

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

- **Kullanılacak kaynaklar:**
<http://www.cplusplus.com/doc/tutorial/>
Published by Juan Soulié
- **C++ ile ileri programlama**
Paul Deitel
Harvey Deitel

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

Copy constructor:

bir nesne parametre değerlerini değerini başka bir nesneden alarak başlayabilir. Böylesi durumlarda oluşturulan nesne için çağrılan constructor a COPY CONSTRUCTOR denilir. Copy constructor yapısı:

```
MyClass::MyClass (const MyClass&);
```

Kopya ataman yolları:

```
MyClass foo;
```

```
MyClass bar (foo); // object initialization: copy constructor called
```

```
MyClass baz = foo; // object initialization: copy constructor called
```

```
foo = bar; // object already initialized: copy assignment called
```

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

Örnek:

```
#include <iostream>

#include <string>

using namespace std;

class Example5 {
    string* ptr;

public:
    Example5 (const string& str) : ptr(new string(str)) {}

    ~Example5 () {delete ptr;}

    // copy constructor:
    Example5 (const Example5& x) : ptr(new string(x.content())) {}

    const string& content() const {return *ptr;}

};

int main () {

    Example5 foo ("Example");

    Example5 bar = foo;

    cout << "bar's content: " << bar.content() << '\n';

    return 0;

}
```

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

Move constructor and assignment:

Copy constructora benzer şekilde move constructorlarda bir nesneye parametre değerlerini başkibir nesneden alabilir.Copyden farkı parametre değerlerini bir nesneden diğerine taşımasıdır. Kaynak nesne parametre değerlerini kaybeder.

```
MyClass fn();      // function returning a MyClass object
```

```
MyClass foo;      // default constructor
```

```
MyClass bar = foo; // copy constructor
```

```
MyClass baz = fn(); // move constructor
```

```
foo = bar;        // copy assignment
```

```
baz = MyClass(); // move assignment
```

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

Friendship and inheritance

İlkesel olarak, bir sınıfın özel ve korunan üyelerine, beyan edildiği aynı sınıftan dışından erişemez. Ancak, bu kural "arkadaşlar" için geçerli değildir.

Arkadaşlar, friend anahtar sözcüğüyle bildirilen fonksiyon veya sınıflardır. Üye olmayan bir fonksiyon, o sınıfın arkadaşı ilan edildiğinde, sınıfın özel ve korumalı üyelerine erişebilir. Bu, bu harici fonksiyonun sınıfta bir beyanını ekleyerek ve bunu friend anahtar kelimesiyle yapılır:

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// friend functions
#include <iostream>
using namespace std;
class Rectangle {
    int width, height;
public:
    Rectangle() {}
    Rectangle (int x, int y) : width(x), height(y) {}
    int area() {return width * height;}
    friend Rectangle duplicate (const Rectangle&);
};
Rectangle duplicate (const Rectangle& param)
{
    Rectangle res;
    res.width = param.width*2;
    res.height = param.height*2;
    return res;
}
int main () {
    Rectangle foo;
    Rectangle bar (2,3);
    foo = duplicate (bar);
    cout << foo.area() << '\n';
    return 0;
}
```

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// friend class
#include <iostream>
using namespace std;
class Square;
class Rectangle {
    int width, height;
public:
    int area ()
        {return (width * height);}
    void convert (Square a);
};
class Square {
    friend class Rectangle;
private:
    int side;
public:
    Square (int a) : side(a) {}
};
void Rectangle::convert (Square a) {
    width = a.side;
    height = a.side;
}

int main () {
    Rectangle rect;
    Square sqr (4);
    rect.convert(sqr);
    cout << rect.area();
    return 0;
}
```

Bilgisayar Programlama II

Ders 5

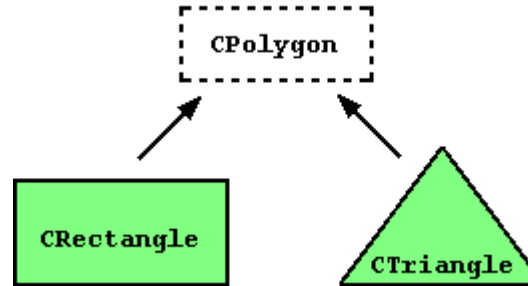
MSGSU Fizik Bölümü

Ferhat ÖZOK

inheritance between classes

Temel sınıfın özelliklerini koruyan yeni sınıflar yaratarak C ++'daki sınıflar genişletilebilir. (inheritance)Kalıtım adı verilen bu süreç bir temel sınıfı ve türetilmiş bir sınıfı içerir: Türetilmiş sınıf, temel sınıfın üyelerini devralır; üst sınıfın üyeleri kendi üyelerini ekleyebilir.

Örneğin, iki çeşit çokgeni tanımlamak için bir dizi sınıf düşünelim: dikdörtgenler ve üçgenler. Bu iki çokgen, alanlarını hesaplamak için gereken değerler gibi bazı ortak özelliklere sahiptir: her ikisi de bir yükseklik ve genişlik (veya taban) ile basitçe tanımlanabilir.



Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

Çokgen sınıfı, her iki tür çokgen için ortak olan üyeler içerir. Bizim durumumuzda: genişlik ve yükseklik. Ve Dikdörtgen ve Üçgen, çokgenlerden birine farklı özelliklere sahip türetilmiş sınıfları olacaktır.

Başkalarından türetilen sınıflar, temel sınıfın erişilebilir üyelerinin tümünü devralır. Bunun anlamı, bir taban sınıfı bir A üyesi içeriyorsa ve B denilen başka bir üyeye birlikte bir sınıf türettiyse, türetilmiş sınıf hem A üyesi hem de B üyesini içerecektir.

İki sınıfın devralma ilişkisi türetilmiş sınıfta bildirilmiştir. Türetilmiş sınıf tanımları aşağıdaki sözdizimini kullanır:

```
class derived_class_name: public base_class_name  
{ /*...*/ }
```

Bilgisayar Programlama II

Ders 5

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// derived classes
#include <iostream>
using namespace std;
class Polygon {
protected:
    int width, height;
public:
    void set_values (int a, int b)
        { width=a; height=b;}
};
class Rectangle: public Polygon {
public:
    int area ()
        { return width * height; }
};
class Triangle: public Polygon {
public:
    int area ()
        { return width * height / 2; }
};

int main () {
    Rectangle rect;
    Triangle trgl;
    rect.set_values (4,5);
    trgl.set_values (4,5);
    cout << rect.area() << '\n';
    cout << trgl.area() << '\n';
    return 0;
}
```