



PROGRAMI I ORIENTUAR NE OBJEKTE

Trashëgimia - Inheritance

TRASHËGIMIA - INHERITANCE



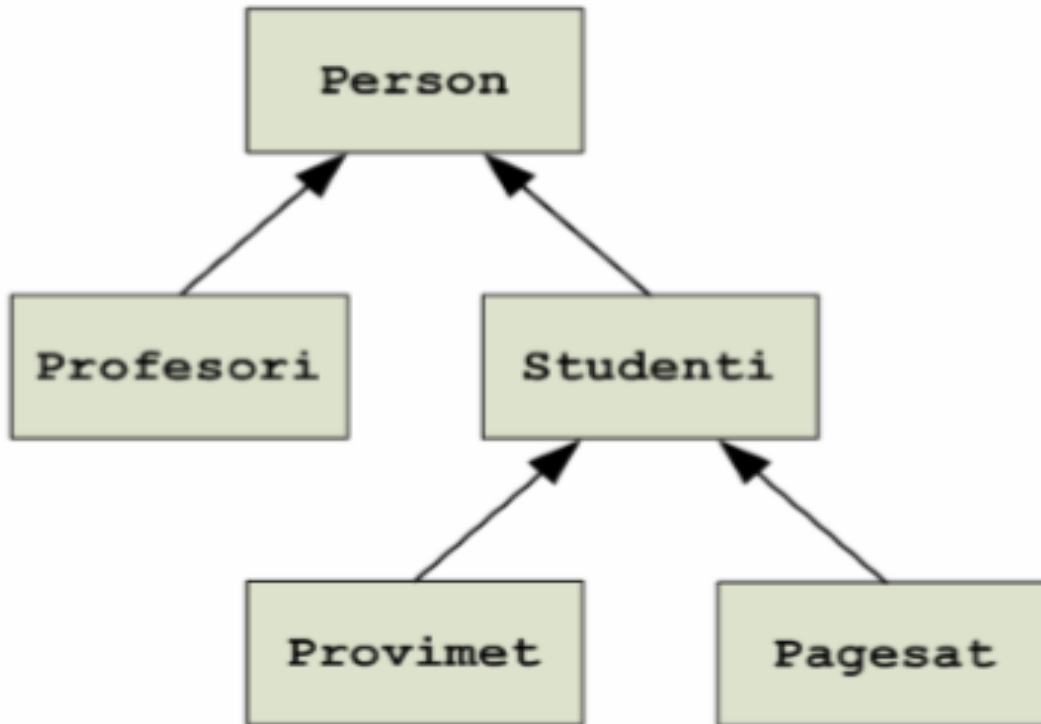
- Përmes mekanizmit të trashëgimisë (ang. inheritance), anëtarët e klasës ekzistuese mund të shfrytëzohen gjatë definimit të klasave të reja.
- Klasa ekzistuese njihet si klasë bazë (ang. base class), kurse klasa e re i trashëgon karakteristikat e klasës bazë dhe quhet klasë e nxjerrë (ang. derived class).
- Klasa e re krijohet duke e plotësuar klasën ekzistuese me anëtarë të tjerë, qofshin ato anëtarë të dhënash ose funksione.

TRASHËGIMIA - INHERITANCE



- ⦿ Gjatë kësaj, klasa bazë nuk pëson asnjë ndryshim.
- ⦿ Njëkohësisht, klasa e nxjerrë mund të shfrytëzohet si klasë bazë për nxjerrjen e klasave të tjera.
- ⦿ Një mundësi e tillë e krijimit të klasave të reja, si prej klasave bazë ashtu edhe prej klasave të nxjerra, përmes mekanizmit të trashëgimisë, paraqet një organizim hierarkik të klasave, gjë që programimin e orientuar në objekte e bën edhe më të fuqishëm.

