

# Result Class

## INF212 Class Design Project

Prepared by  
171024093  
Muhammet Fatih KESKİN

INF 212 Algorithms and Programming II  
Spring 2019

Date Submitted: May 14, 2019

### Project objective /Projenin amacı

Projenin amacı, standart devre elemanları ile devre analizi yapmak. Temel olan ohm kanunundan yardım alarak devre elemanların alabileceği değerleri hesaplayan bir program yaptım. Kullanıcının seçimine göre Volt-Direnc-akım hesaplayan devre tasarımı yaptım.

Daha sonra dirençlerin seri veya paralel olup olmadığının kontrolü yapıp Eşdeğer dirence göre Eşdeğer devre çizimi yapan programı oluşturdum. Bu programa ek olarak cout komutu ile ekrana girilen devre değerlerini bir devre şeklinde her değer için ayrı bir devre çizen programı yaptım.

Exception Handling kısmında ise kullanıcın olası bir yanlış tuslamasında ve devre değerleri girilirken ohm yasasına uygun olup olmadığının kontrolünü yapan bir program yazdım.

Program sayesinde karışık devreler aza indirgenebilir ve ohm yasası kullanımı kavranabilir.

### Description of class design/ Sınıf tasarımının tanımlanması

Asıl amacım olan devre analizi olduğu için ilk class olarak Devre ismini kullandım.

Daha sonra Myexception olarak hata yakalama classı oluşturdum ve içinde oluşturduğum fonksiyonu yeni class olan Devreçiz olarak belirledim.

Yani class içinde class ve bu classın fonksiyonlarını kullanan başka bir class yaptım.

## Description of class member and non-member functions / Sınıfın üye fonksiyonların ve üye olmayan fonksiyonların tanımlanması

```
class Devre{

    int r,i,v;

    char *name;

    public:

        Devre();

        Devre(float ,float ,float ,char*);

        Devre(float ,float ,float );

        Devre(const Devre&);

        Devre &operator = (const Devre &);

        ~Devre(){   };

//    cout <<"destructor\n"<<endl;

    void Print();

    void setAll(float,float,float,char*);

    void setR(float);

    void setV(float);

    void setI(float);

    void setName(char*);

    void Hesapla();
```

```
const float getR();
```

```
const float getI();
```

```
const float getV();
```

```
const char* getName();
```

```
Devre operator + (Devre const &obj);
```

```
Devre operator - (Devre const &obj);
```

```
Devre operator * (Devre const &obj);
```

```
Devre operator / (Devre const &obj);
```

```
Devre operator += (Devre const &obj);
```

```
Devre operator -= (Devre const &obj);
```

```
// Devre operator == (Devre const &obj);
```

```
// Devre operator != (Devre const &obj);
```

```
bool operator == (Devre &obj);
```

```
bool operator != (Devre &obj);
```

```
class Myexception{
```

```
    int resistor,current,voltage;
```

```
    public:
```

```
        void Hata();
```

```
};
```

```
class DevreCiz{
```

```
    float A,x9,y9,z9;
```

```
    public:
```

```
        void Hata1();
```

```
void Devre9(float x9,float y9,float z9){
    cout <<"\t\t\t Res="<<x9<<" l="<<y9<<endl;
    cout <<"\t\t\t---vvvvvvv---->----"<<endl;
    cout <<"\t\t\tİ          İ"<<endl;
    cout <<"\t\t\tİ          İ"<<endl;
    cout <<"\t\t\tİ          İ"<<endl;
    cout <<"\t\t\tİ          İ"<<endl;
    cout <<"\t\t\tİ-----| i-----İ"<<endl;
    cout <<"\t\t\t V="<<z9<<endl;}

void Karisik(float s,float p,float i,float rs,float r,float r1,float rp,float rpt){
cout<<"\nSeri Direnc Sayisi Gir: ";
cin>>s;
if(s!=0){
    for(i=0;i<s;i++){
        cout<<"Seri Direnc Degerlerini Sirasiyla Giriniz: ";
        cin>>r;
        rs+=r;
    }
}
else{
    rs=0;
}

cout<<"Seri Direnclerin Toplami: "<<rs<<endl<<endl;
cout<<"Paralel Direnc Sayisi Gir: ";
cin>>p;
if(p!=0){
```

```

    for(i=0;i<p;i++){

        cout<<"Paralel Direnc Degerlerini Sirasiyla Giriniz: ";

        cin>>r1;

        rp+=1/r1;

        rpt=1/rp;

    }

}

else {

    rpt=0;}

cout<<"Paralel Direnclerin Toplami: "<<rpt<<endl<<endl;

cout<<"Karisik Devrenin Toplam Direnci: "<<rs+rpt<<endl<<endl;}

//paralelde bir direnc bile 0 girilirse akim o kablodan gececegi için Reş 0 olur.

};

friend ostream& operator << (ostream &abc, const Devre &obj){

    abc << obj.r << obj.v << obj.i;

    return abc;}

friend ostream& operator >> (ostream &abc, const Devre &obj){

    abc << obj.r << obj.v << obj.i;

    return abc;}

};

```

Devre için: r direnc,i akim,v volt olarak private tanımlı. Public olarak constructor ve destructor set ve get fonksiyonlar ve en son overloading tanımlı.

Myexception için: resistor direnc,current akim, voltage volt olarak private tanımlı. içerisinde sadece tek bir fonksiyon var.

DevreCiz için: İçerisinde bir önceki classta tanımlı olan hata fonksiyonu vardır bu sayede class içindeki fonksiyonu başka bir classta kullanabildiğimi gösterdim. Devre9 fonksiyonu sadece ekrana devre şekli çizdirmek için kullandığım bir fonksiyondur. Karışık devre fonksiyonu sayesinde karışık devre cozumu yaptım. İşlemleri fonksiyonda yapıp main fonksiyonunda sadece fonksiyonu çağırdım.

