




PROGRAMIRANJE
Predavanje 13 – Pozivi funkcija

Ishod učenja 5

1

REFERENCE

Strana • 2

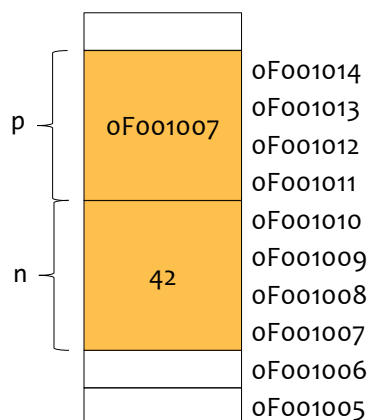


2

Kratki pregled pokazivača

- Pokazivač sadrži memorijsku adresu
 - Ta memorijska adresa sadrži nešto – kažemo da pokazivač pokazuje na nešto korisno
 - Pokazivač koristi 4 ili 8 bajtova

```
int n = 42;
int* p = &n;
```



Strana * 3



3

Reference

- Referenca je **dodatno ime za postojeću varijablu/objekt**
 - Dakle, nije nova varijabla niti novi objekt
- Referenca se deklarira pomoću operatora '&'
- Nemojmo brkati s operatorom uzimanja adrese – potpuno različita priča!

```
int number = 42;
int& answer = number;

Person p;
p.first_name = "Vanessa";
p.last_name = "Brown";
p.age = 31;
Person& sister = p;
```

Strana * 4



4

Vizualizacija

Ova varijabla ima dva imena:
number i answer

Ovaj objekt ima dva imena:
p i sister


42

Vanessa

Brown

31

Strana • 5



5

PARAMETRI FUNKCIJA

Strana • 6



6

Uvod

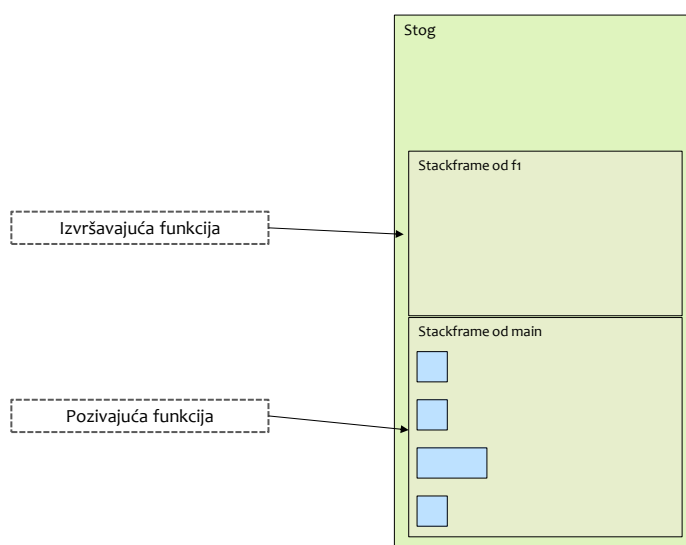
- Znamo da funkcija prima kopiju originalne vrijednosti parametra
- Ponekad bi bilo dobro kad bi funkcija mogla raditi s originalnom vrijednosti:
 - Performanse (zašto kopirati veliki objekt samo radi ispisa?)
 - Funkcionalnosti (vratiti više vrijednosti iz funkcije)
- Želimo da izvršavajuća funkcija može pristupiti varijablama/objektima/poljima pozivajuće funkcije
 - Ali samo ako mi odlučimo tako
 - Želimo moći odlučiti za svaki parametar želimo li njegovu vrijednost kopirati ili poslati original

Strana • 7



7

Vizualizacija



Strana • 8



8

Način 1: poziv funkcije po vrijednosti

- Kod **poziva funkcije po vrijednosti** (engl. *call by value*), od prosljeđenih varijabli se rade **lokalne kopije** za tu funkciju
 - Način na koji smo do sada koristili funkcije
- Promjena kopije ne mijenja vrijednost originalne varijable

```

void multiply(int n) {
    n *= 2;
}

int main() {
    int n = 17;

    multiply(n);

    cout << n << endl;

    return 0;
}

```

Prima lokalnu kopiju

Mijenja kopiju na 34

Prikazuje originalnu vrijednost 17

Strana * 9



9

Način 2: poziv funkcije pokazivačem

- Kod **poziva funkcije pokazivačem** (engl. *call by pointer*), u pozvanoj funkciji dobivamo pokazivač na dio memorije pozivajuće funkcije

```

void multiply(int* n) {
    *n *= 2;
}

int main() {
    int n = 17;

    multiply(&n);

    cout << n << endl;

    return 0;
}

```

Prima adresu originalne vrijednosti

Mijenja **originalnu** vrijednost na 34

Šalje adresu

Prikazuje promijenjenu vrijednost 34

- C-ovski pristup

Strana * 10



10

Način 3: poziv funkcije putem reference

- Kod **poziva funkcije putem reference** (engl. *call by reference*), samo radimo **lokalno ime** za originalnu kućicu

```
void multiply(int& n) {
    n *= 2;
}

int main() {
    int n = 17;

    multiply(n);

    cout << n << endl;

    return 0;
}
```

