C++ pokazivači

Deklaracija i inicijalizacija pokazivačkih varijabli, prijenos i vraćanje adresa, rezerviranje memorije...

Pokazivači

- Varijable za pamćenje memorijskih adresa.
- Rad sa adresama
  - omogućuje izvođenje operacija niže razine
  - daje značajnu kontrolu i povećava snagu C++ jezika
- Greške u programiranju -> opasnosti!
- Pokazivači se mogu efikasno koristiti
  - pri radu s poljima
  - za povezivanje klasa i objekata
  - kreiranje podatkovnih struktura (stogova, redova, lista...)
  - rezerviranje i oslobađanje memorije u vrijeme izvođenja programa
- Pokazivači povećavaju vještine programera...

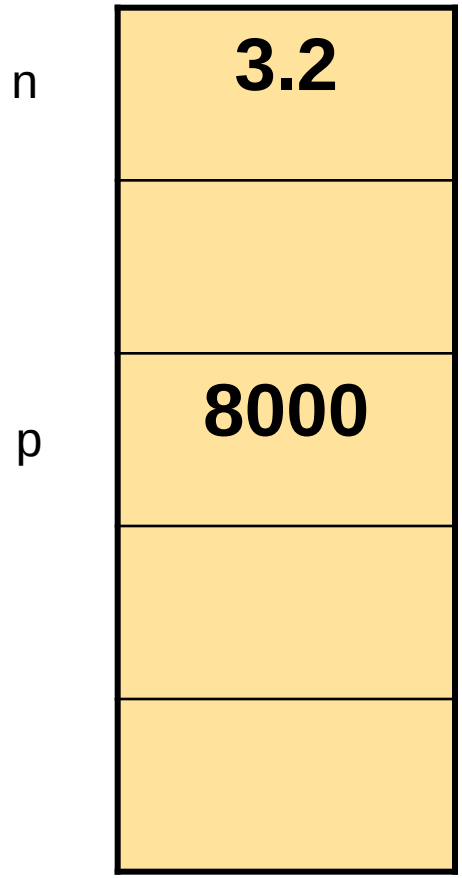
Osnove

- Pojam adrese
- Vrijednost sa poznate adrese uzimamo upotrebom operatora * (unarna *) - operator dereferenciranja ili indirekcije.
- Adresu neke vrijednosti/varijable uzimamo pomoću operatora & (operator adrese), koji znači “adresa od”
- Deklaracijom pokazivača određena je veličina vrijednosti koja će moći biti spremljena na nekoj adresi (npr. int, double, float).

Primjer

- Deklaracijom i inicijalizacijom
  - double n = 3.2;
- Rezervirana je memorija za varijablu n tipa double, npr. 8 bajtova. Početna adresa je npr. 8000, završna 8008 (8000 + sizeof(double)).
  - double* p;
- Deklarirali smo pokazivač p koji pokazuje na varijablu tipa double (bez obzira na tip podataka na koji pokazuje, pokazivač u memoriji zauzima istu količinu memorije).
- Pokazivaču p dodjeljujemo neku vrijednost (usmjeravamo ga da pokazuje na memorijsku lokaciju varijable tipa double):
  - p = &n;
- inicijalizirali smo pokazivač p tako da se na njegovoj adresi nalazi podatak koji govori koja je adresa varijable na koju pokazivač pokazuje

Stanje u memoriji



p = 8000

*p = 3.2;

p je adresa na koju pokazivač pokazuje, a *p vrijednost koja se nalazi na adresi na koju pokazivač pokazuje.