## Description of class usage/ Sınıfın kullanımın tanımlanması

Sınıfın içerisinde tanımladığım constructor ve fonksiyonları programın ilerleyen kısımlarında . veya :: ile çağırıp kolayca kullandım. Devre classını ana class olarak diğer classları alt başlık classı olarak kullandım.

Bu sayede program yazarken .(dat) ile çıkan kısımda rahatca buldum. Buda amacına uygun kullanıldığını göstermekteydi.

## User's guide / Kullanıcı Rehberi

Programda constructor yapıları,cout,cin veya input-output fonksiyonları, overloaded operatorler, bool operatorler ve exception handling fonksiyonlarının hepsinin kullanımına yer verilmiştir.

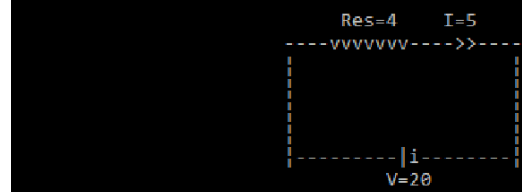
Devre ilk olarak size constructor degerlerine göre çizilen devreyi sunmaktadır. Daha sonra overloading operator tanımlamaları ile eşdeğer devre çizdirmiştir. (Genelde voltajlarda oynama yapıldı).

Daha sonra istenilen degeri girilen devrenin akımını voltunu veya eşdeğer direncini veya karışık devre çözümü yapabilen bir program gelmektedir. Y yapmak istediğiniz işlemi seçip kullanmak istediğiniz degerleri giriyorsunuz. Ulaşmak istediğiniz deger ekrana yazdırılacaktır.

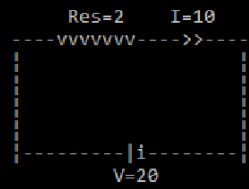
Son olarak exception handling sayesinde yapabileceğiniz hataları size düzeltmeniz için gösteren hata yakalayan bir programdır. Eger bütün işlemleri Dogru tamamlırsanız hata algılanmadı uyarısı yazar ve programı sonlandırır.

## Results of the solution / Programın Sonuçları

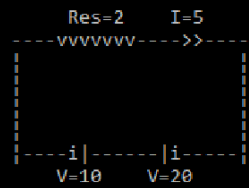
--	--



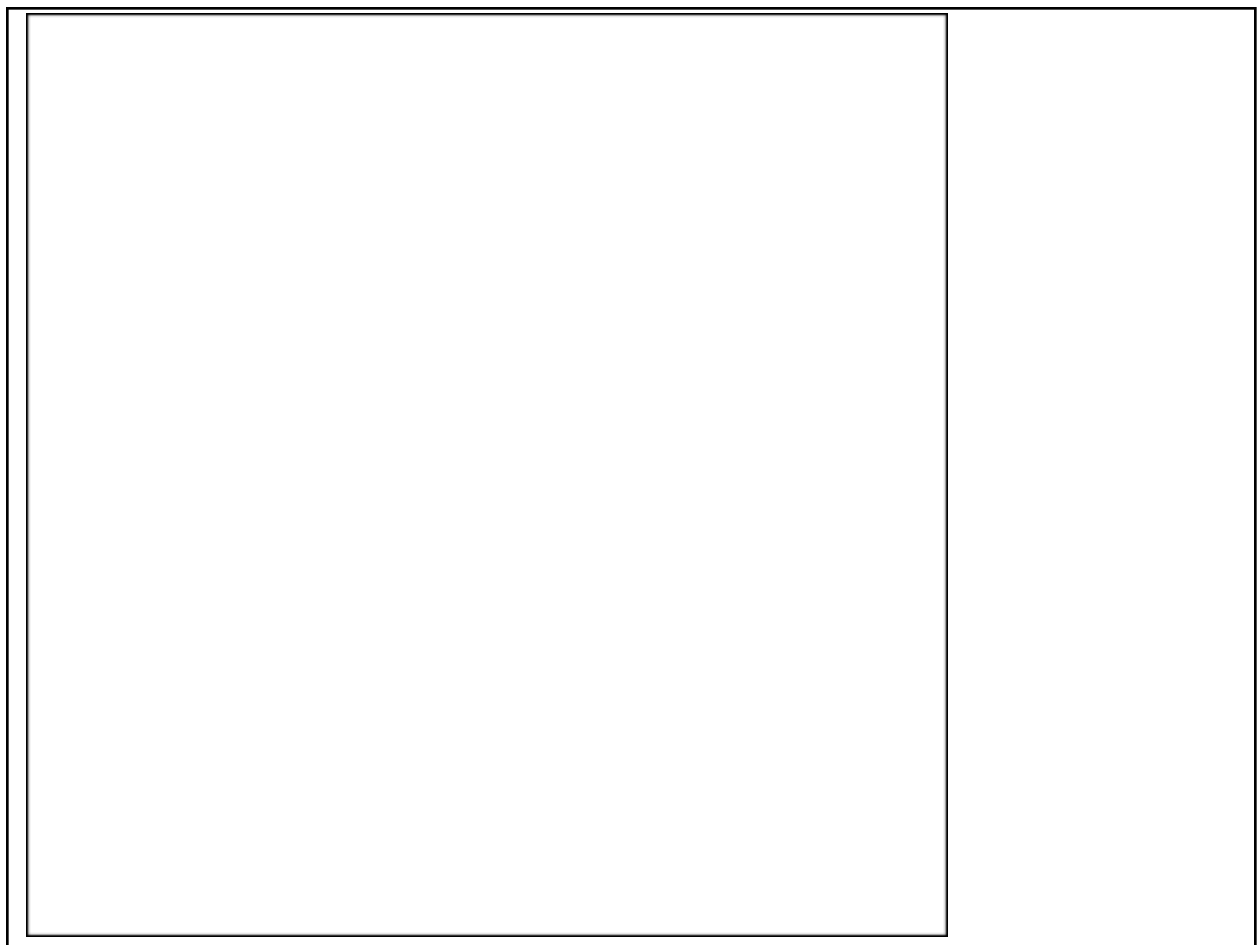
Devre Elemanlarının Değerleri : Direnc = 2      Voltaj = 20      Akım = 10



