

Universiteti i Prishtinës  
Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike



# Algoritmet dhe struktura e të dhënave

Vehbi Neziri

FIEK, Prishtinë 2015



- Referencat
  - Referencat e zakonshme
  - Parametrat formalë referentë
  - Fushat referente
  - Referencat e zakonshme
  - Funkcionet me parametra referentë
  - Fushat referente

# Referencat e zakonshme

- Në C++, përmes referencave mund të deklarohen dy ose më shumë variabla ose objekte, tek të cilat ruhen të dhëna të njëjta, si dhe përcillen ndryshimet që ndodhin tek secila.
- Referencat kryesisht shfrytëzohen gjatë operimit me parametrat e funksioneve
- Për deklarimin e referencave shfrytëzohet operatori për referim **&**.

```
t &r = v;
```

**t** – tipi i të dhënave për variablën referente.

**&** - operatori për referim

**r** – variabla referente.

**v** - variabla së cilës i referohet variabla referente.

# Shembull 4.1

- Të shkruhet programi në të cilin deklarohet një variabël **v** e tipit integer si dhe variabla referente **vref**. Vlera e variablës **vref** të rritet për 1 dhe më pas të shtypet vlera e variablës **v**.

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

int main()
{
    int v=15;
    int &vref = v;
    v++;

    cout<< "Vlera e a-së është: "
         <<vref
         <<endl;

    system("pause");
    return 0;
}
```

# Parametrat formalë referentë

- Nëse pas tipit të parametrin formal të një funksioni shënohet simboli **&**, variabla përkatëse paraqet një *parametër formal referent*.
- Emri parametrin në trupin e funksion të thirrur i referohet variablës origjinale në funksion thirrës.
- Janë të dobishëm për kthimin e vlerave/rezultateve të shumëfishta.

Parametër  
referentë

```
void ndrysho(int &pref)
{
    pref = pref + 2;
}
```

Lexo: “**pref** është referencë në një **int**”

# Shembull 4.2

- Të shkruhet programi në të cilin deklarohet funksioni **param\_ref** i cili ka një parametër formalë referentë dhe parametrin formal e rrit për një. Nga funksioni kryesor të thirret funksioni **param\_ref** për dy variabla të deklaruara dhe iniciuara brenda funksionit kryesor.

```
1  #include <iostream>
2  using namespace std;
3
4  void param_ref(int &pref)
5  {
6      pref++;
7  }
8
9  int main()
10 {
11     int a=10;
12     int b=20;
13
14     param_ref(a) ;
15     param_ref(b) ;
16
17     cout<< "a = " << a
18         << " b = " << b
19         <<endl;
20
21     system("pause");
22     return 0;
23 }
```

# Fushat referente

- Gjatë operimit me anëtarët e fushave, si referente mund të deklarohen edhe fushat.
- Vlerat e anëtarëve të fushës referente barazohen me vlerat e anëtarëve të një fushe tjetër.

$$t \text{ (&R)[m]} = V;$$

$$t \text{ (&R)[m][n]} = M;$$

**t** – tipi i të dhënave për vektorin referent.

**&** - operatori për referim

**R** – vektori referent.

**m** – numri anëtarëve/rreshtave të vektorit/matricës referente.

**n** – numi i shtyllave të matricës referente.

**V** – vektori të cilit i referohet vektori referent.

**M** – matrica të cilës i referohet matrica referente.

## Shembull 4.3

- Të shkruhet programi përmes të cilit tregohet deklarimi i vektorit **B(m)** si vektor referent i vektorit **A(m)**. Vektori **A(m)** gjatë deklarimit të iniciohet me vlera.

```
1 #include<iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main()
6 {
7     const int m=5;
8     int A[m]={2,5,8,9,1};
9     int (&B)[m]=A; //vektori referent
10
11     cout << "Vektori origjinal A\n\n";
12     for (int i=0;i<m;i++)
13         cout << setw(3)
14             << A[i];
15     cout << "\n\nVektori referent B\n\n";
16     for (int i=0;i<m;i++)
17         cout << setw(3)
18             << B[i];
19     cout << "\n\n";
20
21     system("pause");
22     return 0;
23 }
```





- Parametrat e funksioneve që shfrytëzohen brenda komponentëve të strukturave mund të merren si parametra referentë.
- Kjo ka rëndësi të veçantë kur prej strukturës merren rezultatet e llogaritjeve të ndryshme.

