Sveučilište u Zagrebu
PMF – Matematički odjel

Objektno programiranje (C++)

Vježbe 10 – Lambda izrazi

Matej Mihelčić

Lambda izrazi

- **Lambda calculus** je formalni sustav matematičke logike pomoću kojega **izražavamo računanje** koristeći **apstrakciju funkcije** i **vezanje i supstituciju varijabli**.

- (λx.M) označava **anonimnu funkciju** koja kao **argument ima x** i on se **supstituira unutar izraza M**. Povratna vrijednost lambda izraza (ukoliko postoji) se određuje iz **izvrednjavanjem** izraza M uz odgovarajuću **supstituciju** varijable x.

- Lambda izrazi u programskom jeziku C++ (od verzije 11) su **implementacija** ovoga formalizma.

- [=] () mutable throw() -> int

{

    Naredbe tijela izraza;

}

- Lambda izrazi u C++-u se sastoje od: a) **lambda uvodnik**a (element korištenja) [=], b) potencijalnih **parametara** (navode se unutar oblih zagrada), c) potencijalna **specifikacija promijenjivosti** (mutable), d) potencijalna **specifikacija iznimaka** (throw), e) potencijalna **specifikacija povratnog tipa**, f) **tijelo lambda izraza**.

Lambda uvodnik:

- Lambda izraz može **pristupiti** ili **koristiti** varijable iz blokova koji ga okružuju. Od **C++14**, lambda izrazi mogu **definirati i nove varijable unutar tijela**.

- Lambda uvodnik specificira koje varijable se koriste i na koji način (po **vrijednosti** ili po **referenci**).

Lambda izrazi

- Varijable koje imaju (&) se koriste **preko reference** a one koje nemaju **preko vrijednosti**. Ukoliko je lambda uvodnik definiran kao [], tada lambda izraz **ne pristupa** varijablama iz blokova koji ga okružuju. - Može se definirati i **standardni** oblik korištenja varijabli (**vrijedi za sve varijable** iz blokova koji okružuju lambda izraz a koriste se unutar izraza). Ukoliko je lambda uvodnik definiran kao [&] tada se **sve varijable iz okruženja koriste preko reference** dok se uz uvodnik [=] **sve varijable iz okruženja koriste preko vrijednosti**. **Standardni oblik korištenja varijabli se može kombinirati sa oblikom korištenja individualnih varijabli**. Npr. lambda uvodnik [&, prva] definira da se sve varijable iz okruženja koriste preko reference osim varijable *prva* koja se koristi preko vrijednosti.

- **Ne smije se** kombinirati standarni oblik lambda uvodnika s individualnim oblikom istog tipa. Npr. [&, &prvi] će uzrokovati grešku pri prevođenju.

- Kao lambda uvodnik se **može koristiti i pokazivač *this*** koji omogućava pristupanje elementima klase preko reference. Ne smije se koristiti ukoliko je definiran standardni lambda uvodnik [=].

- Korištenje varijabli iz okruženja preko **reference omogućava izmjenu njihovih vrijednosti unutar lambda izraza**, dok **pristup preko vrijednosti zahtjeva da se varijabla tretira kao konstanta**.

Lambda parametri:

- Navode se unutar oblih zagrada **tipom i imenom varijable**. Moguće je definirati lambda izraze kao parametre lambda izraza. Od C++14 je moguće definirati i varijablu tipa *auto* **ukoliko se radi o varijabli generičkog tipa**.

Lambda izrazi

Specifikacija promijenjivosti:

- Dodavanje ključne riječi *mutable* **omogućava mijenjanje vrijednosti kopija onih varijabli kojima se pristupa preko vrijednosti** unutar lambda izraza.

Specifikacija iznimke:

- Moguće koristiti ključnu riječ *noexcept* da se naznači da lambda izraz **ne prijavljuje iznimku**.

Povratni tip:

- Povratnu vrijednost je moguće definirati navođenjem „->” *tip*.

- Povratna vrijednost se može ispustiti ukoliko **lambda izraz sadrži jednu return naredbu** ili **ukoliko ne vraća vrijednost**.