DUKJA E NJË ORGANIZIMI HIERARKIK TË KLASAVE



DEFINIMI I KLASËS BAZË DHE KLASËS SË NXJERRË PREJ SAJ



```
1 #include <iostream>
2 using namespace std;
3 class KlasaBaze
4 {
5 private:
6     int a;
7 public:
8     double b;
9     void KB()
10    {
11        a=5;
12        b=8;
13        cout << "\nVlerat te klasa Baze"
14              << "\n\n    a="
15              << a
16              << "\n    b="
17              << b
18              << "\n\n";
19    }
20 };
```

1

```
20 class Klasa_e_Nxjerre :public KlasaBaze
21 {
22 private:double c;
23 public:int d;
24 void KN()
25 {
26     b=2;
27     c=4;
28     d=7;
29     cout << "Vlerat te klasa e Nxjerre"
30          << "\n\n    b="
31          << b
32          << "\n    c="
33          << c
34          << "\n    d="
35          << d
36          << "\n";
37 }
38 };
```

2

```
39 int main()
40 {
41     KlasaBaze X;
42     Klasa_e_Nxjerre Y;
43     X.KB();
44     Y.KN();
45     X.KB();
46
47     cin.get();
48     return 0;
49 }
```

3

SHFRYTËZIMI I ANËTARËVE GJATË TRASHËGIMISË



- Anëtarët e klasës bazë bëhen edhe anëtarë të klasës së nxjerrë. Te klasa e nxjerrë mund t'u qasemi direkt anëtarëve publikë dhe anëtarëve të mbrojtur të klasës bazë.
- Por, nuk lejohet qasja direkte tek anëtarët e mbrojtur të klasës bazë, prej klasës së nxjerrë.

SHFRYTËZIMI I ANËTARËVE GJATË TRASHËGIMISË



- ◉ Qasja direkte te anëtarët privat të klasës bazë është e mundshme vetëm brenda klasës bazë.
- ◉ Anëtarët publikë të klasës bazë trashëgohen te klasa e nxjerrë si anëtarë publikë, kurse anëtarët privatë ose të mbrojtur, varësisht nga specifikuesi i qasjes që zgjedhet gjatë definimit të klasës së nxjerrë.

SHFRYTËZIMI I ANËTARËVE GJATË TRASHËGIMISË



1.

```
class Bazë
{
    ...
};
class KNxjerrë:
    public Bazë
{
    ...
};
```

- Këtu, anëtarët publikë të klasës *Bazë* trashëgohen te klasa e nxjerrë *KNxjerrë* si anëtar publik.

2.

```
class Bazë
{
    ...
};
class KNxjerrë:
    private Bazë
{
    ...
};
```

- Në këtë rast, anëtarët publikë të klasës *Bazë* te klasa e nxjerrë *Knxjerrë* trashëgohen si anëtar privat.



SHFRYTËZIMI I ANËTARËVE GJATË TRASHËGIMISË

3.

```
class Bazë
{
    ...
};
class Knxjerrë: Bazë
{
    ...
};
```

- ◉ Meqë këtu nuk është shënuar specifikuesi i qasjes, kompjuteri e nënkupton atë si private.
- ◉ Prandaj, në këtë rast, si edhe në rastin paraprak, anëtarët publikë të klasës *Bazë* trashëgohen si anëtarë privatë të klasa e nxjerrë *KNxjerrë*.

RIDEFINIMI I FUNKSIONEVE TË KLASËS BAZË



- ⦿ Funkcionet në komponentet publike të klasës bazë mund të ridefinohen te klasat e nxjerra prej saj.
- ⦿ Përkatësisht, në klasën e nxjerrur mund të ridefinohen funksione me emra të njëjtë, numër dhe tip të njëjtë të parametrave.
- ⦿ Por, ky ridefinim i funksioneve vlen vetëm për klasën e nxjerrë dhe jo edhe për klasën bazë.

RIDEFINIMI I FUNKSIONEVE TË KLASËS BAZË



- ◉ Përmes shembullit të ardhshëm tregohet ridefinimi te klasa e nxjerrë *Klasa_e_Nxjerrë* i funksionit **KB**, përmes së cilit te klasa *Klasa_Bazë* llogaritet vlera e shprehjes:

$$g = 3a + 2b - 1$$

- ◉ kurse te klasa e nxjerrë *Klasa_e_Nxjerrë* vlera në fjalë llogaritet me shprehjen:

