

BM102 - ALGORİTMA VE PROGRAMLAMA – 2

ÖDEV -2

AÇIKLAMALAR;

- Ödevler Bireysel olarak yapılacaktır.
- https://turnitin.com/login_page.asp?lang=tr sitemine yüklenecektir.
- Turnitin kullanımı <http://akademik.duzce.edu.tr/arafatsenturk/Profil/Dokumanlar> adresinde mevcuttur.
- Sınıf Numarası: **25888105** Sınıf Şifresi:**2468**'dir.
- Ödevler her hangi bir derleyici (DevC gibi) kullanılarak yapılacaktır
- PDF dosyasının ismi şu şekilde olacaktır: ÖğrenciNO.pdf
Bu standartta gönderilmeyen ödevler değerlendirilmeye alınmayacaktır.
- Turnitin'a yükleme yaparken BAŞLIK bölümü de öğrenci No giriniz.
- Turnitin sistemindeki Ödev 2'in son yüklenme tarihi **28 Ağustos 2020 saat: 14:00'dir.** Bu tarih ve saatten sonra sistem otomatik olarak kapılacaktır.
- Turnitin sisteminin dışında e-mail veya elden teslim edilen ödevler kesinlikle kabul **EDİLMEYECEKTİR.**

ÖDEV -2

- 1 Aşağıda Termometre sınıfı ve bu sınıfa ait yapıcı fonksiyon tanımları verilmiştir. Bu sınıf tanımını dikkate alarak aşağıdaki komutların geçerli ya da geçersiz olup olmadıklarını belirtiniz.

```
class Termometre
{
    private:
        float deger;
        string birim;
    public:
        Termometre(){}
        Termometre(float _deger);
        Termometre(float _deger, string _birim)
            :deger(_deger),birim(_birim){}
        float dereceGoster()
        {
            cout<<deger<<" "<<birim<<endl;
        }
};
Termometre::Termometre(float _deger)
{
    deger=_deger;
}
```

- a. Termometre t;
- b. Termometre t(24.5);
- c. Termometre t(33.4, "santigrat");
- d. Termometre t("santigrat");
- e. Termometre t = Termometre (25.6);
- f. Termometre t(22);
- g. Termometre t();

2 DERSTE ÇÖZÜMÜ YAPILDI

Ulke adında bir sınıf tanımı yapılmıştır. Bu sınıftan aşağıda verilen nesnelerin yaratılması istenmektedir. Buna göre Ulke sınıfını ve bu sınıfa ait gerekli yapıcı fonksiyon tanımlarını yapınız.

- a. Ulke turkiye("Ankara", "TR"); //başkent ve uluslararası plaka
- b. Ulke almanya;
- c. Ulke ingiltere("Londra");
- d. Ulke kosova(2000000); //nüfusu
- e. Ulke ıspanya("Madrid", "E", "İspanyolca");
- f. Ulke estonya(372); //telefon kodu

3 Aşağıda verilen programın çıktısını yazınız:

```
#include <iostream>
using namespace std;
class A
{
    public:
        int x;
    public:
        A(int _x);
        ~A();
};
A::A(int _x)
{
    x = _x;
    cout << "A(int) " << x << endl;
}
A::~~A()
{
    cout << "~A() " << x << endl;
}
class B
{
    private:
        A a;
    public:
        B(A pa): a(pa)
        {
            cout << "B(A) " << endl;
        }
        ~B()
        {
            cout << "~B() " << endl;
        }
};
B::B(A pa): a(pa)
{
    cout << "B(A) " << endl;
}
B::~~B()
{
    cout << "~B() " << endl;
}

int main()
{
    A a(7);
    B b(a);
    cout << a.x;
    return 0;
}
```

- 4 Aşağıda verilen programın çıktısını yazınız:

```
#include <iostream>
using namespace std;
int g=5;
class Test
{
    int x,y;
public:
    Test(int _x,int _y):x(_x),y(_y){g += 3;}
    Test(int _x);
    ~Test();
    int f(int&, int=8);
};
Test::Test(int _x)
{
    x=_x;
    y=_x;
}
Test::~~Test()
{
    cout<<x<<" "<<y<<" "<<g<<endl;
}
int Test::f(int& s, int t)
{
    s*=2;
    t*=3;
    x+=2;
    y+=3;
    cout<<s<<" "<<t<<endl;
    return s+t;
}

int main(){
    int d=1, e=2;
    Test obj1(3), obj2(2,4);
    g=obj1.f(d,e);
    g=obj2.f(e);
    cout<<d<<e<<endl;
    return 0;
}
```

