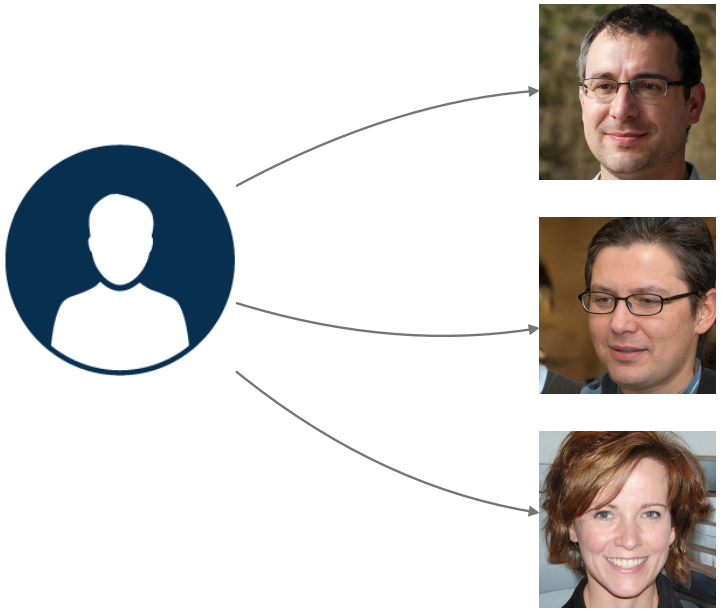
NESNEYE YÖNELİK PROGRAMLAMA

Sınıflar ve Nesneler



SINIF VE NESNE KAVRAMLARI

- Sınıf
  - Nesneye yönelik programlama
  - Struct benzeri
- Nesne
  - Örnek

YAPILAR (STRUCT)

- Yapı tanımı
  - Özellikler
  - Kullanıcı tanımlı değişkenler
  - ~~Metotlar~~
  - ~~Yazdırma~~
  - ~~Java~~

YAPILAR – SINIFLAR

```
class kisi{
    public String ad;
    public int yas;
}

public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.ad = "Emir";
        k.yas = 253;
        System.out.println(k);
    }
}
```

net.emirozturk.kisi@77459877

```
System.out.println(k.ad + " " + k.yas);
```

```
class kisi{
    public String ad;
    public int yas;
    public int no;
}
```

```
k.yas = 11121;
```

```
public class Main {
    public int sonIkiHane(kisi k){
        return k.no % 100;
    }
    public boolean ilkiBuyukMu(kisi k1,kisi k2){
        return sonIkiHane(k1)>sonIkiHane(k2);
    }

    public static void main(String[] args){

    }
}
```

YAPILAR – SINIFLAR

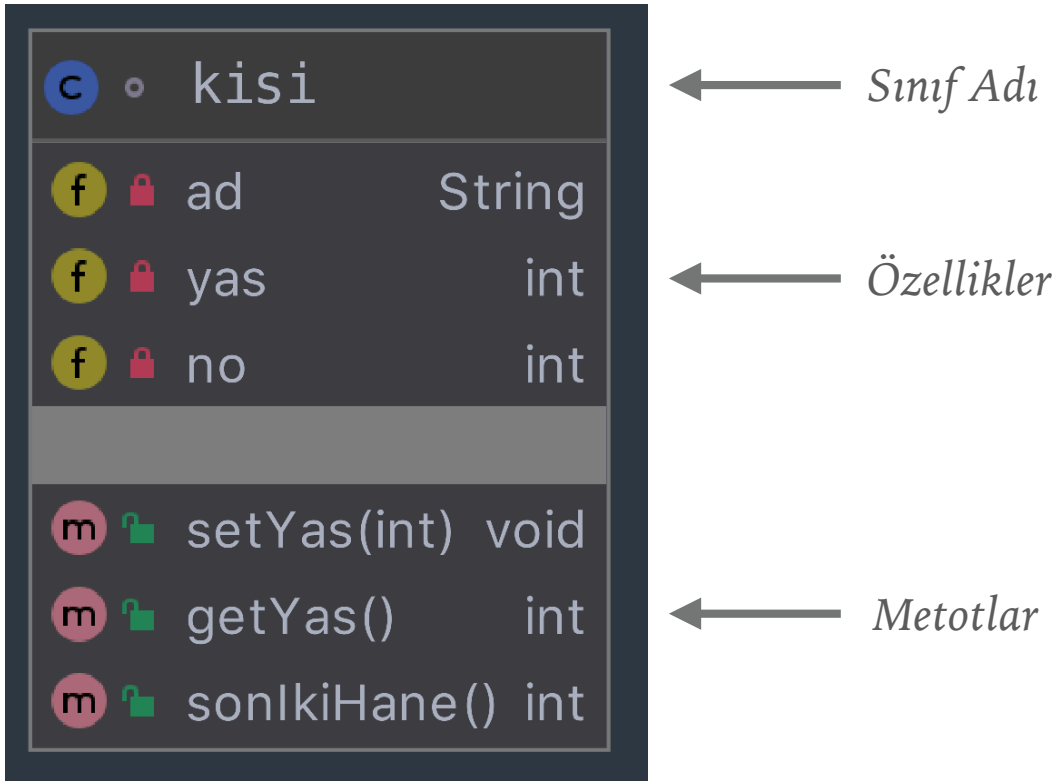
- Yapı içeriğinin doğruluğu
- Yapıyı kullanan fonksiyonların üzerindeki değişiklik