```
struct rrethi
{
    double r,s,p;
    void lexo(double &x,double &y);
    //...
};
```

# Shembull 4.4

- Të shkruhet programi në të cilin shfrytëzohet struktura **rrethi** me variablat **r**, **s**, dhe **p**. Të llogaritet sipërfaqja dhe perimetri i rrethit dhe rezultati të ruhet në variablat e strukturës **s** dhe **p**. Për të lexuar vlerat e variablave **s** dhe **p** të shfrytëzohet funksioni **lexo** me parametrat referentë **x** dhe **y**.

```
1 #include<iostream>
2 using namespace std;
3
4 const double pi=3.1415;
5 struct rrethi
6 {
7     double r,s,p;
8     void vendos_rrezen();
9     void llogarit();
10    void lexo(double &x,double &y);
11 };
12 int main()
13 {
14     rrethi rr;
15     double s,p;
16     rr.vendos_rrezen();
17     rr.llogarit();
18     rr.lexo(s,p);
19     cout<< "\nSipërfaqja s=" << s
20     << "\nPerimetri p="<< p
21     << "\n\n";
22     system("pause");
23     return 0;
24 }
25 void rrethi::vendos_rrezen()
26 {
27     cout << "\nRrezja r=";
28     cin >> r;
29 }
30 void rrethi::llogarit()
31 {
32     s=pi*r*r;
33     p=2*pi*r;
34 }
35 void rrethi::lexo(double &x,double &y)
36 {
37     x=s;
38     y=p;
39 }
```

# Variablat referente brenda klasave

- Ngjashëm sikurse tek strukturat, edhe tek klasat mund të shfrytëzohen parametra referentë.
- Parametrat referentë janë si parametra të funksioneve brenda anëtarëve të tyre.

```
class rrethi
{
private:
    double r,s,p;
public:
    void lexo(double &x,double &y);
    //...
};
```

# Shembull 4.5

- Të shkruhet programi në të cilin shfrytëzohet klasa **rrethi** me anëtarët privat **r**, **s**, dhe **p**. Të llogaritet sipërfaqja dhe perimetri i rrethit dhe rezultati të ruhet në anëtarët privat të klasës, **s** dhe **p**. Për të lexuar vlerat e variablave **s** dhe **p** të shfrytëzohet funksioni **lexo** me parametrat referentë **x** dhe **y**.  
**(Modifikim i shembullit paraprak)**

```
1 #include<iostream>
2 using namespace std;
3
4 const double pi=3.1415;
5 class rrethi
6 {
7 private:
8     double r,s,p;
9 public:
10     void vendos_rrezen();
11     void llogarit();
12     void lexo(double &x,double &y);
13 };
14 int main()
15 {
16     rrethi rr;
17     double s,p;
18     rr.vendos_rrezen();
19     rr.llogarit();
20     rr.lexo(s,p);
21     cout<< "\nSipërfaqja s=" << s
22     << "\nPerimetri p="<< p
23     << "\n\n";
24     system("pause");
25     return 0;
26 }
27 void rrethi::vendos_rrezen()
28 {
29     cout << "\nRrezja r=";
30     cin >> r;
31 }
32 void rrethi::llogarit()
33 {
34     s=pi*r*r;
35     p=2*pi*r;
36 }
37 void rrethi::lexo(double &x,double &y)
38 {
39     x=s;
40     y=p;
41 }
```

# Objektet referente

- Objektet e strukturave dhe të klasave mund të deklarohen si objekte referente plotësisht njëllor, siç deklarohen edhe variablat e zakonshme.
- Si rezultat, të dhënat që u shoqërohen anëtarëve të strukturave ose të klasave do të barazohen me vlerat e anëtarëve përkatës në objektet të cilave u referohen.

## Shembull 4.6

- Të shkruhet programi në të cilin është deklaruar struktura **librat** me anëtarët: **titulli**, **autori** dhe funksioni **shtypja**. Të deklarohet objekti referent **libRef** i objektit **libri** dhe më pas të thirret funksioni **shtypja** përmes objektit **libri** dhe **libRef**.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct librat
6 {
7     string titulli;
8     string autori;
9     void shtypja();
10 };
11
12 void librat::shtypja()
13 {
14     cout<<"\nTitulli i librit: "
15         <<titulli
16         <<"\nAutori i librit: "
17         <<autori;
18 }
19
20 int main()
21 {
22     librat libri;
23     cout<<"Titulli i librit: ";
24     getline(cin,libri.titulli);
25
26     cout<<"Autori i librit: ";
27     getline(cin,libri.autori);
28
29     cout<<"Shtypja përmes objektit libri\n";
30     libri.shtypja();
31
32     librat &libRef=libri;
33
34     cout<<"Shtypja përmes objektit libRef\n";
35     libRef.shtypja();
36
37     system("pause");
38     return 0;
39 }
```