- 5 Aşağıdaki programda bir oyun oynayan üç oyuncunun bilgileri girilmiş ve en yüksek puan alan oyuncu puanıyla ekrana yazılmıştır. Verilen örnek çıktıyı dikkate alarak şekilde aşağıdaki programda eksik bırakılan yerleri tamamlayınız.

```
#include <iostream>
using namespace std;
class Oyuncu
{
    private:
        string rumuz;
        int puan;
        _____:
        Oyuncu(string _rumuz, int _puan):
            _____ (_____),
            _____ (_____){}
        Oyuncu(string _rumuz);
            _____ (){}
            _____ puanAl();
        void durumYazdir();
};
Oyuncu::Oyuncu(string _rumuz)
{
    _____ = _____
}
int Oyuncu_____puanAl()
{
    return _____;
}
void _____::durumYazdir()
{
    cout<<_____<<endl;
    cout<<_____<<endl;
}
int main()
{
    Oyuncu oyuncular_____
    _____oyuncu1("Hugo",300);
    _____oyuncu2("Mario",460);
    _____oyuncu3("Keloglan",500);
    oyuncular[0] = oyuncu1;
    oyuncular[1] = oyuncu2;
    oyuncular[2] = oyuncu3;
    int max = oyuncular[0]. _____ ();
    for (int i=0; i<3; i++)
    {
        if(oyuncular[i].puanAl()>max){
            max = _____ .puanAl();
        }
    }
}
```

```

for (int i=0; i<3; i++)
{
    if(oyuncular[i].puanAl() == max){
        cout<<"Kazanan..."<<endl;
        _____durumYazdir();
        break;
    }
}
return 0;
}

```

Örnek Çıktı

```

Kazanan...
Keloglan
500

```

6 DERSTE ÇÖZÜMÜ YAPILDI.

Aşağıda isim ve telefon numaralarını içeren TelefonRehberi tipinde rehber1 nesnesi oluşturulmuştur. Bu rehberdeki bilgileri de içerecek şekilde yeni bir nesne oluşturmak istenmektedir. Kopya yapıcı fonksiyonları kullanarak rehber1 nesnesinin içeriğini kopyalayarak rehber2 nesnesini yaratan ve yeni nesnenin içeriğini aşağıda verilen örnekteki gibi ekrana yazdıran bir program yazınız.

```

string isimler[] = {"Aydan Sener","Fatma Girik",
                  "Filiz Akin","Hulya Kocyigit",
                  "Turkan Soray"};

long telefonlar[] =
    {1111111,2222222,3333333,4444444,5555555};

TelefonRehberi rehber1(isimler,telefonlar);
TelefonRehberi rehber2(rehber1);
rehber2.yazdir();

```

Örnek Çıktı

```

Aydan Sener 1111111
Fatma Girik 2222222
Filiz Akin 3333333
Hulya Kocyigit 4444444
Turkan Soray 5555555

```

7 Bir polinomu aşağıdaki formülle tanımlayabiliriz:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0,$$

a_0, \dots, a_n : katsayılar

n : polinomun derecesi (polinomdaki en büyük üs)

Polinom sınıfının aşağıdaki gibi tanımlandığını düşünelim:

```
class Polinom
{
    int *katsayi;           // Katsayilari tutan dinamik dizi
    int derece;             // En büyük üs
public:
    Polinom (int);          // Dereceye göre dinamik katsayı dizisini yaratır
    void katsayiAta();       // Katsayilari diziye okur
    void yaz();              // Polinom katsayılarını yazar
    ...
};
```

Polinom sınıfına `topla()` isimli bir üye fonksiyon yazalım. Bu fonksiyon parametre olarak aldığı polinomu içinde bulunduğu polinomla toplar ve sonucu içinde bulunduğu polinomun içine atar. Toplanacak polinomların derecelerinin aynı olduğunu varsayabilirsiniz. `main()` fonksiyonu ve örnek çıktı aşağıda verilmiştir. Not: Polinomlarda toplama işlemi aynı dereceden terimlerin toplanması ile yapılır.