$$g = 2b + 3c - d$$

- ◉ Llogaritja e vlerës së variablës **g**, si në klasën bazë *Klasa_Bazë* ashtu edhe në klasën e nxjerrë *Klasa_e_Nxjerrë*, është definuar brenda funksionit **KB**, i cili përfshihet në komponentet publike përkatëse.



```
4 class Klasa_Baze
5 {
6     public: double a,b,g;
7           Klasa_Baze();
8           void KB();
9 };
10
11 class Klasa_e_Nxjerre:public Klasa_Baze
12 {
13     private: double c;
14     public: int d;
15           Klasa_e_Nxjerre();
16           void KB();
17 };
18
19 int main()
20 {
21     Klasa_Baze X;
22     Klasa_e_Nxjerre Y;
23
24     X.KB();
25     cout << "\nVlera e llogaritur te klasa Baze g="
26          << X.g;
27     Y.KB();
28     cout << "\n\nVlera e llogaritur te klasa e Nxjerre g="
29          << Y.g
30          << "\n\n";
31
32     cin.get();
33     return 0;
34 }
```



```
35  [= Klasa_Baze::Klasa_Baze()
36  {
37      a=5;
38      b=8;
39  }
40  [= Klasa_e_Nxjerre::Klasa_e_Nxjerre()
41  {
42      c=4;
43      d=7;
44      cout << "\nVlerat e variablave"
45           << "\n\n    a="
46           << a
47           << "\n    b="
48           << b
49           << "\n    c="
50           << c
51           << "\n    d="
52           << d
53           << "\n";
54  }

55  [= void Klasa_Baze::KB()
56  {
57      g=3*a+2*b-1;
58  }
59  [= void Klasa_e_Nxjerre::KB()
60  {
61      g=2*b+3*c-d;
62  }
```



- ◉ Shembulli i ardhshëm tregon shoqërimin e vlerave të variablave të komponenteve të objektit *kater1* të klasës brinjet, variablave përkatëse të komponenteve të objektit *kater2*.



```
1 #include <iostream>
2 using namespace std;
3 class brinjet
4 {
5 private: double a,b;
6 public:
7     void vlerat(double x,double y);
8     void shtypja(); };
9
10 int main()
11 {
12     brinjet kater1,kater2;
13     kater1.vlerat(7.5,4.2);
14     cout << "\nVlerat origjinale";
15     kater1.shtypja();
16     kater2=kater1;
17     cout << "Vlerat e kopjuara";
18     kater2.shtypja();
19
20     cin.get();
21     return 0;
22 }
23
24 void brinjet::vlerat(double x,double y)
25 {
26     a=x; b=y;
27 }
28 void brinjet::shtypja()
29 {
30     cout << "\n\n    a="
31         << a
32         << "\n    b="
33         << b
34         << "\n\n";
35 }
```

KLASAT E NDËRTHURURA



- ⦿ Brenda klasave mund të definoohen klasa të tjera, për të krijuar struktura të klasave të ndërthurura.
- ⦿ Te strukturat e tilla, klasa në të cilën përfshihet një klasë njihet si klasë e jashtme, kurse klasa e përfshirë në të njihet si klasë e brendshme.
- ⦿ Gjatë kësaj, klasat e brendshme mund të vendosen në pjesën private ose në pjesën publike të klasës së jashtme.



BRENDA KOMPONENTES PUBLIKE TË KLASËS SË JASHTME KJ DEFINOHET KLASA E BRENDSHME KB.

```
3 class KJ
4 {
5     private: void MesazhiA();
6     public:  int a;
7             void vleraA(int x);
8
9     class KB
10    {
11        private: void MesazhiB();
12        public:  int b;
13                void vleraB(int y);
14    };
15 };
17 int main()
18 {
19     KJ Alfa;
20     KJ::KB Beta;
21     Alfa.vleraA(55);
22     cout << "\nVlera e variablës a="
23          << Alfa.a; Beta.vleraB(33);
24     cout << "\nVlera e variablës b="
25          << Beta.b
26          << "\n\n";
27
28     cin.get();
29     return 0;
30 }
```

```
31 void KJ::MesazhiA()  
32 {  
33     cout << "\nKlasa e jashtme KJ\n";  
34 }  
35 void KJ::vleraA(int x)  
36 {  
37     a=x;  MesazhiA();  
38 }  
39 void KJ::KB::MesazhiB()  
40 {  
41     cout << "\n\n\nKlasa e brendshme KB\n";  
42 }  
43 void KJ::KB::vleraB(int y)  
44 {  
45     b=y;  MesazhiB();  
46 }
```

