu tipovi funkcija, referenci i tip *void*; objekti mogu biti nekog od ovih tipova
  - *skalarni tipovi* (*scalar types*) su (potencijano cv-kvalifikovani) tipovi koji nisu tipovi nizova ili klasni tipovi; entiteti ovih tipova nemaju u sebi ugrađene objekte
  - ❖ *trivijalni tipovi* (*trivial types*), tzv. *POD* (*Plain Old Data*) tipovi, tipovi literala (*literal types*) i drugi
- ❖ Za svaku od navedenih kategorija i potkategorija, u standardnoj biblioteci postoji odgovarajuća šablonska struktura *is\_...* sa podatkom članom *value* koji u vreme prevođenja rezultuje vrednostima *true* ili *false*, u zavosnosti od toga da li je tip koji je argument šablona pripadnik date kategorije; na primer:

```
bool b = std::is_integral<float>::value; // b==false
b = std::is_reference<float&>::value; // b==true
```

- ❖ Specifikator *decltype* određuje tip datog entiteta ili tipa izraza datog u zagradama, na osnovu deklaracija, i to u vreme prevođenja (izraz kao operand se ne izračunava za vreme izvršavanja, već se njegov tip određuje za vreme prevođenja); ovo se može koristiti za definisanje tipova koje je teško odrediti, ili koji zavise od tipova drugih entiteta, tipično u šablonima; na primer:

```
b = is_reference<decltype(x+y)>::value;
```