Örnek: $(2x^3+4x+2) + (3x^3+2x^2+2x+1) = (5x^3+2x^2+6x+3)$

Örnek Çıktı

```
Polinomun derecesini giriniz:3
Katsayilari giriniz:
Katsayı 3: 2
Katsayı 2: 0
Katsayı 1: 4
Katsayı 0: 2
Katsayilari giriniz:
Katsayı 3: 1
Katsayı 2: 2
Katsayı 1: 2
Katsayı 0: 1
(2x3+4x+2) + (3x3+2x2+2x+1) = (5x3+2x2+6x+3)
```

```
int main()
{
    int d;
    cin>>" Polinomun derecesini giriniz:";
    cin>>d;
    Polinom p1(d),p2(d);
    p1.katsayiAta();
    p2.katsayiAta();
    p1.yaz();
    cout<<" + ";
    p2.yaz();
    cout<<" = ";
    p1.topla(p2);
    p1.yaz();
    return 0;
}
```


- 8 Tarih isimli üç veri üyesi (gun,ay ve yıl) içeren bir sınıf yazınız. Sınıfın üye fonksiyonları aşağıda verilmiştir:

- Yapıcı fonksiyon: Gelen 3 parametrenin değerlerini gun, ay ve yıl'a atar. Parametre yollanmadığı durumda tarihi 1.1.2012'den başlatır.
- arttir() : şu andaki tarihi 1 gün arttırır. Ayın sonunda ise bir sonraki aya, yılın sonunda ise bir sonraki yıla geçer. Atık yıl olmadığını varsayabilirsiniz.
- goster () : şu andaki tarihi gun/ay/yıl formatında yazdırır.

Ayların kaç çektiğini tanımlamak için aşağıdaki genel değişkeni kullanabilirsiniz.

```
int aylar[ ] = {31, 28, 31, 30, 31, 30, 31, 31,30, 31, 30, 31};
```

Sınıfınızı denemek için aşağıdaki main() fonksiyonu verilmiştir:

```
int main()
{
    Tarih tarih(30,7);
    cout<<"Bugun: ";
    tarih.goster();
    cout<<"Onumuzdeki 10 gun:";
    for (int i=1; i<=10; ++i){
        tarih.arttir();
        tarih.goster();
    }
    return 0;
}
```

Örnek Çıktı

```
Bugun: 30 / 7 / 2012
Onumuzdeki 10 gun:
31 / 7 / 2012
1 / 8 / 2012
2 / 8 /2012
...
9 / 8 / 2012
```


- 9 Bir sinemada tek bir salon mevcuttur ve bu salonda günde üç seans (12, 15, 18) film oynatılmaktadır. Salonun kapasitesi 20 kişiliktir ve seyirciler istedikleri koltukta oturabilmektedir. Bir seansın bilgilerini tutan bir sınıf tasarlayın. Bu sınıfta 20 kişilik salonun doluluk bilgisi bir tamsayı veri üyesi olarak tutulacaktır. Seans sınıfının üye fonksiyonları aşağıda listelenmiştir.

- Yapıcı fonksiyon: Salon doluluk bilgisine ilk atama yapar.
- yerVar():Salonda n adet koltuk boşsa true döndürür, yoksa false döndürür.
- yerAyir():Yer varsa n koltuk ayırır, yoksa kullanıcıya mesaj yazar.
- iptal():n koltukluk rezervasyonu iptal eder
- kalanKoltuk():kalan koltuk sayısını döndürür.

Sinemanın günlük rezervasyon işlemlerini yapan bir program yazınız. Programınızda her seansı tutacak 3 elemanlı bir nesne dizisi yaratıp örnek çıktıda görüldüğü gibi her seansla ilgili rezervasyon, iptal ve kalan koltuk sayısı gibi işlemleri kullanıcı programı bitirene kadar gerçekleştiriniz.