YAPILAR – SINIFLAR

- Doğruluğun sağlanması
  - Erişim belirleyicileri
  - Erişim metotları
  - Yapı ile ilgili metotlar

SINIF BİLDİRİMİ

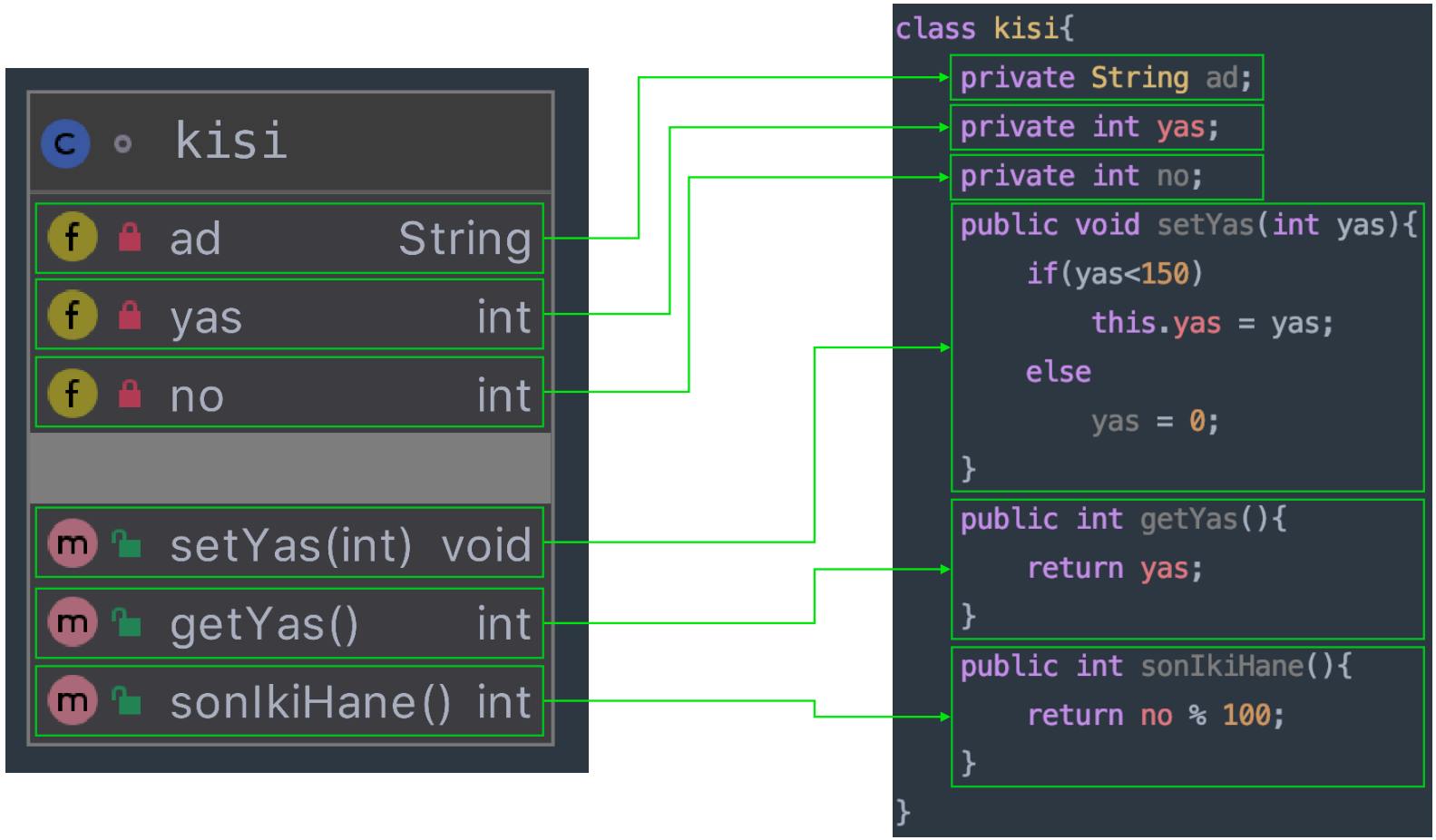
- Sınıflar
  - Sınıf ismi
  - Özellikler (~~değişkenler~~)
  - Metotlar (~~fonksiyonlar~~)



