Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Algoritmet dhe struktura e të dhënave

Vehbi Neziri

FIEK, Prishtinë 2013

- Numërimet (grupimet)
  - Degëzimi përmes vlerave të numëruara
  - Disa variabla të numëruara të tipit të njëjtë
  - Përcaktimi dhe deklarimi njëkohësisht
  - Shoqërimi direkt i vlerave
  - Operimi me variabla të numëruara
  - Leximi i të dhënave të numëruara
  - Definimi i disa grupeve njëkohësisht
  - Të dhënat e numëruara në nënprograme

Degëzimi përmes vlerave të numëruara

- Vlerat e numëruara mund të shfrytëzohen për realizimin e degëzimeve duke përdorur komandën if, ose switch.

```
enum gjendjet
{
    zero,
    nje
};
```

```
int main()
{
    gjendjet gjendja;

    if (gjendja==zero)
        cout << "\nTensioni eshte 0V";
    else
        cout << "\nTensioni eshte 5V";
    return 0;
}
```

Shembull 4.1

- Nëse është dhënë grupi gjendjet, të shkruhet programi në të cilin deklarohet variabla e numëruar nota dhe përmes komandës if të shtypet rezultati.

```
enum gjendjet
{
    dobet,
    mire,
    shumemire,
    shkelqyeshem
};
```

```
1  #include<iostream>
2  using namespace std;
3
4  enum gjendjet
5  {
6      dobet,
7      mire,
8      shumemire,
9      shkelqyeshem
10 };
11
12 int main()
13 {
14     gjendjet nota;
15     nota=shkelqyeshem;
16     if (nota==dobet)
17         cout << "\nKeni arritur sukses t% dob%t\n";
18     else if (nota==mire)
19         cout << "\nKeni arritur sukses t% mir%\n";
20     else if (nota==shumemire)
21         cout << "\nKeni arritur sukses shum% t% mir%\n";
22     else if (nota==shkelqyeshem)
23         cout << "\nKeni arritur sukses t% shkelqyesh%m\n";
24
25     system("pause");
26     return 0;
27 }
```

Shembull 4.2

- Nëse është dhënë grupi java, të shkruhet programi i cili përmes komandës switch dhe variablës së numëruar dita, shtyp rezultatin për ditën e zgjedhur.

```
enum java
{
    hene,
    marte,
    merkure,
    enjte,
    premte,
    shtune,
    diele
};
```

```
1  #include <iostream>
2  using namespace std;
3
4  enum java
5  {
6      hene,
7      marte,
8      merkure,
9      enjte,
10     premte,
11     shtune,
12     diel
13 };
14
15 int main()
16 {
17     java dita;
18     dita=enjte;
19     cout << "\nDita q‰ u zgjodh ‰sht‰ dita e ";
20     switch (dita)
21     {
22         case hene: cout << "h‰n‰";break;
23         case marte: cout << "mart‰";break;
24         case merkure: cout << "m‰rkur‰";break;
25         case enjte: cout << "‰njte";break;
26         case premte: cout << "premte";break;
27         case shtune: cout << "shtun‰";break;
28         case diel: cout << "diel";break;
29     }
30     cout << "\n\n";
31     system("pause");
32     return 0;
33 }
```

Disa variabla të numëruara të tipit të njëjtë

- Sikurse të variablat e tipave standard, brenda një programi mund të deklarohen edhe disa variabla të numëruara të tipit të njëjtë.
- Sintaksa:

```
e v1, v2, v3, . . . ;
```

e – emri i grupit
v1, v1, v3… – variablat e deklaruara të grupit

Shembull 4.3

- Të shkruhet programi në të cilin deklarohet grupi java me ditët e javës. Përmes variablave punuese dhe pushuese, të tregohet se me cilin numër fillojnë ditët e punës dhe me cilin numër fillojnë ditët e pushimit (fundjavës).

```
1  #include <iostream>
2  using namespace std;
3
4  enum java
5  {
6      hene,
7      marte,
8      merkure,
9      enjte,
10     premte,
11     shtune,
12     diel
13 };
14
15 int main()
16 {
17     java punuese;
18     java pushuese;
19
20     punuese=hene;
21     pushuese=shtune;
22     cout << "\nDit%t e pun%s fillojn% me numrin: "
23     << punuese
24     << "\n\nDit%t e pushimit fillojn% me numrin: "
25     << pushuese
26     << "\n\n";
27     system("pause");
28     return 0;
29 }
```

Përcaktimi dhe deklarimi njëkohësisht

- Gjatë përcaktimit të grupit me të dhëna të numëruara, mund të bëhet edhe deklarimi i variablës përkatëse.

```
enum e
{
   a0,
   a1,

   . . .

   an
}
var;
```

```
enum  ylberi
{
     kuqe,
     portokalli,
     verdhe,
     gjelber,
     kalter,
     indigo,
     vjollce
}
ngjyra;
```

e – emri i grupit

a_0, a_1, …a_n – anëtarët e grupit

var - variabla e deklaruar e tipit e.