Örnek Çıktı

```
Lutfen asagidakilerden birini seciniz:
R: Rezervasyon
I : Iptal
K: Kalan koltuk sayisi
S: Son
Secenek: R
Seans Saati (12,15,18): 18
Koltuk sayisi: 5
5 koltuk reserve edildi.
Secenek: K
Seans Saati (12,15,18): 15
20 koltuk reserve edilebilir.
Secenek: S
```

- 10 Verilen program parçasını dikkate alarak aşağıda verilen komutlardan geçerli ve geçersiz olanları belirtiniz.

```
#include <iostream>
using namespace std;
class A
{
    int x;
    const int y;
    public:
        A(int _y):y(_y){}
        int yAl() const;
        int xAl() const;
        void xAta(int x);
};
a. int A::xAl() const
{
    return x++;
}
b. void A::xAta(int _x)
{
    const int x= _x;
}
c. int A::yAl() const
{
    return y;
}
d. void A::yAta(int _y) const
{
    y= _y;
}
```

- 11 Aşağıdaki programın hatalarını bulunuz ve hata nedenlerini açıklayınız

```
#include <iostream>
class Test
{
    private:
        int deg1;
        const int deg2;
        int &deg3;
        void degistir()
        {
            cout<<"merhaba ";
        }

    public:
        Test(int);
        int bir()const
        {
            cout<<"hoscakal ";
            deg1*=2;
        }
        int iki()
        {
            return deg1;
        }
};
Test::Test (int s):deg2(s)
{
    deg1=s;
    deg3=s;
}
int main()
{
    Test t(5);
    const Test z(10);
    t.deg1=8;
    int k=t.iki();
    t.degistir ();
    cout<<z.iki();
    return 0;
}
```

12 Aşağıda verilen programın çıktısını yazınız.

```
#include <iostream>
using namespace std;
class A
{
    static int m;
public:
    A();
    int mAl();
    friend class B;
};
A::A()
{
    m++;
}

int A::mAl()
{
    return m;
}
int A::m=0;
class B
{
    static int n;
public:
    B(A& a){n++;a.m++;}
    int nAl();
};
int B::n=0;
int B::nAl()
{
    return n;
}
int main()
{
    A a1;
    A a2;
    B b1(a1);
    B b2(a2);
    A a3;
    B b3(a3);
    cout<<"A->m:"<<a1.mAl()<<endl;
    cout<<"B->n:"<<b1.nAl()<<endl;
    return 0;
}
```

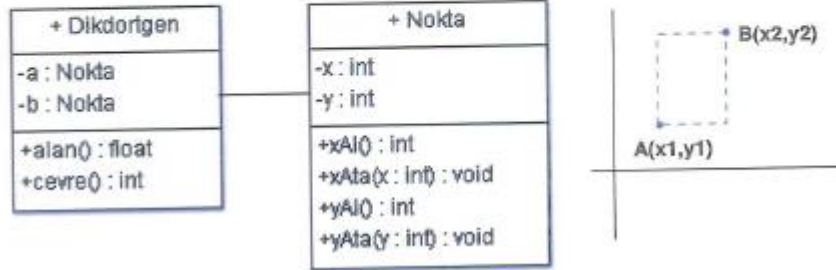
- 13 Aşağıda verilen programın çıktısını yazınız.

```
#include <iostream>
using namespace std;
class X
{
    static int one;
    int number;
public:
    X(int =0);
    X& change(int);
    void print();
};
int X::one=3;

X::X(int a)
{
    one+=3;
    number=a;
}
void X::print()
{
    cout<<one<<setw(5)<<number<<endl;
}
X& X::change(int b)
{
    number*=b;
    return *this;
}
int main()
{
    X obj1(3);
    obj1.print();
    obj1.change(4).change(5).print();
    X obj2;
    obj2.print();
    return 0;
}
```

14 DERSTE ÇÖZÜMÜ YAPILDI.

Aşağıda şekilde gösterilen ve iki nokta ile tanımlanan dikdörtgenin çevresini ve alanını hesaplayan bir program yazınız. Programda aşağıda sınıf diyagramları verilen Dikdortgen ve Nokta sınıflarını yaratınız ve bu sınıfları kullanarak örnek çıktıda verildiği gibi çalışan bir program yazınız.