Isti učinak imaju:

n = 1.2 i *p = 1.2

Usporedba akcija

Primjer u svakodnevnom životu	Slična operacija u programu	C++ operacija	Komentar C++ operacije
Dohvat paketa s lokacije	**Dohvat vrijednosti sa adrese**	***pokaz_varijabla**	**Upotrebom unarnog * operatora**
Traženje adrese prijatelja	**Dohvat adrese varijable**	**&varijabla**	**Upotrebom operatora & u izvršnoj naredbi**
Određivanje veličine raspoloživog prostora na adresi	**Deklariranje tipa pokazivačke varijable na neku adresu**	**double* pokaz_varijabla;**	**Upotreba * u deklaraciji pokazuje da čuva adresu od *double*.**

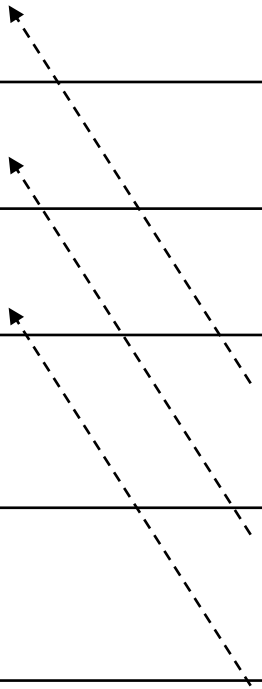
Primjer

```
#include <iostream>
using namespace std;

int main() {
   double vol, vis = 10.5, sir = 5.2, dub = 3.1;
   double* vis_adr;                  //tri pokaziv. na podatke tipa double
   double *sir _adr;                 //varijable za čuvanje adresa lokacija
   double* dub_adr;                  //na kojima je podatak tipa double!
   vis_adr = &vis;
   sir_adr = &sir;                   //pridruživanje adresa pokazivačima
   dub_adr = &dub;
   vol = vis * sir * dub;
   cout << "Volumen je " << vol << endl;
         //ili...
   vol = (*vis_adr) * (*sir_adr) * (*dub_adr);
   cout << "Volumen je " << vol << endl;
   return 0;
}
```

! Rezervira prostor za memoriranje, a zatim treba inicijalizirati ili pridružiti adresu pokazivačkoj varijabli

Tablica varijabli

Ime	Tip	Adresa	Vrijednost
vol	double	FFEE	169.26
vis	double	FFE6	10.5
sir	double	FFDE	5.2
dub	double	FFD6	3.1
vis_adresa	adresa od double	FFD4	FFE6
sir_adresa	adresa od double	FFD2	FFDE
dub_adresa	adresa od double	FFD0	FFD6



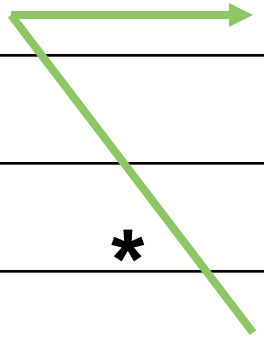
Upotreba varijabli tipa pokazivač

- Za pristup vrijednostima preko adrese koju čuvaju
- pomoću * uzimamo vrijednost, npr. *vis_adr
- uobičajeni sufix za pokazivače je ptr, npr. vis_ptr

- Operator *:
  - Binarni operator množenja
  - Kod deklaracije za označavanje da će u memorijsku ćeliju varijable biti spremljena adresa
  - Unarni operator koji označava pristup vrijednosti ili memorijskoj ćeliji na nekoj adresi
- Operator &:
  - Operator za logičku bitovnu "i operaciju" ('and')
  - U deklaraciji (npr. funkcije) za označavanje reference
  - U naredbi (ili funkcijskom pozivu) za označavanje "adresa od"

Prikaz & i * akcija &vis i *dub_adresa

Ime	Tip	Adresa	Vrijednost
vol	double	FFEE	169.26
vis	double	FFE6	10.5
sir	double	FFDE	5.2
dub	double	FFD6	3.1
vis_adr	adresa od doble	FFD4	FFE6
sir_adr	adresa od double	FFD2	FFDE
dub_adr	adresa od double	FFD0	FFD6



& : “adresa od”
*** : “dobavi vrijednost sa adrese”**

Dozvoljene operacije

- Sve operacije koje su dozvoljene izravno na cjelobrojnoj varijabli, mogu se provesti i preko pokazivača, npr:

- *p += 5; isto kao i n += 5;
- n = *p – 2; isto kao i n = n-2;

