



Fenerbahçe Üniversitesi

BLM 102 – Algoritmalar ve Programlama II

LAB 6: Çok Biçimlilik (Polymorphism)

Önemli Not: Aşamaları tamamladıkça, dersin hocası veya asistanı yanınıza çağırarak, tamamladığınız aşamayı gösterdikten sonra diğer aşamaya geçiniz.

LAB Hakkında:

C++ dilinde sınıflarda çok biçimlilik kullanımı ile ilgili deneyler yapılacaktır.

LAB'ın aşamaları ve puanlar:

1- Ön aşamalar (0 Puan)

Karbon salınımı yapan fabrika, ev ve araba'ların toplam karbon salınımı miktarını hesaplayan bir yazılım geliştirilecektir. Bu yazılımda 4 adet sınıf bulunacaktır.

Ana sınıf olan **karbonSalinimi** sınıfında,

Protected olarak

```
* int karbonSalinimiM3;
```

Public olarak

```
* karbonSalinimi(int)
```

```
*virtual int karbonSalinimiAl()
```

Değişken ve fonksiyonları bulunmaktadır.

karbonSalinimi constructor'u aldığı parametreyi karbonSalinimiM3 değerine yazmaktadır. karbonSalinimiAl fonksiyonu ise pure virtual bir fonksiyondur.

Diğer **fabrika**, **ev** ve **araba** sınıfları karbonSalinimi sınıfından türetilmiştir.

Her sınıfın bir constructor'u vardır ve bu constructor aldığı tek integer argümanı, kalıtım yaptığı karbonSalinimi sınıfının constructoru yardımıyla karbonSalinimiM3 değerine yazmaktadır.

Her sınıfta bulunan karbonSalinimiAl fonksiyonu ile geriye karbonSalinimiM3 değeri geri döndürülecektir. Aynı zamanda ekrana fabrika için "Fabrika salinimi geri döndürülüyor." yazılmalıdır. Diğer sınıflar içinde aynı mantıkta yazılar yazılmalıdır.

Bu sınıfları test edecek main kodunda, karbonSalinimi sınıfından 10 adet pointer'lar oluşturup, bu pointer'lara

6 fabrika

3 ev

1 araba

sınıflarından objeler atayınız.

Bu pointer'ları bir vektör'e yazınız, sonrasında vektörden bir döngüde karbonSalinimiAl fonksiyonu ile okuyarak toplam karbon salınımını ekrana gösteriniz.

2- Aşama 1 (50 Puan)

Havyan, kedi ve köpek isminde sınıflar oluşturulacaktır.

Tüm sınıflarda sesCikart fonksiyonu olacaktır.

Base sınıf olan hayvan sınıfındaki sesCikart fonksiyonu pure virtual olarak tanımlanmalıdır.

Kedi ve köpek sınıfı hayvan sınıfından kalıtım yapılacaktır.

Kedi sınıfındaki sesCikart fonksiyonu ekrana "Miyav!" yazdırmalı.

Köpek sınıfındaki sesCikart fonksiyonu ise ekrana "Hav!" yazdırmalıdır.

Main'de 5 elemanı olan hayvan sınıfı pointer array'i oluşturunuz. 5 elemanın ilk 3'üne oluşturulan kedi sınıfından 3 objenin adresi atanmalı, diğer 2 elemana ise köpek sınıfından oluşturulan elemanların adresi atanmalıdır.

Bir döngü ile, tüm elemanların sesCikart fonksiyonları sırası ile çağrılarak ekranda 3 kez "Miyav!" ve ardından 2 kez "Hav!" yazıldığı görülmelidir.

3- Aşama 2 (50 Puan)

karmasikSayilar ve hesapMakinasi isminde iki sınıf oluşturunuz.

karmasikSayilar sınıfında

Public olarak

* int imajiner

* int reel

değişkenleri bulunmaktadır.

Public olarak

- * karmasikSayilar(int, int)
- * karmasikSayi karmasikSayiTopla(int)
- * karmasikSayi karmasikSayiTopla(karmasikSayi)

fonksiyonları bulunmaktadır.

