

Ödev İçeriği

Yazacağınız Eclipse projesi Java dilinde ve konsol uygulaması olmalıdır. Program açıldığında src klasörü içerisinde bulunan Program.cpp dosyasını okuyacaktır. Dosya yolu olarak fiziksel adres verilmeyip direk projenin içerisinden okuyabilmelidir. Okunan Program.cpp dosyası aşağıdaki gibi parçalanmalıdır.

- Sınıflar listelenecektir.
- Her sınıfın içerisinde bulunan ve sadece public olan metotlar listelenecektir.
- Her public fonksiyonun kaç parametre aldığı ve türlerinin ne olduğu ve ne türden bir şey döndürdüğü yazılmalıdır.
- Kalıtım alınan sınıflar ve kaç kez alındıkları yazılmalıdır.

Örneğin Program.cpp dosyasının içeriğinin aşağıdaki gibi olduğu düşünülürse, çıktı aşağıda verilen gibi olacaktır.

```
// Program.cpp içeriği
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <ctime>
using namespace std;

enum MALZEME{
    Ahsap, Alüminyum, Fiberglas, Plastik
};
class Kisi{
private:
    string isim;
    int dogumYili;
    double kilo;
public:
    Kisi(string isim,int dogumYili,double kilo){
        this->isim = isim;
        this->dogumYili = dogumYili;
        this->kilo = kilo;
    }
    Kisi(){
        this->isim = " ";
        this->dogumYili = 1880;
        this->kilo = 55;
    }
}
```

```

        string getIsim()const{
            return isim;
        }
        int getDogumYili()const{
            return dogumYili;
        }
        double getKilo()const{
            return kilo;
        }
        void YemekYe(double kalori){
            this->kilo += (kalori/1000);
        }
        void Kos(double metre){
            this->kilo -= (metre/1000);
        }
    };

```

```

class Arac{
protected:
    Kisi *surucu;
    double benzinLitre;

public:
    Arac(string isim,int dogumYili,double kilo,double benzinLitre){
        this->surucu = new Kisi(isim,dogumYili,kilo);
        this->benzinLitre = benzinLitre;
    }
    Arac(){
        this->surucu = new Kisi();
        this->benzinLitre = 60;
    }
    void Sur(double km){
        benzinLitre -= ((km/100)*7);
        surucu->Kos(km/100);
    }
    ~Arac(){
        delete surucu;
    }
};

```

```

class Tekne : public Arac{
private:
    MALZEME malzeme;

public:
    Tekne(Kisi* kaptan,double depo,MALZEME malzeme):Arac(kaptan->getIsim(),kaptan->getDogumYili(),kaptan->getKilo(), depo)
    {
        this->malzeme = malzeme;
    }
    Tekne():Arac(){
        surucu = new Kisi();
        benzinLitre = 80;
        malzeme = Ahsap;
    }
    string MalzemeToString(){
        switch(malzeme){

```

```

        case Ahsap:
            return "Ahsap";
        case Aluminyum:
            return "Aluminyum";
        case Fiberglas:
            return "Fiberglas";
        case Plastik:
            return "Plastik";
        default:
            return "";
    }
}

friend ostream& operator<<(ostream& ekran, Tekne& sag){
    ekran<<"Malzeme: "<<sag.MalzemeToString()<<endl;
    ekran<<"Depo: "<<sag.benzinLitre<<endl;
    ekran<<"Kaptan: "<<sag.surucu->getIsim()<<endl;
    return ekran;
}

};

int main(){
    Tekne **tekneler = new Tekne*[3];
    tekneler[0] = new Tekne();
    Kisi *ahmetKaptan = new Kisi("Ahmet",1974,85);
    Kisi *mehmetKaptan = new Kisi("Mehmet",1965,81);
    tekneler[1] = new Tekne(ahmetKaptan,75,Fiberglas);
    tekneler[2] = new Tekne(mehmetKaptan,80,Aluminyum);
    cout<<*tekneler[1]<<endl;

    for(int i=0;i<3;i++) delete tekneler[i];
    delete [] tekneler;
    return 0;
}

```

Program Çıktısı:

Sınıf: Kisi

Kisi

Parametre: 3 (string, int, double)

Dönüş Türü: Nesne Adresi

Kisi

Parametre: 0

Dönüş Türü: Nesne Adresi

getIsim

Parametre: 0

Dönüş Türü: string

getDogumYili

Parametre: 0

Dönüş Türü: int

getKilo

Parametre: 0
Dönüş Türü: double

YemekYe

Parametre: 1 (double)
Dönüş Türü: void

Kos

Parametre: 1 (double)
Dönüş Türü: void

Sınıf: Arac

Arac

Parametre: 4 (string, int, double, double)
Dönüş Türü: Nesne Adresi

Arac

Parametre: 0
Dönüş Türü: Nesne Adresi

Sur

Parametre: 1 (double)
Dönüş Türü: void

~Arac

Parametre: 0
Dönüş Türü: void

Sınıf: Tekne

Tekne

Parametre: 3 (Kisi*, double, MALZEME)
Dönüş Türü: Nesne Adresi

Tekne

Parametre: 0
Dönüş Türü: Nesne Adresi

MalzemeToString

Parametre: 0
Dönüş Türü: string

operator<<

Parametre: 2 (ostream&, Tekne&)
Dönüş Türü: ostream&

Super Sınıflar:

Arac: 1

Yukarıdaki örnekte Arac kelimesinin yanında yazan 1, sınıftan kaç kez kalıtım alındığını göstermektedir. C++ dilinde ifadeler arasındaki boşluk sayısı az veya çok olması bir şeyi değiştirmedigi için dosyayı okurken buna dikkat edilmelidir.

Yazılacak proje mutlaka Eclipse IDE 2021-03 ortamında açılabilen ve derlenebilen proje olmalıdır.

JRE versiyonu mutlaka JavaSE 15 olmalıdır.