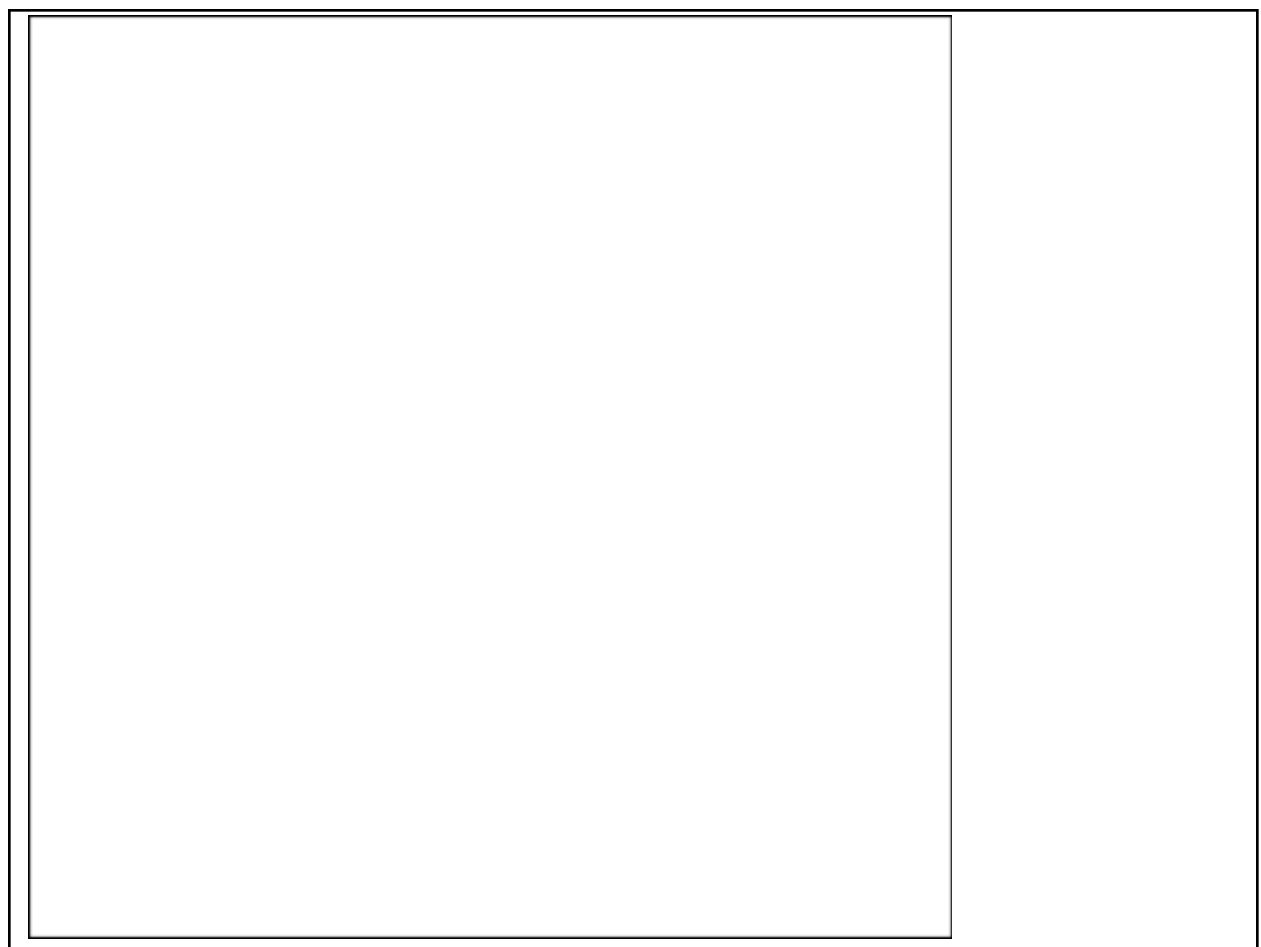
Hesaplama Sonrasi Yeni Degerler:  $Direnc = 2$        $Voltaj = 10$        $Akim = 5$



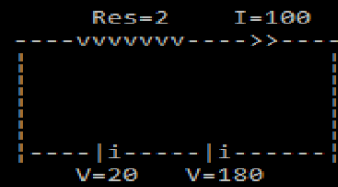




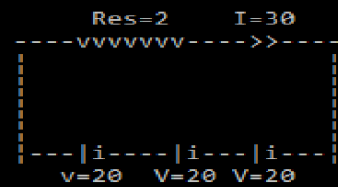
[illegible]



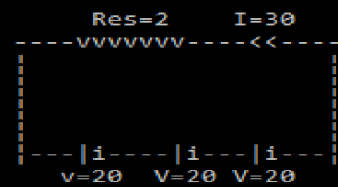
```
~~VOLT CARPIMI~~(*)=200
```



```
~~TOPLAM VOLTAJ~~(+)=60
```



```
~~FARK VOLTAJ~~(-)=-60(Akim Ters Yondedir)
```



```
~~ESIT ESIT MIDIR VOLTAJ~~(==)200?60
```

