

FEJLETT PROGRAMOZÁSI NYELVEK, 2009

7. GYAKORLAT – Származtatás

Cél:

- származtatás
- tiszta virtuális tagfüggvények
- saját kivételosztály készítése
- absztrakt osztály készítése

I.Szemely.h, Szemely.cpp, Alkalmazott.h, Alkalmazott.cpp, Manager.h, Manager.cpp, main.cpp

Készítse el a következő típusokat úgy, hogy a megadott main függvény helyesen működjön és a megadott kimenetet eredményezze.

A Szemely típus attribútumai: vezetéknev, keresztnév és születési év. Az Alkalmazott típus tartalmazza a munkakör attribútumot is, illetve a Manager típusnak van egy saját titkára, amelyet Alkalmazott típusként adunk meg.

```
class Szemely{
//Kód
};

class Alkalmazott: public Szemely{
//Kód
};

class Manager:public Alkalmazott{
//Kód
};
```

```
#include <cstdlib>
#include "Szemely.h"
#include "Alkalmazott.h"
#include "Manager.h"
#include <iostream>
#include <list>
using namespace std;

int main(int argc, char** argv) {
    Szemely sz1("Papa", "Daniel", 1980), sz2("Bodo","Maria", 1986);
    cout<<sz1<<endl;
    cout<<sz2<<endl;

    Alkalmazott a1("Smith", "John", 2000, "beszerzo");
    Alkalmazott a2("Biro", "Erzsebet", 2001, "titkarno");
    cout<<a1<<endl;
    cout<<a2<<endl;
    Manager m("Biro","Istvan",1965,"manager", a2);

    cout<<m<<endl;
```

FEJLETT PROGRAMOZÁSI NYELVEK, 2009

7. GYAKORLAT – Származtatás

```
list<Szemely*> l;
l.push_back(&sz1); l.push_back(&sz2);
l.push_back(&a1); l.push_back(&a2);
l.push_back(&m);
list<Szemely*>::iterator it;
cout<<endl<<"Szemelyek halmaza: "<<endl;
for( it = l.begin(); it != l.end(); ++it )
    cout<<*it<<endl;
return (EXIT_SUCCESS);
}
```

Kimenet:

```
Popa Daniel 1980
Bodo Maria 1986
Smith John 2000 beszerzo
Biro Erzsebet 2001 titkarno
Biro Istvan 1965 manager Biro Erzsebet 2001
```

```
Szemelyek halmaza:
Popa Daniel 1980
Popa Daniel 1980
Smith John 2000 beszerzo
Biro Erzsebet 2001 titkarno
Biro Istvan 1965 manager Biro Erzsebet 2001
```

II. Halmaz - HalmazException.h, AsztraktHalmaz.h, Halmaz.h, Main.cpp

1. Készítsen egy **AbsztraktHalmaz** nevű absztrakt osztályt, amely a következő matematikai halmazokon értelmezett műveleteket deklarálja:

- halmaz elemszámának lekérdezése
- adott elem keresése a halmazban
- új elem felvétele a halmazba
- adott elem törlése a halmazból
- adott pozíció levő elem lekérdezése

A műveleteket úgy kell deklarálni, hogy bármilyen típusra érvényesek legyenek (osztálysablon). Készítsen saját kivételosztályt is. Kivételt dobhat az új elem felvétele abban az esetben, ha már létező elemet akarunk felvenni, nem létező elem törlése esetén és nem létező pozícióról való lekérdezés esetében.

```
class HalmazException{
    string message;
public:
    HalmazException( string message ):message( message ){}
    friend ostream& operator<<( ostream& os, HalmazException e ){
        return os<<e.message<<endl;
    }
};

template <class T>
class AbsztraktHalmaz{
```

FEJLETT PROGRAMOZÁSI NYELVEK, 2009

7. GYAKORLAT – Származtatás

```
public:
    virtual int getSize() const =0;
    virtual bool find( const T& e) =0;
    virtual void add( const T& e ) throw (HalmazException) =0;
    virtual void remove( const T& e ) throw (HalmazException) =0;
    virtual T get( int index ) const throw (HalmazException) =0;
};
```

2. Készítsen egy **Halmaz** osztályt, amely az **AbsztraktHalmaz** osztály leszármazottja.

- Az osztály tömböt használ a halmaz elemeinek tárolására. (nem használhat dinamikus tömböt)
- A konstruktor beállítja a halmaz kapacitását (maximális elemszám)

```
template <class T>
class Halmaz : public AbsztraktHalmaz<T>{
    ...
};
```