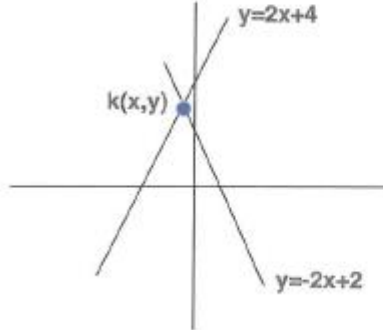
```
int main()
{
    int x,y;
    cout<<"a noktasi (x,y):";
    cin>>x>>y;
    Nokta a(x,y);
    cout<<"b noktasi (x,y):";
    cin>>x>>y;
    Nokta b(x,y);
    Dikdortgen d(a,b);
    cout<<"Cevre:"<<d.cevre()<<endl;
    cout<<"Alan:"<<d.alan()<<endl;
    return 0;
}
```

Örnek Çıktı

```
a noktasi (x,y): 4 5
b noktasi (x,y): 10 15
Cevre:32
Alan:60
```

15

Kesişen iki doğrunun ($y=2x+4$ ve $y=-2x+2$) kesişim noktasını bulan ve ekrana yazdıran program aşağıda verilmiştir. Bu programda Nokta.h ve Dogru.h özel başlık dosyaları kullanılmıştır. Örnek verilen çıktıyı dikkate alarak, programın çalışması için gerekli olan bu özel başlık dosyalarını ve programda eksik bırakılan kısımları yazınız.



Dogru-1 $y=a_1x+b_1$; eğim= a_1

Dogru-2 $y=a_2x+b_2$: eğim= a_2

$$k(x,y) \Rightarrow \left(\frac{b_2-b_1}{a_1-a_2}, a_1x + b_1 \right)$$

```
#include <iostream>
#include "Dogru.h"
#include "Nokta.h"
#include "math.h"
using namespace std;
float Dogru::egim()
{
}
Nokta Dogru::kesisim(Dogru &dogru)
{
}
int main()
{
    Dogru dogru1(2.0,4.0);
    Dogru dogru2(-2.0,2.0);
    cout<<"Dogru-1 egim:"<<dogru1.egim()<<endl;
    cout<<"Dogru-2 egim:"<<dogru2.egim()<<endl;
    Nokta k = dogru1.kesisim(dogru2);
    cout<<"Kesisim (K) noktasi (x,y):"<<k.xAl()<<","<<k.yAl()<<endl;
    return 0;
}
```

Örnek Çıktı

```
Dogru-1 egim:2
Dogru-2 egim:-2
Kesisim (K) noktasi (x,y):-0.5,3
```


16

Girilen boy ve kilo değerlerine göre vücut kütle indeksi değerini hesaplayan ve kilo durumunu aşağıda verilen tabloya göre gösteren bir program yazınız. Bu programda aşağıdaki sınıf diyagramına bağlı kalmanız istenirken, sınıf içinde yazacağınız `vkiHesapla()` ve `kiloDurumu()` fonksiyonlarını test etmek için `testVkiHesapla()` ve `testKiloDurumu()` arkadaş fonksiyonlarını da yazmanız istenmektedir. Programa ait `main()` fonksiyonu ve örnek çıktı aşağıda verilmiştir.

Vücut Kütle İndeks Kategorileri

<15	Açlık
15-18.5	Zayıf
18.5-25	Normal
25-30	Kilolu
>30	Obez

$$\text{Vücut Kütle İndeksi (KVi)} = \text{kilo} / \text{boy}^2$$

$$\text{İdeal Kilo} = \text{boy}^2 \cdot 22$$

```
class VucutKutlesi
{
    float boy;
    float kilo;
    string durum;
public:
    VucutKutlesi(float _boy, float
_kilo):boy(_boy),kilo(_kilo){}
    float vkiHesapla();
    float idealKiloHesapla();
    friend void testVkiHesapla();
    friend void testIdealKiloHesapla();
};
int main()
{
    testVkiHesapla();
    testIdealKiloHesapla();
    return 0;
}
```

Örnek Çıktı

VKI hesabi test ediliyor...

Kilo: 100 kg

Boy: 1.80 m

Test Başarısız...

Hesaplanan: 15.4321

Olması gereken: 30.8642

İdeal kilo hesabi test ediliyor...

Kilo: 100 kg

Boy: 1.80 m

Test başarılı...

