Cihan Hoca 1
30 December 2016, Son Labdan Once (Polymorphism)
GUI Graphical Programming Interface
API Application Programming Interface

*Son kullanıcı arayüzü sunman lazım koddan anlamayan insanlara. (GUI)
GUI, hangisinin nasıl ne ile çalıştığını bilmek zorunda değil. (POLYMORPHISM)
LOJISTIK MÜDÜR, ABSTRACT CLASS.

YAPIYI EN BAŞTAN KURMAK ZORUNDA KALABİLİRSİN.
Zaman, Maliyet, Optimizasyon PROBLEMİ ÖNEMLİ !

POLYMORPHIC YAPI'DA ABSTRACT CLASS'DAN TÜRET DİĞERLERİNİ.

HASH MAP çok önemli.

1. sütun key, 2. sütun value.

```
_____________________
|           |         |
|           |         |
|           |         |
```

Key giriyorsun, değeri veriyor.

VTABLE

```
key         value
_____________________
offset          function addres
|           |         |
|           |         |
|           |         |
```

Class'ın başlangıç adresi (TEXT dosyasında) bir yerlerde tutulur.
Ve onun içindeki fonksiyonu bulmak için, belli bir şeyi başlangıç adresine ekleyip, fonksiyonun yerini bulur.
OFFSET : ADRES BULMAK İÇİN KAYDIRMA.

COMPILER'in compile etmeden önce yaptığı EN ÖNEMLİ İŞ, OPTIMIZATION (PERFORMANS).
ASSEMBLY KODU genellikle sen yazma COMPILER SENDEN DAHA İYİ BİR ŞEKİLDE BU İŞİ YAPIYOR.

COMPILER, neden CONSTRUCTOR ya da DESTRUCTOR ekliyor??
VTABLE için yapıyor.
İlk satırında, eğer bu class için daha önce bir VTABLE yaratılmadıysa, bir tane VTABLE yaratıyor.
	$vtable = new ...

Person ( )
{
	$vtable = new...
}

~Person ( )
{
	delete $ vtable
}

xx		PURE VIRTUAL FUNCTION VARSA, türetilmiş sınıflarda da o fonksiyon olmak zorunda.

HANGİ DURUMLARDA PURE VIRTUAL TANIMLAMA YAPILIR?

class Baby : public Person
{
run dışında bütün virtual fonksiyonları implemente etmiş olsun.
Bebek için run metodu bir şey ifade etmiyorsa, PURE VIRTUAL OLARAK TANIMLANMAZ.
BABY classı o metodu implemente emtek zorunda değildir.
}

Chain of Responsibility
Genelde işler, tek bir sınıfın altına yığılmaz.
Soyut Sınıftan (Abstract Class), başka soyut sınıflar yarat ki, küçük parçalarla problemi çöz.

Bazen, soyut sınıft

Person* p = new Student ();
Student* st = (student)p;
st->enter exam

UNSAFE CAST
Person* p = ...		**new Student (),		new Baby(),		new Worker()**

Student* st = (student *) p

Student* st = dynamic_cast <student *> (p); //Parantezlerle cast işleminden farklı olarak, güvenliyse yapıyor.
Eğer cast etmek istediğin obje (p), student* türünden bir şeyi işaret etmiyorsa, güvensiz oluyorsa ve STUDENT **NULL** yapıyor.

```
if ( st )
    st -> EnterExam ()
```

Person * p = new Student ( ) **DownCasting**
Student tipindeki nesnenin Person tipindeki pointer nesnesine cast ediyoruz.

Student* st = dynamic_cast <Student* > (p) **UpCasting**
Person tipindeki nesneyi Student tipindeki pointer nesnesine cast ediyoruz.

AGGREGIATION
ÖNEMLİ OLAN KİM YOK EDECEK !
Composition, nesnenin yaratılmasından kim sorumluysa silinmesinde de o sorumlu.