Voltaj Degerleri Esit Degildir False

```
~~ESIT DEGIL MIDIR VOLTAJ~~(!=)200?60
```

Voltaj Degerleri Esit Degildir True

--	--

Journal Pre-proof

~ 0-Volt Bulma ~      ~ 1-Akim Bulma ~      ~ 2-Direnc Bulma ~      ~ 3-Karisik Devre Cozumu ~

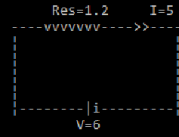
Islem Numarasi: 2

Akim ve Volt giriniz

5

6

Direnc=1.2



YAPMAK ISTEDIGINIZ ISLEMI SECINIZ

~ 0-Volt Bulma ~      ~ 1-Akim Bulma ~      ~ 2-Direnc Bulma ~      ~ 3-Karisik Devre Cozumu ~

Islem Numarasi: 3

Seri Direnc Sayisi Gir: 3

Seri Direnc Degerlerini Sirasiyla Giriniz: 6

Seri Direnc Degerlerini Sirasiyla Giriniz: 8

Seri Direnc Degerlerini Sirasiyla Giriniz: 9

Seri Direnclerin Toplami: 23

Paralel Direnc Sayisi Gir: 4

Paralel Direnc Degerlerini Sirasiyla Giriniz: 8

Paralel Direnc Degerlerini Sirasiyla Giriniz: 6

Paralel Direnc Degerlerini Sirasiyla Giriniz: 7

Paralel Direnc Degerlerini Sirasiyla Giriniz: 1

Paralel Direnclerin Toplami: 0.697095

Karisik Devrenin Toplam Direnci: 23.6971



-----EXCEPTION HANDLING-----