- U tom slučaju, povratna vrijednost se **automatski zaključuje** iz tipa izraza nakon *return* naredbe ukoliko lambda izraz sadrži samo jednu naredbu *return*. U ostalim slučajevima je zaključena vrijednost *void*.

Tijelo lambda izraza:

- Može sadržavati iste komponente kao i tijelo funkcije: a) varijable iz blokova koji okružuju izraz, b) parametre, c) lokalno definirane varijable (C++14), d) elemente članove klasa (izraz treba biti definiran unutar klase i lambda uvodnik treba sadržavati *this)*, e) proizvoljne varijable sa statičkim memorijskim trajanjem (npr. globalne varijable).

Zadatak

Zadatak:

- Neka su u glavnom programu zadane dvije varijable int x = 1, y = 2;
    - Definirajte lambda izraz i pridijelite ga funkciji *f* odgovarajućeg prototipa, tako da pozivom *f(x,y)* kao rezultat dobijemo x+y.
    - Promijenite funkcionalnost funkcije *f* tako da pozivom *f(x,y)* kao rezultat dobijemo x-y.
    - Definirajte lambda izraz i pridijelite ga funkciji *g* odgovarajućeg prototipa tako da pozivom *g(&x)* dobijemo ++x.
    - Definirajte lambda izraz koji koristi varijable iz okruženja preko reference i pridjelite ga funkciji g2 (koristite *auto* za tip), pozivom g2() trebamo inkrementirati i varijablu x i varijablu y.
    - Napravite funkciju g3 i odgovarajući lambda izraz tako da pozivom g3() inkrementiramo kopije varijabli x i y (unutar lambda izraza imamo ++x, ++y) no izvan lambda izraza su vrijednosti od x i y nepromijenjene.
    - Definirajte klasu točka koja sadrži cjelobrojne koordinate x i y te definirajte pripadni konstruktor, predefinirajte operator <<, tako da možemo ispisati proizvoljnu točku u formatu (x , y). Definirajte funkciju *pomakni(int a, int b)* koristeći lambda izraz tako da za *Tocka t(1,2)* pozivom *t.pomakni(-1,2)* točka t ima koordinate (0,4).

Zadatak

Zadatak:

- Definirajte vektor realnih brojeva te ih:
  - Sortirajte prema vrijednosti
  - Sortirajte prema apsolutnoj vrijednosti
  - Sortirajte prema kubu vrijednosti
  - Ispitajte jesu li svi brojevi u vektoru parni
  - Ispitajte jesu li svi brojevi u prvoj/drugoj polovici vektora djeljivi s 5
  - Ispitajte je li bilo koji od prvih *k* brojeva vektora negativan (isprobajte za nekoliko različitih vrijednosti parametra *k*)
  - Ispitajte je li istina da niti jedan broj u vektoru nije veći od 200, a) po vrijednosti, b) po apsolutnoj vrijednosti

Zadatak

Zadatak:

- Definirajte vektor stringova:
  - Promijenite svaki string u vektoru tako da su sva slova mala
  - Izbacite sve samoglasnike iz stringova
  - Invertirajte stringove
  - Pronađite prvi string koji sadrži slovo 's'
  - Pronađite prvi string koji ne sadrže podstring „rp”
  - Definirajte dodatni vektor stringova i konkatenirajte te stringove sa stringovima prvog vektora. Rezultat treba biti spremljen u prvi vektor (koristite *transform*).
  - Zamijenite sva pojavljivanja string koji sadrži više od 10 znakova stringom „-1”.

Zadatak:

- Generirajte vektor od *n* stringova. Svaki string sadrži točno *duljina* slova (*duljina* je slučajno generirani broj <20). Prvo slovo stringa je slučajno generirano veliko slovo, a zatim slijede *duljina – 1* slučajno generiranih malih slova. Svaki string ima jedinstveni id koji se dodaje nakon zadnjeg znaka u stringu (koji odgovara rednom broju stringa u vektoru). Stoga cijeli string može imati i veću duljinu od 20 (iako ima obavezno <20 slova).