Shembull 4.4

- Të shkruhet programi në të cilin përcaktohet grupi viti me vlera stinët e vitit. Gjatë përcaktimit të deklarohet variabla stina. Variabla të iniciohet në funksionin main dhe me pas në varësi të vlerës së iniciuar të shfaqet mesazhi i formuar "Stina e ...".

```
1  #include <iostream>
2  using namespace std;
3
4  enum viti
5  {
6      pranvera,
7      vera,
8      vjeshta,
9      dimri
10 }
11 stina;
12
13 int main()
14 {
15     stina=pranvera;
16     if (stina==pranvera)
17         cout << "\nStina e pranver%s";
18     else if (stina==vera)
19         cout << "\nStina e ver%s";
20     else if (stina==vjeshta)
21         cout << "\nStina e vjesht%s";
22     else
23         cout << "\nStina e dimrit";
24     cout << "\n\n";
25     system("pause");
26     return 0;
27 }
```

Shoqërimi direkt i vlerave

- Gjatë përcaktimit të grupit, mund të bëhet edhe deklarimi i variablës përkatëse.

```
enum e
{
   a0=k0,
   a1=k1,

   . . .

   an=kn
};
```

```
enum  ylberi
{
     kuqe=10,
     portokalli=20,
     verdhe=30,
     gjelber=40,
     kalter=50,
     indigo=60,
     vjollce=70
};
```

e – emri i grupit

$\mathbf{a_0, a_1, \ldots a_n}$ – anëtarët e grupit

$\mathbf{k_0, k_1, \ldots k_n}$ - konstante që u shoqërohen anëtarëve të grupit

Shembull 4.5

- Të shkruhet programi dhe të deklarohet një grup me emrin nota i cili si anëtarë ka notat nga pesë deri në dhjetë. Secilit anëtar të i shoqërohet vlera numerike. Më pas të shtypet rezultati për notat në algoritme, programim dhe matematikë.

```
1  #include <iostream>
2  using namespace std;
3
4  enum nota
5  {
6      pese=5,
7      gjashte=6,
8      shtate=7,
9      tete=8,
10     nente=9,
11     dhjete=10
12 }algoritme, matematike, programim;
13
14 int main()
15 {
16     matematike=shtate;
17     programim=tete;
18     algoritme=nente;
19
20     cout << "\nNota n% Matematik% %sht% "
21          << matematike
22          << "\nNota n% Algoritme %sht% "
23          << algoritme
24          << "\nNota n% Programim %sht% "
25          << programim
26          << "\n";
27     system("pause");
28     return 0;
29 }
```

Operimi me variabla të numëruara dhe leximi

- Variabla e numëruar mund t'i shoqërohet një variable tjetër të numëruar, përmes operatorit të barazimit.

```
v1 = v2;
```

- Për të shfrytëzuar variablat e numëruara në shprehje të ndryshme duhet të bëhet konvertimi eksplicit përmes static_cast.

```
algoritme=static_cast<nota>(algoritme+2);
```

- Vlerat e të dhënave të numëruara lexohen në mënyrë indirekte përmes variablave.

Shembull 4.6

- Të shkruhet programi dhe të deklarohet një grup me emrin nota i cili si anëtarë ka notat nga pesë deri në dhjetë. Secilit anëtar të i shoqërohet vlera numerike. Vlera fillestare për notën në algoritme të lexohet nga tastiera dhe përmes komandës switch të shtypet nota, më pas të rritet nota për 2 dhe të shtypet përsëri.

```
1  #include <iostream>
2  using namespace std;
3
4  enum nota
5  {
6      pese=5,
7      gjashte=6,
8      shtate=7,
9      tete=8,
10     nente=9,
11     dhjete=10
12 }algoritme;
13
```

Shembull 4.6

- Të shkruhet programi dhe të deklarohet një grup me emrin nota i cili si anëtarë ka notat nga pesë deri në dhjetë. Secilit anëtar të i shoqërohet vlera numerike. Vlera fillestare për notën në algoritme të lexohet nga tastiera dhe përmes komandës switch të shtypet nota, më pas të rritet nota për 2 dhe të shtypet përsëri.

```
14  int main()
15  {
16      int vlera;
17      cout << "\nSa %sht% nota n% Algoritme: ";
18      cin >> vlera;
19
20      cout << "\nNota n% Algoritme %sht% ";
21      switch (vlera)
22      {
23          case 6:
24              algoritme=gjashte;
25              cout << algoritme;break;
26          case 7:
27              algoritme=shtate;
28              cout << algoritme;break;
29          case 8:
30              algoritme=tete;
31              cout << algoritme;break;
32          case 9:
33              algoritme=nente;
34              cout << algoritme;break;
35          case 10:
36              algoritme=dhjete;
37              cout << algoritme;break;
38          default:
39              cout << "Jo kaluese";break;
40      }
41
42      algoritme=static_cast<nota>(algoritme+2);
43
44      cout<< "\nNota n% Algoritme %sht% rritur p%r 2 "
45      <<"dhe tani %sht%:" << algoritme
46      << "\n";
47      system("pause");
48      return 0;
49  }
```