Sınıf'ın constructor'u aldığı iki integer argümanı, real ve imajiner'in üzerine yazmaktadır. Diğer fonksiyonlar ise, karmasikSayi sınıfından yaratılmış objelerin karmasikSayiTopla fonksiyonuna verilen argümanın türlerine göre çağrılacak şekilde tasarlanmıştır.

Yani argüman int ise;

- * karmasikSayi karmasikSayiTopla(int)

fonksiyonu çağrılacaktır.

Argümanı karmasikSayi ise, sınıfın içindeki real ve imajiner ile, argüman olarak alınan karmasikSayinin real ve imajineri toplanarak yeni bir karmasikSayi objesi geri döndürülecektir.

hesapMakinasi sınıfı;

Public olarak

- * int toplamalslemiYap(int, int)
- * karmasikSayi toplamalslemiYap(karmasikSayi, int)
- * karmasikSayi toplamalslemiYap(int, karmasikSayi)
- * karmasikSayi toplamalslemiYap(karmasikSayi, karmasikSayi)

Fonksiyonları bulunmaktadır.

İki integer argüman aldığında, sayılar toplayıp geri döndürmektedir.

Argümanlardan en az biri karmasikSayi olduğunda, o argümanın karmasikSayiTopla fonksiyonunu çağırarak, fonksiyondan dönen karmasikSayi objesini geri döndürmektedir.

Main'de hesapMakinasi sınıfından obje türetip, bu objenin toplamalslemiYap fonksiyonuna;

int, int
karmasikSayi, int
int, karmasikSayi
karmasikSayi, karmasikSayi

objeler vererek (Karmaşık sayılar için karmasikSayilar sınıfından objeler türetiniz) , dönen sonucu ekrana gösteriniz.

Aşama 0 Çözüm:

```
#include <iostream>
#include <vector>
using namespace std;

class karbonSalinimi {
protected:
    int karbonSalinimiM3;

public:
    karbonSalinimi(int arg) {
        karbonSalinimiM3 = arg;
    }

    virtual int karbonSalinimiAl() = 0;
};

class ev : public karbonSalinimi {
public:
    ev(int a) : karbonSalinimi(a) {

    }

    int karbonSalinimiAl() {
        cout << "Ev salinimi geri donduruluyor: " << karbonSalinimiM3 << "\n";
        return karbonSalinimiM3;
    }
};

class fabrika : public karbonSalinimi {
public:
    fabrika(int a) : karbonSalinimi(a) {

    }

    int karbonSalinimiAl() {
        cout << "Fabrika salinimi geri donduruluyor: " << karbonSalinimiM3 << "\n";
        return karbonSalinimiM3;
    }
};

class araba : public karbonSalinimi {
public:
    araba(int a) : karbonSalinimi(a) {

    }

    int karbonSalinimiAl() {
        cout << "Araba salinimi geri donduruluyor: " << karbonSalinimiM3 << "\n";
        return karbonSalinimiM3;
    }
};

int main() {

    vector<karbonSalinimi*> ptr;

    ptr.push_back(new fabrika(10));
    ptr.push_back(new fabrika(10));
    ptr.push_back(new fabrika(13));
    ptr.push_back(new fabrika(12));
    ptr.push_back(new fabrika(11));
```

```
ptr.push_back(new fabrika(10));

ptr.push_back(new ev(5));
ptr.push_back(new ev(4));
ptr.push_back(new ev(3));

ptr.push_back(new araba(1));

//karbonSalinimi* obj1[10] = { new fabrika(10) , new fabrika(5), new fabrika(11), new
fabrika(12), new fabrika(13), new fabrika(14),
//new ev(2) , new ev(1), new ev(1), new araba(4) };

int total = 0;

for (int i = 0; i < ptr.size(); i++) {
    total += ptr[i]->karbonSalinimiAl();
}

cout << "Toplam: " << total;

return 0;
}
```