Girdiginiz Degerler Kontrol Edilecektir.

Direnc Degeri Giriniz: 5

Akim Degeri Giriniz: 6

Voltaaj Degeri Giriniz: 2

HATA---->>Hatali Guc Kaynagi Sectiniz

Girdiginiz Degerler Kontrol Edilecektir.

Direnc Degeri Giriniz: 0

Akim Degeri Giriniz: 1

Voltaaj Degeri Giriniz: 2

HATA---->>Kisa Devre Guc Kaynagini Devreden Cikartiniz

Girdiginiz Degerler Kontrol Edilecektir.

Direnc Degeri Giriniz: 1

Akim Degeri Giriniz: 0

Voltaaj Degeri Giriniz: 2

HATA---->>Devreye Guc Kaynagi Bagli Degildir Veya Acik Devre

Girdiginiz Degerler Kontrol Edilecektir.

Direnc Degeri Giriniz: 1

Akim Degeri Giriniz: 2

Voltaaj Degeri Giriniz: 0

HATA---->>Devreye Guc Kaynagi Bagli Degildir



-----EXCEPTION HANDLING-----

Girdiginiz Degerler Kontrol Edilecektir.

Direnc Degeri Giriniz: 1

Akim Degeri Giriniz: 5

Voltaj Degeri Giriniz: 5

Hata Algılanmadi

Girdiginiz Degerler Kontrol Edilecektir.

Direnc Degeri Giriniz: 9

Akim Degeri Giriniz: 6

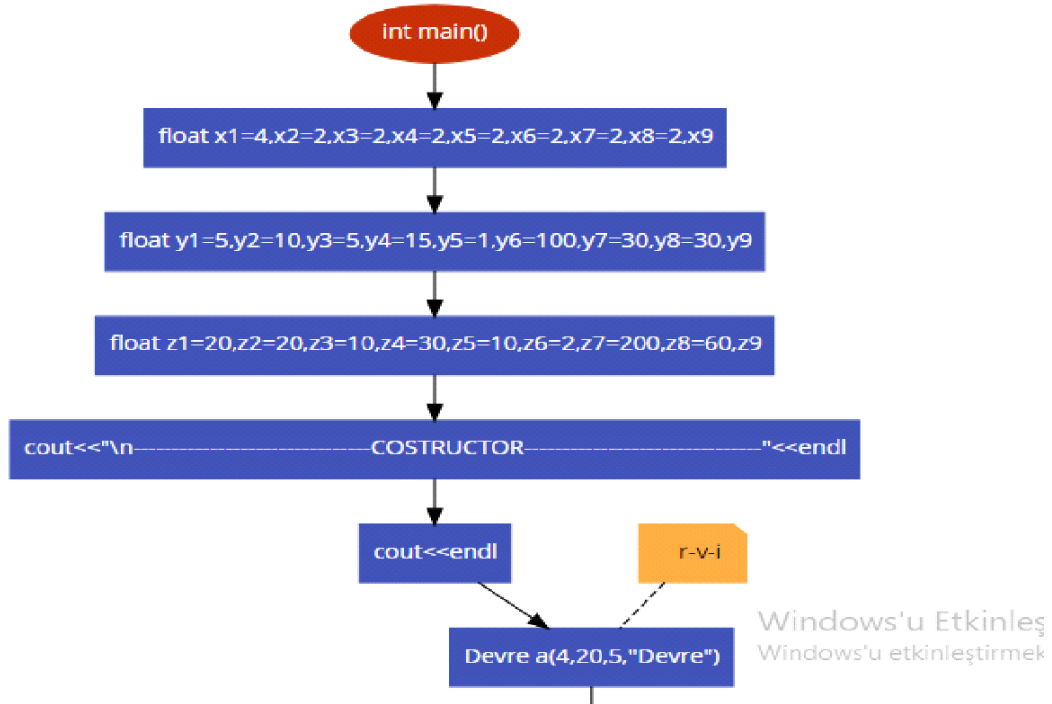
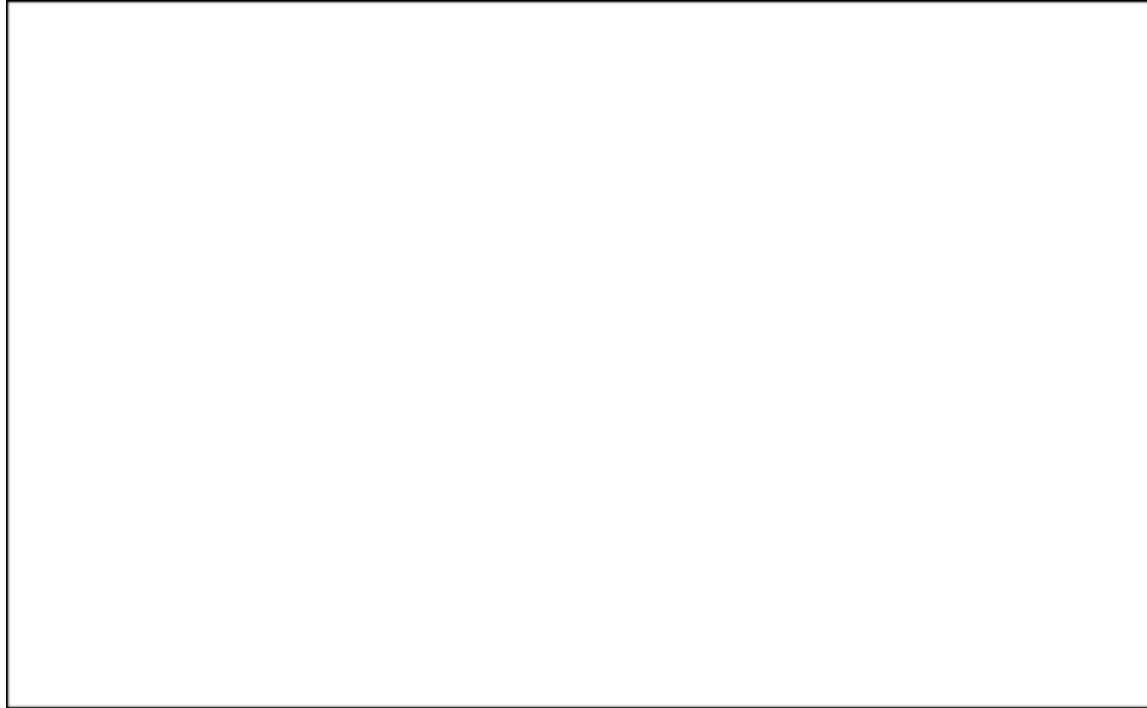
Voltaj Degeri Giriniz: 54