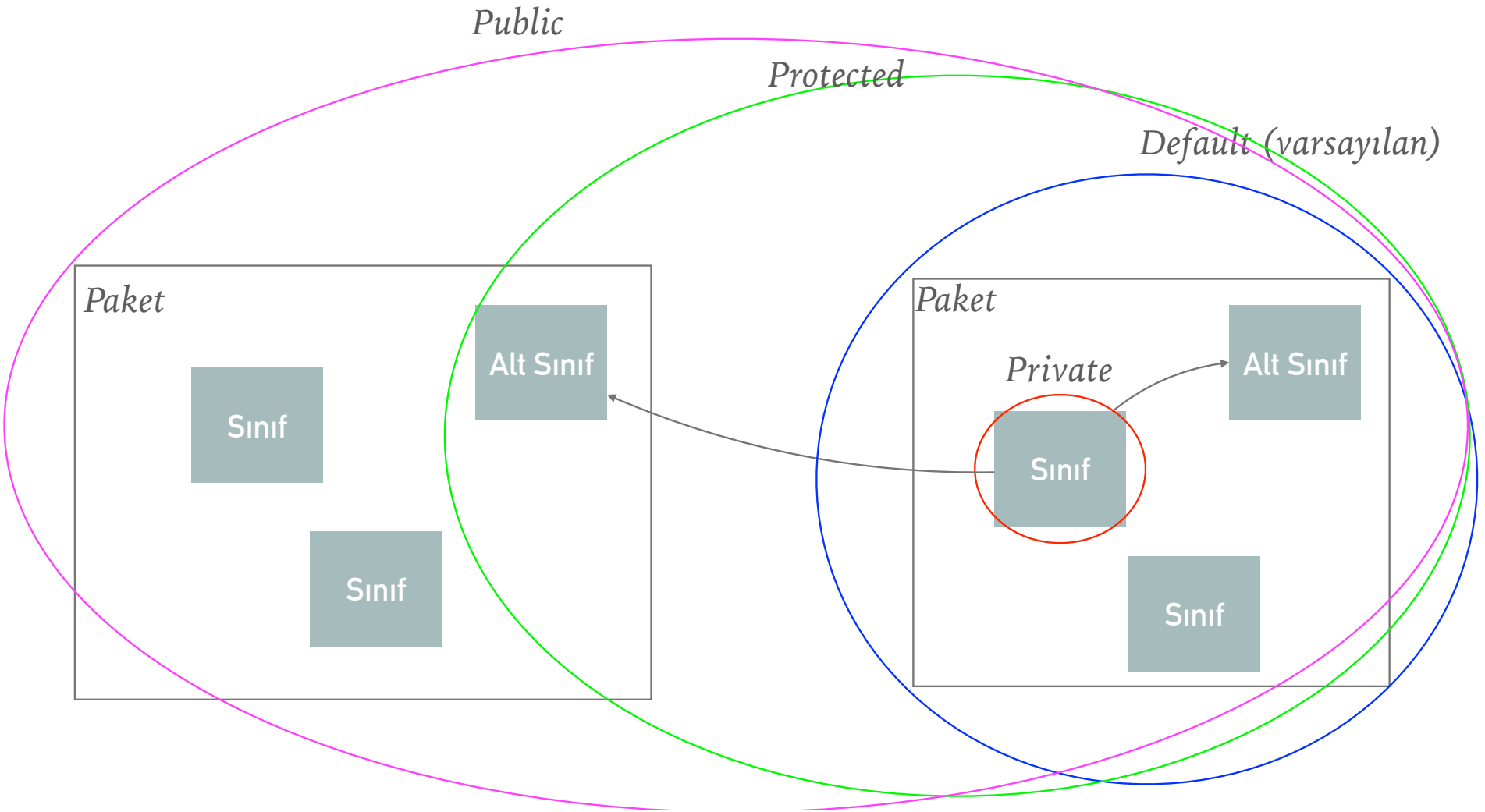
SINIF BİLDİRİMİ (JAVA)

```
class SınıfAdı {

    ErişimTürü DeğişkenTürü ad;
    ErişimTürü DeğişkenTürü ad;
    ErişimTürü DeğişkenTürü ad;

   ErişimTürü DönüşTipi İsim(Tür ad){
        //İşlemler
    }
    ErişimTürü DönüşTipi İsim(){
        //İşlemler
    }
}
```