Definimi i disa grupeve njëkohësisht

- Në një program mund të definohen dhe të shfrytëzohen disa grupe në të njëjtën kohë.

```
enum moti
{
    kthjellet,
    vranet,
    ere,
    ftohte
};
```

```
enum koha
{
    diell,
    shi,
    bore
};
```

- Në një program nuk lejohen grupime në të cilat përfshihen të dhëna të njëjta:

```
enum obligative
{
    Matematika,
    Fizika,
    Kompjuterika,
    Kimia
};
```

```
enum zgjedhore
{
    Programimi,
    Matematika,
    Algoritme
};
```

Shembull 4.7

- Të shkruhet programi në të cilin deklarohen dy grupe me emrin moti dhe koha. Grupi moti të përmbajë gjendjet: *kthjellet*, *vranet*, *ere*, *ftohte* dhe *diell*; kurse grupi koha të përmbajë: *mjegull*, *shi* dhe *bore*. Duke përdorur komandën if të kontrollohen disa gjendje dhe të shtypet mesazhi sipas modelit të mëposhtëm.

```
Moti është i ftohtë dhe me shi
```

```
1  #include <iostream>
2  using namespace std;
3
4  enum moti
5  {
6      kthjellet,
7      vranet,
8      ere,
9      ftohte,
10     diell
11 } mt;
12
13 enum koha
14 {
15     mjegull,
16     shi,
17     bore
18 } kh;
19
20 int main()
21 {
22     mt=ftohte;
23     kh=bore;
24
25     if (mt ==ftohte && kh== shi)
26     {
27        cout<<"Moti %sht% i ftoht% dhe me shi\n";
28     }
29     else if (mt ==ftohte && kh== bore)
30     {
31         cout<<"Moti %sht% i ftoht% dhe me bor%\n";
32     }
33     system("pause");
34     return 0;
35 }
```

Të dhënat e numëruara në nënprograme

- Të dhënat e numëruara mund të përdoren edhe si parametra të funksioneve.

```
void funksioni(enum grupi);
```

Shembull 4.8

- Të shkruhet programi në të cilin deklarohet një grup me emrin Motori si dhe funksioni Kontrollo i cili ka si parametër hyrës grupin Motori. Grupi përmban dy vlera ON dhe OFF. Funksioni Kontrollo të thirret për ON dhe për OFF.

```
1  #include <iostream>
2  using namespace std;
3
4  enum Motori
5  {
6      OFF,
7      ON
8  } statusi;
9
10 void Kontrollo(enum Motori);
11
12 int main()
13 {
14     statusi = OFF;
15     Kontrollo( statusi);
16
17     statusi = ON;
18     Kontrollo( statusi);
19
20     system("pause");
21     return 0;
22 }
23
24 void Kontrollo(Motori gjendja)
25 {
26    if (gjendja==ON)
27     cout<<"Motori %sht% i ndezur\n";
28    else
29         cout<<"Motori %sht% i fikur\n";
30 }
```

Detyra: 4.1

- Nëse deklarimet e mëposhtme janë pjesë e një programi të rregullt, çfarë do të shtypet pas ekzekutimit.

```
enum drejtimi { V, J, L, P };
// ...
cout << P << " " << L << " "
    << J << " " << V << endl;
```

```
enum drejtimi { V=3, J, L=6, P };
// ...
cout << P << " " << L << " "
    << J << " " << V << endl;
```

Detyra: 4.2

- Të shkruhet programi dhe të deklarohet një grup me emrin nota i cili si anëtarë ka notat nga pesë deri në dhjetë. Secilit anëtar të i shoqërohet vlera numerike. Më pas të shtypet rezultati për notat në algoritme, programim dhe matematikë si dhe të tregohet se në cilën lëndë është nota më e lartë.

Detyra: 4.3

- Të shkruhet programi dhe të deklarohet një grup me emrin nota i cili si anëtarë ka notat nga pesë deri në dhjetë. Secilit anëtar të i shoqërohet vlera numerike. Vlera fillestare për notën në programim të lexohet nga tastiera. Përmes komandës if të shtypet nota e lexuar nga tastiera, më pas të rritet nota për 1 dhe të shtypet përsëri.

Detyra: 4.4

- Të shkruhet programi në të cilin deklarohen dy grupe me emrin ekzekutuar dhe funksionimi. Grupi ekzekutuar të përmbajë gjendjet: *PO* dhe *JO*; kurse grupi funksionimi të përmbajë: *rregull dhe* gabim. Duke përdorur komandën if të kontrollohet se a është ekzekutuar programi dhe a është në rregull apo ka gabime.

Detyra: 4.5

- Nëse është dhënë grupi, të shkruhet programi i cili i mundëson përdoruesit të shtyp shkronjën e parë (a, s, m, k, p), dhe tipin e zgjedhur ta ruajë si vlerë të variablës së tipit enum. Në fund përmes funksionit shtypja i cili si parametër ka grupin të shtypet emri i plotë e pozitës.

```
enum pozita
{
    asistent,
    sekretar,
    menagjer,
    kontabilist,
    programues
};
```

Pyetje ?