Hata Algılanmadi

Girdiginiz Degerler Kontrol Edilecektir.

Direnc Degeri Giriniz:

## Flowchart of the Program / Programın Akış Diyagramı





Devre a(4.20.5."Devre")

a.Print()

Devre b=a

b.Print()

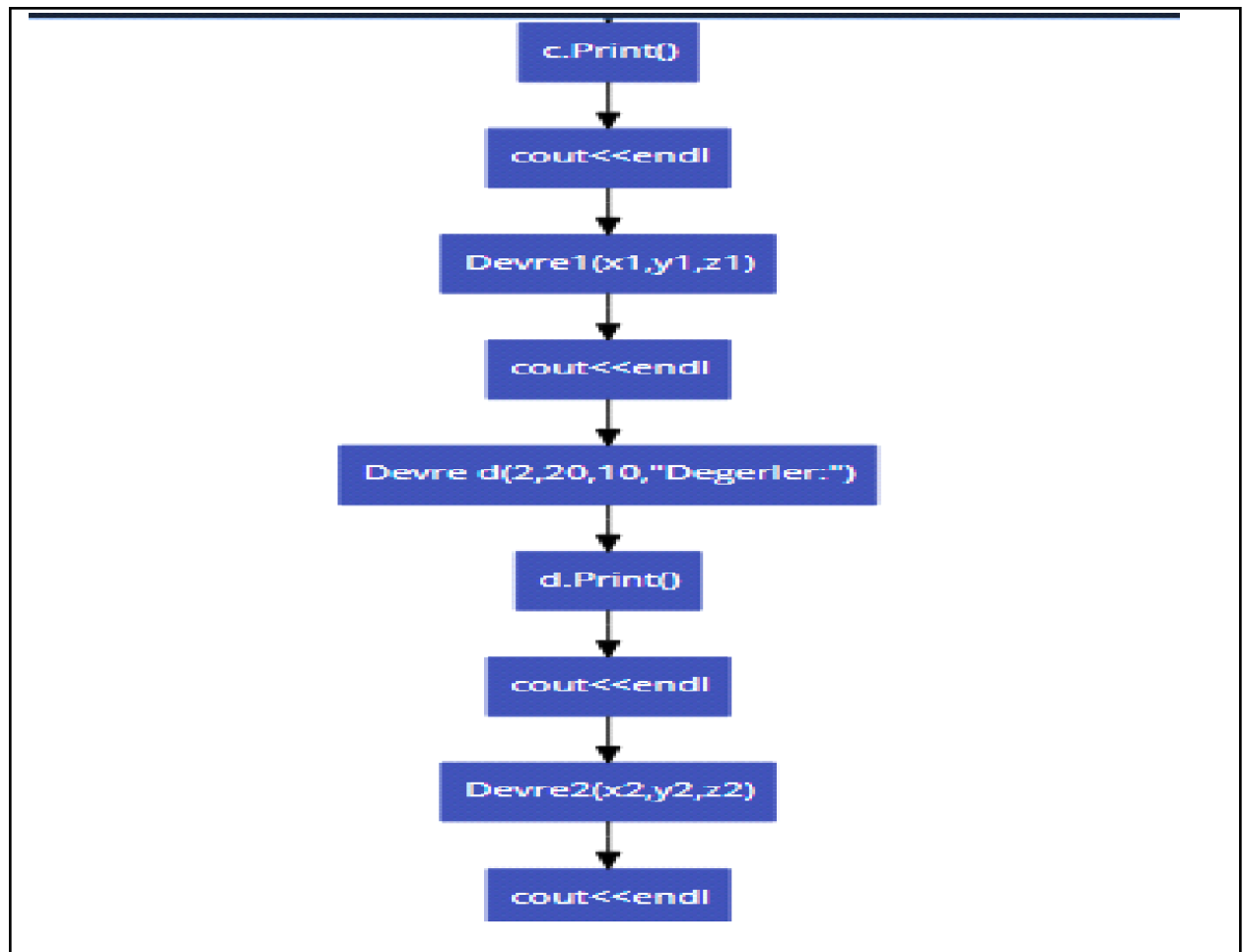
Devre c

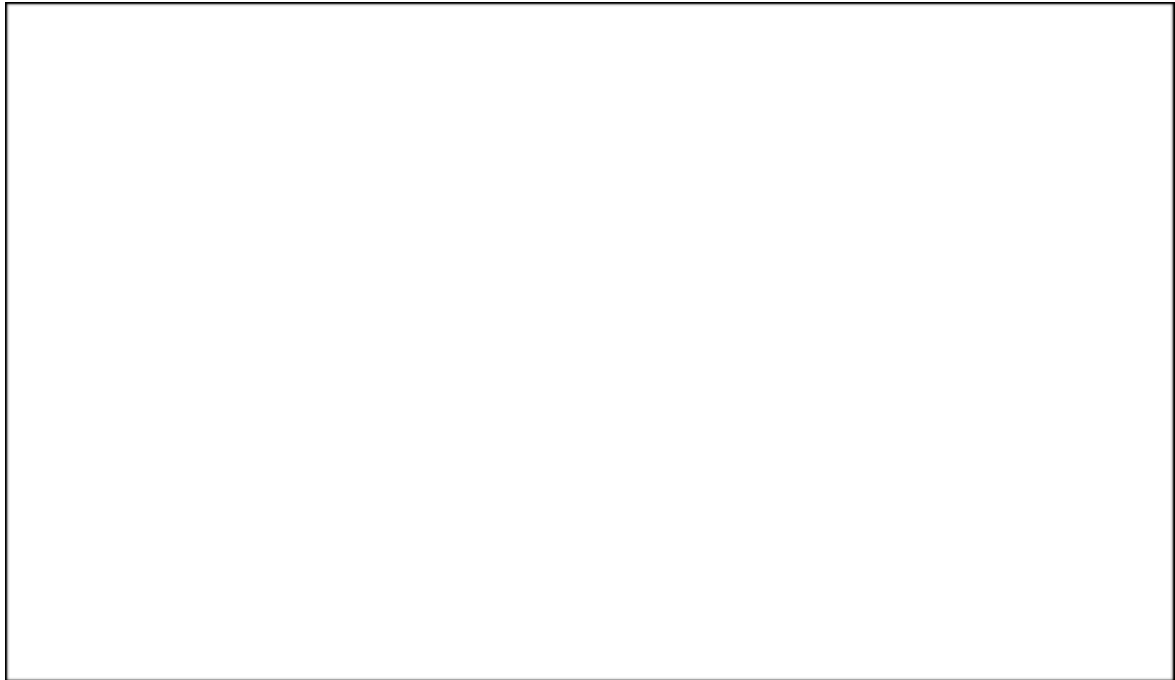
c=a

c.Print()