SINIF BİLDİRİMİ



ERİŞİM BELİRLEYİCİLERİ



ERİŞİM BELİRLEYİCİLERİ

```java
class kisi{
    public String ad;
    public int yas;
    public int no;
}
public class Main {
    public int sonIkiHane(kisi k){
        return k.no % 100;
    }
    public boolean ilkiBuyukMu(kisi k1,kisi k2){
        return sonIkiHane(k1)>sonIkiHane(k2);
    }

    public static void main(String[] args){

    }
}
```

```java
class kisi{
    private String ad;
    private int yas;
    private int no;
    public void setYas(int yas){
        if(yas<150)
            this.yas = yas;
        else
            yas = 0;
    }
    public int getYas(){
        return yas;
    }
    public int sonIkiHane(){
        return no % 100;
    }
}
```

ERİŞİM DÜZEYLERİ

```
class kisi{
    private String ad;
    private int yas;
    private int no;
    public void setYas(int yas){
        if(yas<150)
            this.yas = yas;
        else
            yas = 0;
    }
    public int getYas(){
        return yas;
    }
    public int sonIkiHane(){
        return no % 100;
    }
}
```

```
public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.yas = 30;
    }
}
```

```
public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.setYas(30);
    }
}
```

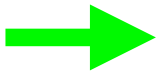
ELEMAN FONKSİYONLAR (METOTLAR)

- Sınıf içerisinde
- Tüm elemanlara erişim
- Aşırı yüklenebilirler

ELEMAN FONKSİYONLAR (METOTLAR)

```
class kisi{
    public String ad;
    public int yas;
}

public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.ad = "Emir";
        k.yas = 253;
        System.out.println(k);
    }
}
```

```
System.out.println(k.ad + " " + k.yas);
```



```
class kisi{
    private String ad;
    private int yas;
    private int no;
    public void setYas(int yas){
        if(yas<150) this.yas = yas;
        else yas = 0;
    }
    public int getYas(){ return yas; }
    public int sonIkiHane(){ return no % 100; }

}
```

```
public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        System.out.println(k.yazdir());
    }
}
```

YAPICI FONKSİYONLAR

- Bildirim anında çağırılır
- İlk değer atama
- İsim
- Dönüş türü
- Aşırı yüklenebilirler

YAPICI FONKSİYONLAR

```java
class kisi{
    private String ad;
    private int yas;
    private int no;

    public kisi(){

    }

    public kisi(String ad,int yas,int no){
        this.ad = ad;
        this.yas = yas;
        this.no = no;
    }
    public void setYas(int yas){
```

```java
public class Main {
    public static void main(String[] args){
        kisi k = new kisi();
        k.ad = "Emir";
        k.yas = 253;
        System.out.println(k);
    }
}
```

```java
public class Main {
    public static void main(String[] args){
        kisi k1 = new kisi();
        kisi k2 = new kisi( ad: "Emir", yas: 255, no: 203022);
    }
}
```

YIKICI FONKSİYONLAR

- Yaşam döngüsü
- Bellek iadesi
- Bitirilmesi gereken işlemler
  - Dosyalar
  - Veritabanları
  - Ağ bağlantıları
- Non - deterministik
- Aşırı yüklenemezler
- Değer almaz / döndürmezler

ÇÖP TOPLAYICI (GARBAGE COLLECTOR)

- Garbage Collector
- Heap
- Nesiller
  - Eden
  - Survivor
  - Tenured
  - Permanent

*https://medium.com/@tugrulbayrak/jvm-garbage-collector-nedir-96e76b6f6239

*https://stackoverflow.com/questions/2129044/java-heap-terminology-young-old-and-permanent-generations