Lokalno ime za originalnu varijablu

Mijenja **originalnu** vrijednost na 34

Prikazuje promijenjenu vrijednost 34

- C++ pristup

Strana • 11



11

Što koristiti

- Možemo koristiti sljedeće smjernice:
 - Kad šaljemo primitivnu vrijednost, obično šaljemo kopiju
 - Osim u rijetkim situacijama kad želimo da funkcija mijenja original (tad šaljemo kao referencu)
 - Kad šaljemo polje, šaljemo njegovu adresu
 - Jer je naziv polja sam po sebi pokazivač
 - Kad šaljemo objekt, obično šaljemo kao referencu

Strana • 12



12

Slanje polja u funkciju

- Naziv polja je i pokazivač na njegov 0-ti element
 - Lako nam je poslati adresu polja funkciji
 - Ali, kako će funkcija znati veličinu polja?
- Određivanje veličine polja:
 - U funkciji gdje je deklarirano, možemo koristiti `sizeof()`

```
int grades[11] = { 2, 4, 4, 5, 1, 1, 2, 1, 2, 3, 1 };
cout << sizeof(grades); // Displays 44
cout << sizeof(grades) / sizeof(int); // Displays 11
```
 - Ali, kad primimo adresu polja u funkciji, `sizeof()` ne radi kako bismo očekivali
 - `sizeof()` će vratiti 4 ili 8 buduće se radi o pokazivaču

Strana • 13



13

Slanje polja u funkciju

- Dakle, kad šaljemo adresu polja u funkciju, moramo poslati i veličinu polja kao dodatni parametar

```

      Adresa polja      Veličina polja
      |                 |
      v                 v
void display(int* sizes, int n) {
    for (int i = 0; i < n; i++) {
        cout << sizes[i] << endl;
    }
}
int main() {
    int numbers[] = { 2, 4, 6, 8, 10 };
    display(numbers, 5);
    return 0;
}
      Adresa polja      Veličina polja

```

Strana • 14



14

Primjeri

1. Napišite funkciju koja prima pokazivač na cijeli broj i množi taj broj s 5. Pozovite funkciju i ispišite rezultat te nacrtajte kako izgleda stog tijekom izvršavanja programa.
2. Napišite funkciju koja prima referencu na cijeli broj i množi taj broj sa 5. Pozovite funkciju i ispišite rezultat te nacrtajte kako izgleda stog tijekom izvršavanja programa.
3. * Napišite funkciju koja prima referencu na pokazivač na cijeli broj i množi taj broj sa 5. Pozovite funkciju i ispišite rezultat te nacrtajte kako izgleda stog tijekom izvršavanja programa.

Strana * 15



15

Primjeri

4. Napišite funkciju koja prima polje cijelih brojeva i ispisuje ih. Napišite drugu funkciju koja prima polje cijelih brojeva i ispisuje ih, ali obrnutim redoslijedom. Za primanje polja u funkcije koristite pokazivač na nulti element i prvi element iza kraja polja. Napravite polje i prosljedite ga u obje funkcije.
5. Riješite prethodni zadatak tako da prosljedite broj elemenata u polju.
6. Napišite funkciju koja prima pokazivač na objekt tipa Osoba (ime, prezime). Neka funkcija učita ime i prezime osobe. Pozovite funkciju i ispišite podatke u mainu.
7. Riješite prethodni zadatak korištenjem referenci.

Strana * 16



16

Primjeri

8. Definirajte strukturu koja čuva podatke o nazivu knjige i broju primjeraka knjige u knjižnici. Napišite funkciju koja kao parametar uzima polje elemenata tipa Knjiga te vraća ukupan broj primjeraka svih knjiga. U glavnom programu kreirajte polje od 3 objekta tipa Knjiga, te pozovite funkciju.
9. Napišite funkciju zamijeni() koja mijenja vrijednosti prosljeđenim varijablama. U glavnom programu napravite polje od 5 elemenata i učitajte od korisnika vrijednosti u njega. Zatim prođite kroz polje i ako je lijevi element veći od desnog napravite njihovu zamjenu. Na kraju ispišite polje.

Strana • 17



17

Primjeri

10. Napišite funkciju koja prima polje cijelih brojeva i vraća vektor sastavljen samo od parnih brojeva. Pozovite funkciju i ispišite vraćeni vektor.
11. Napišite funkciju join() koja prima dva polja stringova i vraća jedno dinamičko polje sastavljeno od sadržaja oba polja. Pozovite funkciju i ispišite rezultat.
12. Napišite funkciju koja vraća i sumu i umnožak prosljeđena dva broja.

Strana • 18



18

Zadaci za sljedećih 7 dana

▪ Prije sljedećeg predavanja trebate:

1. Pročitati iz *Demistificirani C++*:
 - 6.2 Reference
 - 6.3 Razlika između referenci i pokazivača
 - 6.4 Dinamička alokacija objekata
 - Bez 6.4.1
 - 8.4 Lista argumenata
 - Bez 8.4.5, 8.4.6, 8.4.8, 8.4.9