Nesne Yönelimli Programlama

- C# ve Java dillerinde veri tipleri

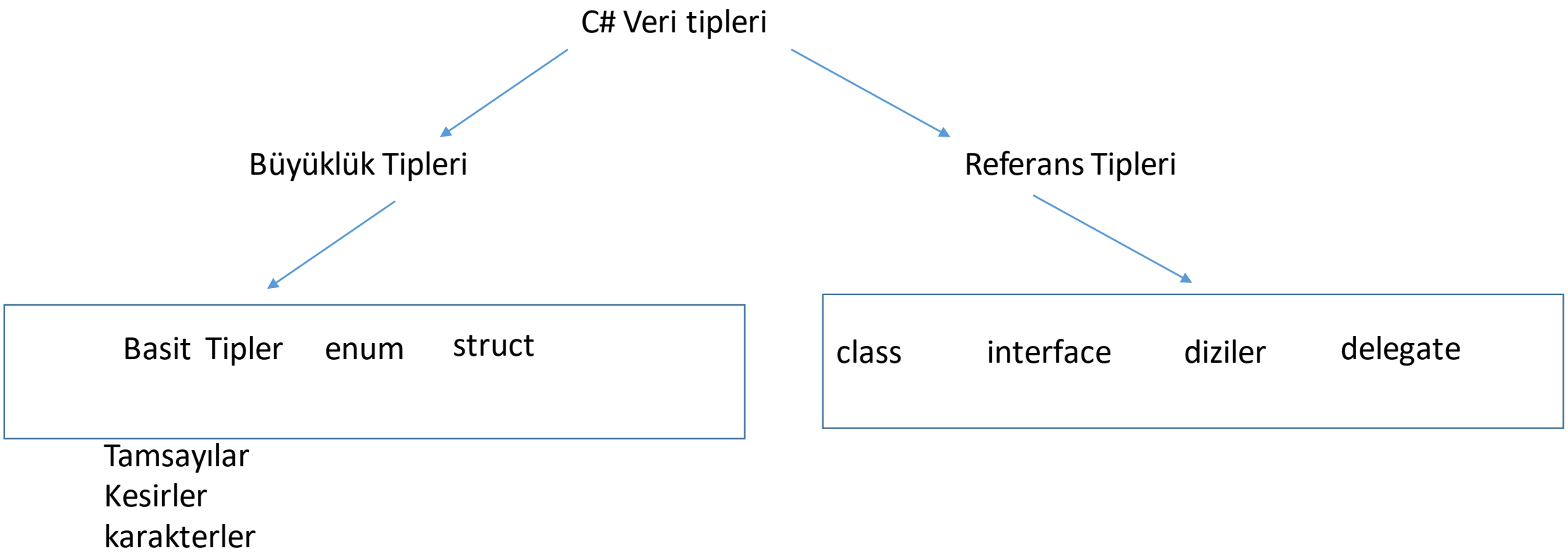
Bildirim

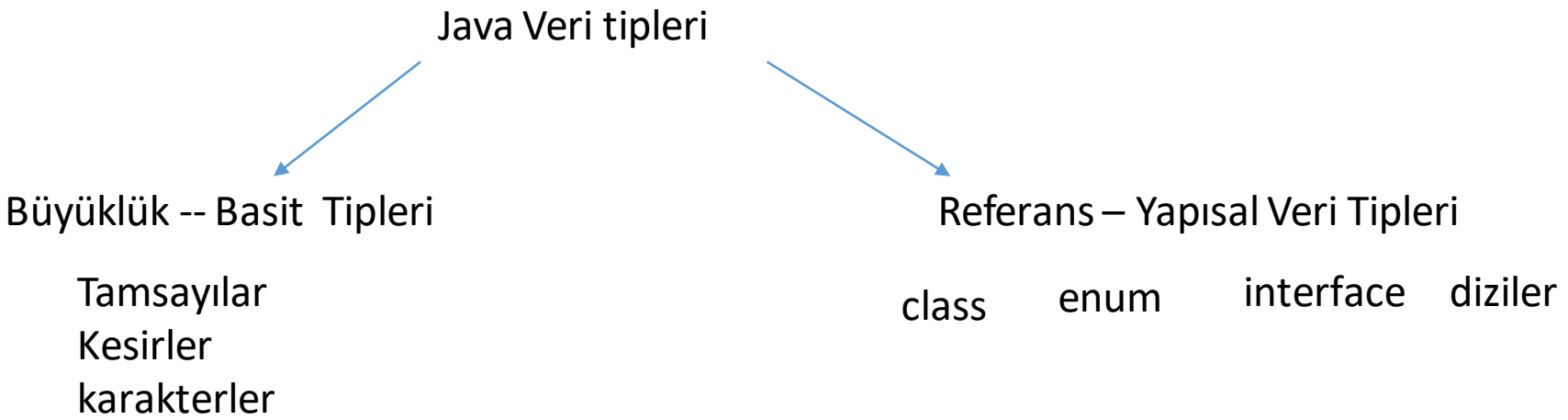
Veri tipi değişken/nesne adı;