- Pokazivač se može inicijalizirati prilikom deklaracije, ali samo na objekt koji već postoji u memoriji.
- Inicijalizaciju pokazivača prilikom deklaracije moguće je izvesti i pomoću operatora new

Nedozvoljene operacije sa & i *

- &vis = FFE6;
  - C++ sam određuje adrese
- *vis = 10.5;
  - * se može koristiti samo uz pokazivače
- vis_adresa = 10.5;
  - vis_adresa može čuvati samo adresu
- Ako je deklaracija int* int_adresa, ne može se pisati int_adresa = &vis
  - Tip pokazivačke varijable mora odgovarati tipu podatka na adresi

Primjer: što će se ispisati?

```
int i = 1;
int j = 10;
int *p = &j;
*p *= *p;
i = i + j;
p = &i;
cout << i << endl << j << endl << *p <<
  endl;
```

Upotreba pokazivača pri radu s poljima

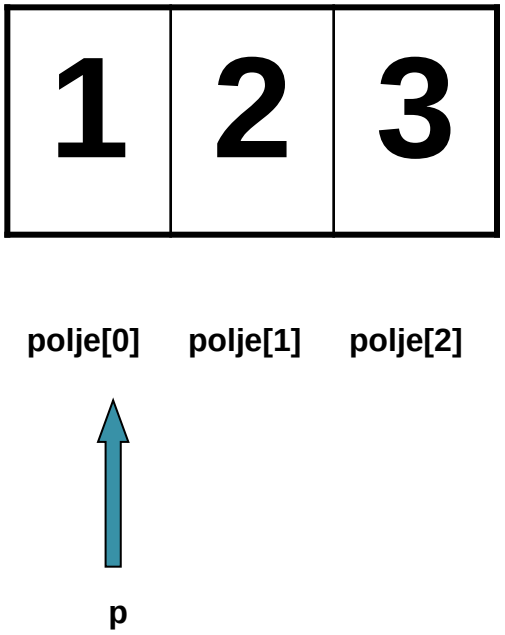
```
#include <iostream>
using namespace std;

int main() {
   long *p, polje[3] = {1, 2, 3};
   p = &polje[0];      //adresa nultog elementa polja
   for (int i = 0; i < 3; i++) {
         cout << "polje[" << i << "] = " << polje[i] << endl;
         cout << "*(p+" << i << ") = " << *(p+i) << endl;
   }
   return 0;
}
```

Primjer:

```
for (int i = 0; i < 3; i++) {
cout << *p++ << endl;}
```

da li je isto ako se napiše *(p+i)?

Pokazivači i polja



Pokazivači i polja – zadaci za utvrđivanje

- Da li je isto?
  - float x[5];float a = x[2]; float a=*(x+2);
- Da li se ispisuje isto?
  - int b[]={10,20,30}
  - cout << b << endl;
  - cout << *b << endl;
- Da li se ispisuje isto?
  - cout << &(b[1]) << endl;
  - cout << (b+1) << endl;
- Da li se ispisuje isto?
  - int x[5];
  - cout << x << endl;
  - cout << &x << endl;

Reference

- poseban tip podataka
- drugo ime za objekt određenog tipa
- novo ime za već postojeću varijablu
- djeluju i implementiraju se slično pokazivačima
- promjenom vrijednosti reference mijenja se vrijednost izvorne varijable

- *Primjer:*

```
int i = 5;
int &iref = i;//cjelobrojna referenca na varijablu i
iref = 75;
```

Usporedba referenci i pokazivača

- Pokazivač se može preusmjeriti na neki drugi objekt
- pri dohvaćanju vrijednosti preko pokazivača neophodan je operator *, a sam pokazivač iziskuje određeni memorijski prostor u kojem će biti pohranjena adresa na koju on pokazuje

- Referenca se inicijalizira prilikom deklaracije i pokazuje na dani *objekt* (mora postojati *objekt* na koji se referenca odnosi)
- nije potreban nikakav operator za dohvaćanje sadržaja
- najveću primjenu reference imaju kao argumenti i povratne vrijednosti funkcija