

Javan suojusmääreet

- `public`
- `package`
- `protected`
- `private`

Suojusmääreiden vaikutusalueet

Viittauspaikka	private	package	protected	public
Luokka itse	X	X	X	X
<i>Samassa pakkauksessa oleva</i>				
perijä		X	X	X
ei-perijä		X	X	X
<i>Eri pakkauksessa oleva</i>				
perijä			X	X
ei-perijä				X

Javan luokkamekanismin erityispiirteitä

- Staattinen sisäluokka
 - liittyy syntaktisesti ulkoluokkaansa
- Esiintymäkohtainen sisäluokka
 - ei voi olla olemassa ilman ulkoluokkansa instanssia
- Nimetön luokka
 - luodaan lennossa kertakäyttöisesti
- Literaaliluokka `enum`
 - voidaan konstruoida vain omassa määrittelylohkossaan

Staattinen sisäluokka

- Normaali luokkamääritys, jonka otsikkoon on kirjoitettu `static`
- Liittyy kiinteästi ulkoluokkaan
- Sisäluokalla vapaa pääsy ulkoluokan piirteisiin
- Sisäluokkatyyppiin viitataan pistenotaatiolla: `Ulkoluokka.Sisäluokka`
- Esimerkkejä
 - `Character.Subset`
 - `Map.Entry`

Esiintymäkohtainen sisäluokka

- Tiukka sidos sisäluokan olion ja sitä vastaavan ulkoluokan olion välille
 - sisäluokan olio ei voi olla olemassa ilman ulkoluokan oliota
- Sisäluokan oliolla vapaa pääsy ulkoluokan olion piirteisiin

Esimerkki: Pankkitili 1(2)

```
public class Pankkitili {
    private long numero, saldo;
    private List<Tapahtuma> tilitapahtumat;
    public enum Tapahtumatyyppi { TALLETUS, NOSTO };

    public class Tapahtuma {
        private Tapahtumatyyppi tapahtumatyyppi;
        private long saldomuutos;

        public Tapahtuma(Tapahtumatyyppi t, long s) {
            tapahtumatyyppi = t; saldomuutos = s;
        }

        public String toString() {
            return numero + ": " + tapahtumatyyppi + "(" +
                saldomuutos + ")";
        }
    }
}
```

Esimerkki: Pankkitili 2(2)

```
public Pankkitili(long numero) {
    this.numero = numero; this.saldo = 0L;
    this.tilitapahtumat =
        new LinkedList<Tapahtuma>();
}
public void pano(long rahamäärä) {
    saldo += rahamäärä;
    tilitapahtumat.add(new Tapahtuma(
        Tapahtumatyyppi.TALLETUS, rahamäärä));
}
public void nosto(long rahamäärä) {
    saldo -= rahamäärä;
    tilitapahtumat.add(new Tapahtuma(
        Tapahtumatyyppi.NOSTO, rahamäärä));
}
}
```

this-viittauksen semantiikka

- Sisäluokan oliossa this viittaa normaalisti ko. olioon
- Ulkoluokan olioon viitataan pistenotaatiolla
 - esim. `Pankkitili.this`

Nimetön luokka

```
JFrame ikkuna = new JFrame("Akkuna");
ikkuna.addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
});
```

Literaali luokka enum

- Käyttö
 - luetellut vakiot (kuten C-kielessä)
 - aito luokka (toisin kuin C-kielessä)
 - konstruktori
 - jäsenmuuttujat
 - metodit
- Literaali luokan rajoitteet
 - ei voida periä
 - ei voida konstruoida kuin omassa määrittelylohkossa
- periytyy luokasta `java.lang.Enum`

Luetellut vakiot

```
enum Presidentti {
    STÄHLBERG, RELANDER, SVINHUFVUD,
    KALLIO, RYTI, MANNERHEIM, PAASIKIVI,
    KEKKONEN, KOIVISTO, AHTISAARI, HALONEN };

Presidentti p = Presidentti.PAASIKIVI;
System.out.println(p);
boolean nykyinen =
    (p == Presidentti.HALONEN);
```

Esimerkki: Kortti 1(2)

```
public final class Kortti {
    public enum Arvo {
        ASSA('A', 1), KAKSI(2), KOLME(3), NELJÄ(4), VIISI(5),
        KUUSI(6), SEITSEMÄN(7), KAHDEKSAN(8), YHDEKSÄN(9),
        KYMMENEN(10), JÄTKÄ('J', 11), KUNINGATAR('Q', 12),
        KUNINGAS('K', 13);
        private String symboli;
        private int numeroarvo;

        private Arvo(char s, int a) {
            symboli = Character.toString(s); numeroarvo = a; }
        private Arvo(int a) {
            symboli = Integer.toString(a); numeroarvo = a; }
        public String annaSymboli() { return symboli; }
        public int annaNumeroarvo() { return numeroarvo; }
    }
}
```

Esimerkki: Kortti 2(2)

```
public enum Maa {
    RISTI('✠'), RUUTU('♠'), HERTTA('♥'), PATA('♣');
    private String symboli;
    private Maa(char s) { symboli = Character.toString(s); }
    public String annaSymboli() { return symboli; }
}

private final Arvo arvo;
private final Maa maa;
private Kortti(Arvo arvo, Maa maa) {
    this.arvo = arvo; this.maa = maa; }
public Arvo arvo() { return arvo; }
public Maa maa() { return maa; }
public String toString() {
    return maa.annaSymboli() + arvo.annaSymboli();
} }
```