İdeal kilo: 71.28

- 17 EnUcuz marketler zincirinin online alışveriş sitesinde kullanılacak bir program yazınız. Bu programda müşterinin aldığı her kalem malın bilgilerini tutmak için Kalem isimli bir sınıf oluşturun. Kalem sınıfı aşağıdaki bileşenlerden oluşacaktır:

- Her kalemin isim, fiyat, adet bilgisi (örneğin müşteri, birim fiyatı 0.9 lira olan 3 adet Selpak mendil (isim) seçebilir).
- Toplam tutarı tutmak için toplam isimli statik veri üyesi.
- Yollanan parametreleri veri üyelerine atayan yapıcı fonksiyon.

Her müşterinin 10 çeşit mal alacağını varsayarak, 10 elemanlık bir diziye alınan kalemleri okuyalım, ve müşterinin ödemesi gereken toplam tutarı ekrana yazdıralım

Örnek Çıktı

```
Kalemin ismini giriniz: SelpakMendil
Kalemin fiyatını giriniz: 0.9
Kalemin adetini giriniz: 3
Kalemin ismini giriniz: PinarSut
Kalemin fiyatını giriniz: 0.7
Kalemin adetini giriniz: 2
...
Toplam tutar: 15.5 lira
```

- 18 EPosta isimli aşağıdaki bileşenlerden oluşan bir sınıf tanımlayın:

Veri üyeleri:

- sayı: epostada yer alan kelime sayısı
- kelimeler: mesajdaki kelimeleri tutan string tipinde dinamik dizi.
- alan: emaili alan insanın soyadı.
- yollayan: emaili yollayan insanın soyadı.
- adet (statik): kelime sayısı 10'den fazla olan email mesajlarının adedi

Üye fonksiyonlar:

- Yapıcı fonksiyon: Mesajdaki kelime sayısını parametre olarak alır ve bu sayıya göre kelimeler dizisini dinamik olarak yaratır.
- oku(): alan ve yollayanın isimlerini ve mesajı okur.

main() fonksiyonu bir dizi içinde 50 EPosta nesnesi yaratıp kaç mesajın 10 kelimeden uzun olduğunu ekrana yazdırır.

Örnek Çıktı

```
Mesajdaki kelime sayisini giriniz: 12
Alanin soyadini giriniz: ozcan
Yollayanin soyadini giriniz: dogru
Mesaji giriniz: merhaba mehmet nasilsin burada isler iyi gidiyor
herkese selam soylemeyi unutma ahmet
...
10 kelimeden uzun mesajlarin sayisi: 14
```

19

Bir havaalanındaki uçuş tarifelerini yöneten bir program yazınız. Program aşağıdaki bileşenlerden oluşacaktır:

- Zaman sınıfı :
 - s,d : saat, dakika
 - oku(): veri üyelerini okur.
 - yaz () : veri üyelerini ekrana yazar.
 - arttır () : dak isimli parametreyi alıp şu andaki saati dak kadar arttırır.
- Ucus sınıfı:
 - ucno: uçuş numarası.
 - kalkis, varis: bir ucagin kalkis ve varis zamanlarını tutan Zaman nesneleri
 - yapıcı fonksiyon: veri üyelerini okur
 - goster () : veri üyelerini ekrana yazdırır.
 - rotar () : rötat zamanını (dakika cinsinden) parametre olarak alıp kalkış ve varış zamanlarını günceller.
- main() :
 - Kullanıcıdan o gün kaç uçuş olacağını okur.
 - Uçuş tarifelerini elemanları Ucuş nesneleri olan dinamik bir dizi olarak yaratır.
 - Rötat bilgisini okur.
 - O günkü tüm uçuşları rötat bilgisine göre günceller.
 - Güncellenmiş uçuş tarifelerini ekrana yazdırır.

Örnek Çıktı

```
Ucuş sayisini giriniz: 2
Ucuş numarası, kalkis ve varis saatlerini giriniz: 111 10 15 12 30
Ucuş numarası, kalkis ve varis saatlerini giriniz: 222 14 10 17 40
Rotar bilgisini dakika degerinde giriniz: 40
Tum ucuşlarda 40 dakikalik rotar olacaktır.
Guncellenmis Ucuş Tarifesi:
Ucuş no: 111 Kalkis: 10:55 Varis: 13:10
Ucuş no: 222 Kalkis: 14:50 Varis: 18:20
```