Atama

değişken/nesne adı = değer;

Veri tipi değişken/nesne adı = değer;


C# Veri tipleri
Büyüklük Tipleri
Referans Tipleri
Basit Tipler
enum
struct
class
interface
diziler
delegate
Tamsayılar
Kesirler
karakterler


Java Veri tipleri
Büyüklük -- Basit Tipleri
Tamsayılar
Kesirler
karakterler
Referans – Yapısal Veri Tipleri
class
enum
interface
diziler

C# struct

```
struct Student
    {
        string name;
        public Student(string n)
        {
            name = n;
        }
        public void  Info() { Console.WriteLine("Student :" + name); }
        public override string ToString()
        {
            return name;
        }
    }

                Student s = new Student("Ali");
```

C# enum

```
enum Days { pazar, pazartesi, sali, carsamba,
persembe, cuma, cumartesi };
```

```
Days d = Days.carsamba;
```

C# delegate

```
delegate int Sum(int x, int y);

int topla(int x, int y)
{
     return x + y;
}

Sum t = topla;
Func<int, int, int> t2 = delegate (int x, int y) { return
x + y; };
```

C# interface

```
interface IStudent{void Info(); }
struct Student :IStudent
    {
        string name;
        public Student(string n)
        {
            name = n;
        }
        public void  Info() { Console.WriteLine("Student :" + name); }
        public override string ToString()
        {
            return name;
        }
    }
Student s = new Student("Ali");
IStudent st = s;// new Student("Veli");
```

C# dizi

```
Days[] din = { Days.pazar, Days.pazartesi,
Days.sali, Days.carsamba, Days.persembe, Days.cuma,
Days.cumartesi };
                foreach (Days x in din)
                    Console.WriteLine(x);
```

Java enum

```
enum Days { pazar, pazartesi, sali, carsamba,
persembe, cuma, cumartesi };
Days dd =Days.carsamba;
```

Java dizi

```
Days[] dd =Days.values();
for(Days x:dd)
      System.out.println(x);
```

Java enum

```
enum TrafficSignal {

    RED("wait",30), GREEN("go",30), YELLOW("slow down",15);
    String action;
    int bekle;
    public String getAction(){
        return this.action;
    }
    public int getDuration() {
    return bekle;
    }
     TrafficSignal(String action,int x){
        this.action = action;
        bekle =x;
    }
}
TrafficSignal[] signals = TrafficSignal.values();
for(TrafficSignal signal : signals){
        System.out.println("name : " + signal.name() + " action: "+ signal.getAction()+"Duration :"+signal.getDuration());
        }
```

Java interface

```
interface ITrafic{
            public String getAction();
}
enum TrafficSignal implements ITrafic {
    RED("wait",30), GREEN("go",30), YELLOW("slow down",15);
     String action;
    int bekle;
    public String getAction(){
        return this.action;}
    public int getDuration() {
    return bekle;}
TrafficSignal(String action,int x){
        this.action = action;
        bekle =x; }}

ITrafic [] dar =TrafficSignal.values();
for(ITrafic x : dar){
        System.out.println( " action: "+ x.getAction());
        }
```