---







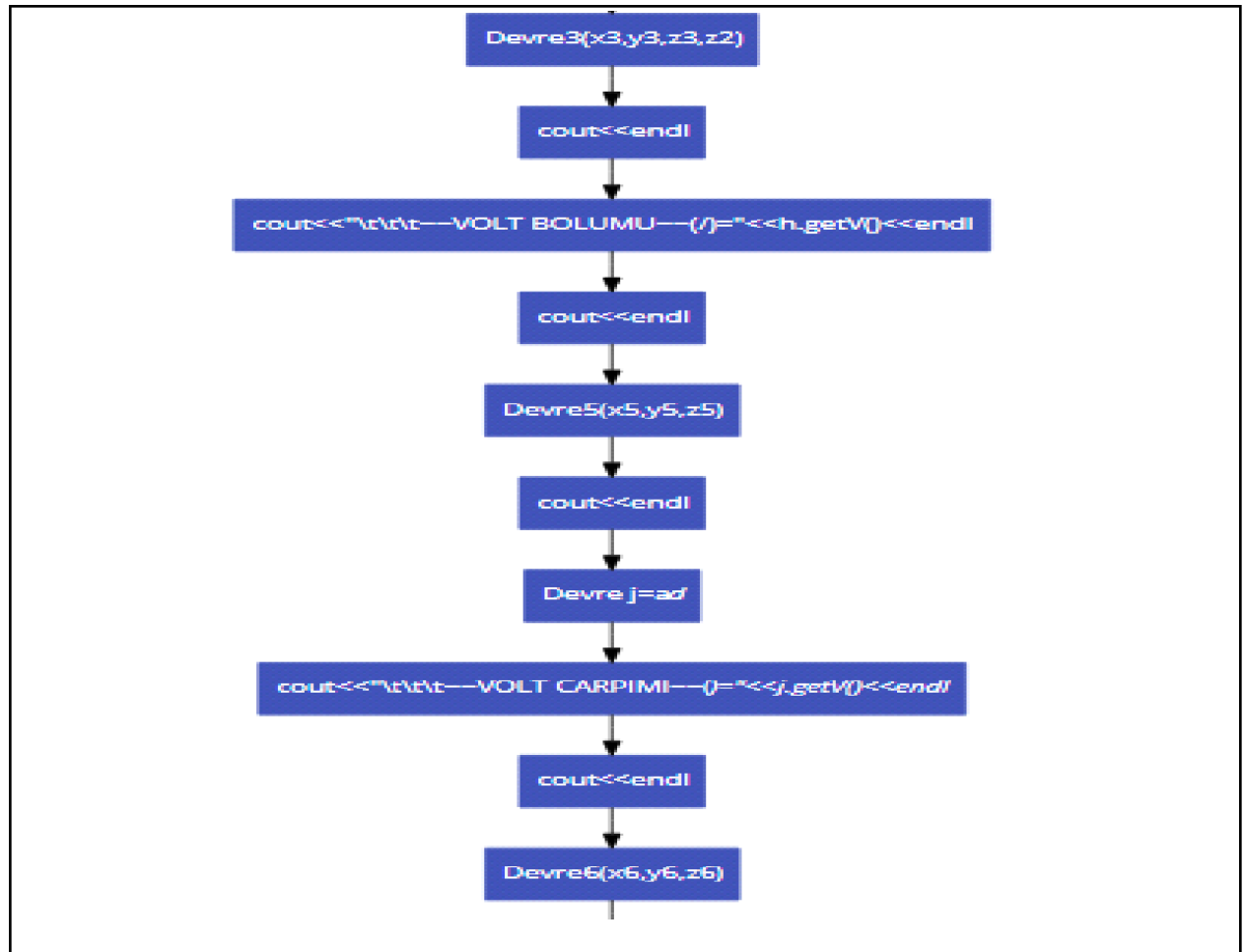
Windows'u E  
Windows'u etkin



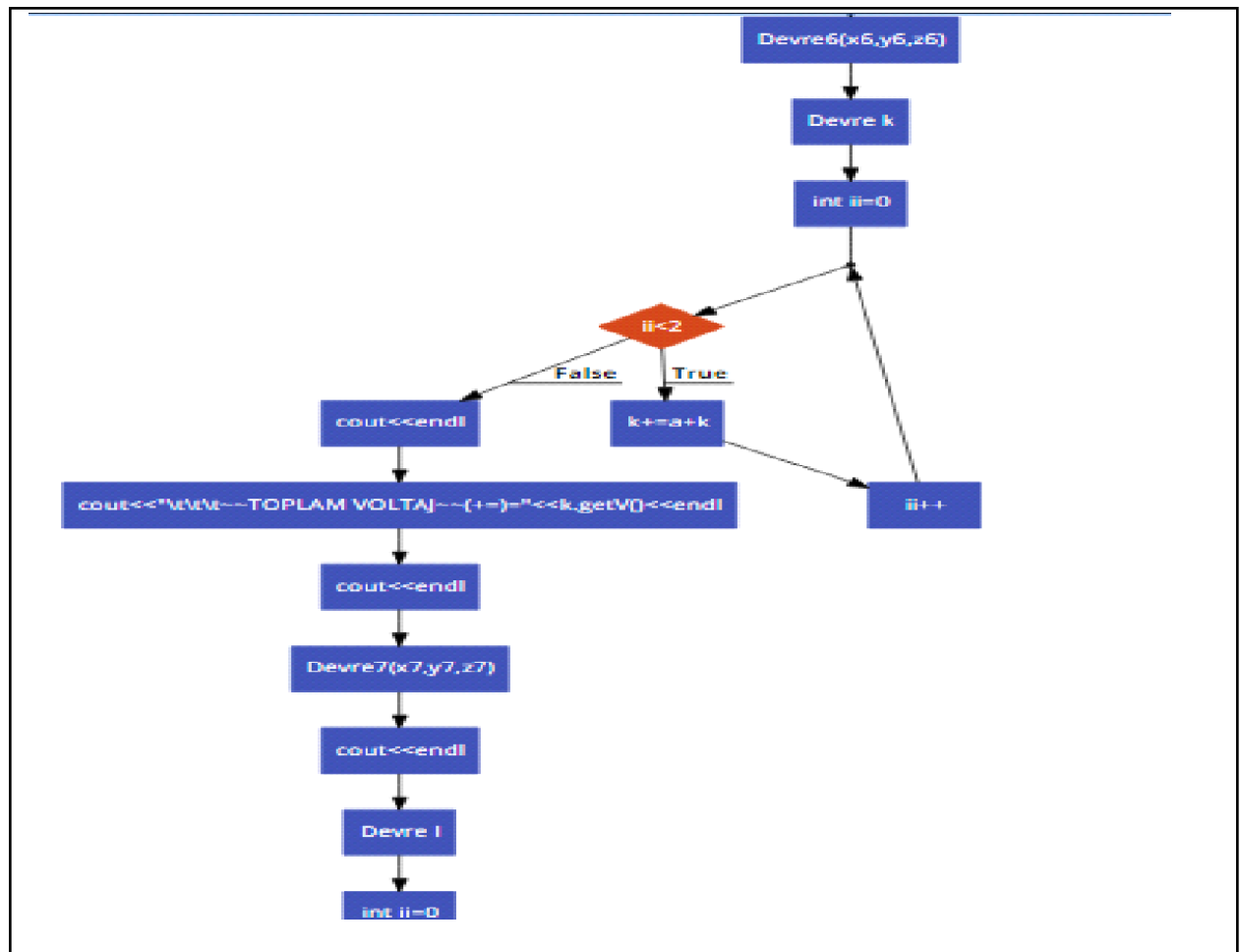




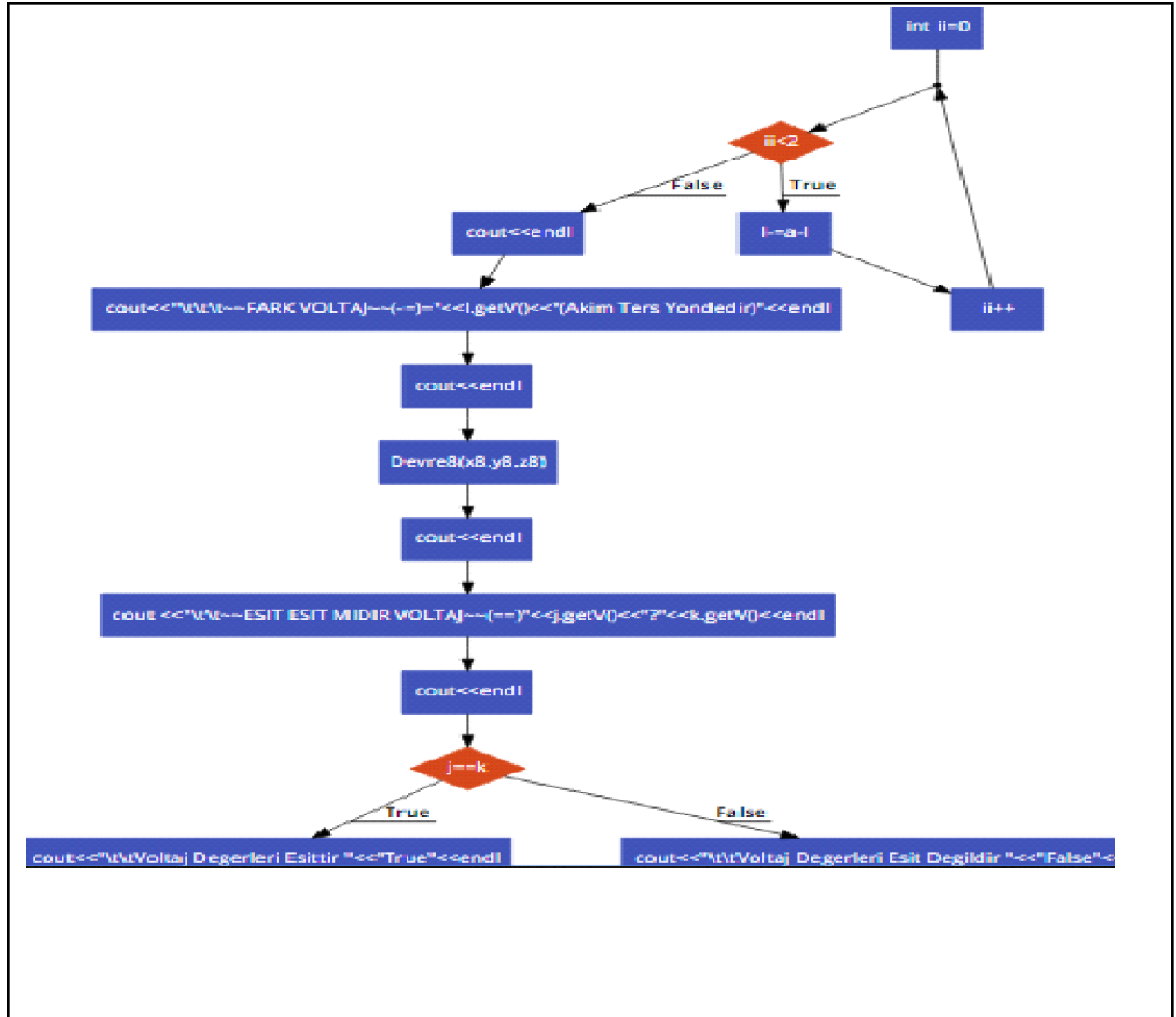












## Conclusion and Remarks / Sonuç ve Notlar

Kullanıcı dostu bir program yapmaya çalıştım. Hatta ayrıca zaman ayırıp kullanıcın verebileceği degerleri ekrana basan eşdeger devre çizimi yaptırdım. Diger kısımda da aynı şekilde girilen devre degerleri bir şekil üzerinde programda yapılan atama sayesinde gösterildi. Birçok yerde Anlaşılır olması ve herhangi bir hata olup olmadığı kontrol edilsin ve kapatılsın diye for içerisinde aynı donguyu seçim yapması için döndürdüm. Kullanıcı karşısına çıkan menüler haricinde br deger giirlirse hata verdiği ve hatayı bulması için exception handling kısmına yonelndiren program hazırladım. Yani basic düzey c++ bilen programımı okuyabilir ve hiç kullanmayı bilmeyenler bile exe dosyasını çalıştırabilip analiz yapabilirler.

Class içinde class tanımlarken kullanım şeklini bilmiyordum ama öğrendim ve gerçekten çok işime yaradı bu sayede içinde bulunan fonksiyonu çağırarak kullanmamı sağladı.

Program yazıyorken satır sayısı konusunda sıkıntı yasadım. Aslında projeye başlamadan önce kafamda planlarım ve yaparım veya yapamam. Ama programa bşlarken sadece ohm kanunu varen fonksiyonları çoğaltıp kullanıcı

dostu olması amacıyla ekrana deger ıktısı bastırmayı dusundum ve ekledim. Sanırım daha guzel oldu. Başka bir yerde problem yaşamadım.

## References / Kaynaklar

<https://code2flow.com/app>

<http://abl.gtu.edu.tr/hebe/w/?/view/989/211/71531484/downloads>

<https://www.tutorialspoint.com/index.htm>

Buckys C++ Programming Tutorials (Youtube channel)

C How to Program - Deitel

## Text of program/ Program kodu

-siteye eklendi-