OBJEKTUMORIENTÁLT PROGRAMOZÁS 2008.
2. GYAKORLAT

CÉL:

- A Java API dokumentáció használata
- Egységbezárás (Encapsulation) tanulmányozása
- Csomagok (packages) használata

1. Feladat:

a. Tanulmányozza a Java API dokumentációját

```
http://java.sun.com/javase/6/docs/api/
```

b. Keresse meg a **java.util** csomagban a **Scanner** osztályt és tanulmányozza.

c. Használja a **Scanner** osztályt beolvasásra:

```
Scanner sc = new Scanner(System.in);
System.out.print("Kerek egy egesz szamot: ");
int i = sc.nextInt();
System.out.println( "Egesz szam: "+i );
```

d. Próbáljon nem egész számot beírni. Mi történik?

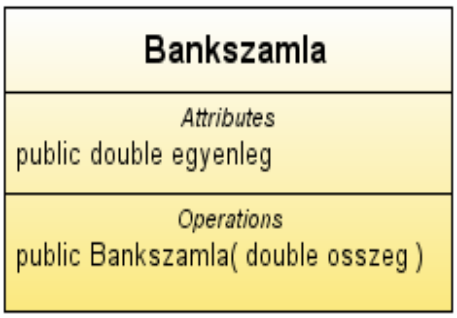
2. Feladat

Cél:

- a nyilvános adattagok veszélyességének bemutatása
- az egységbezárás szükségessége

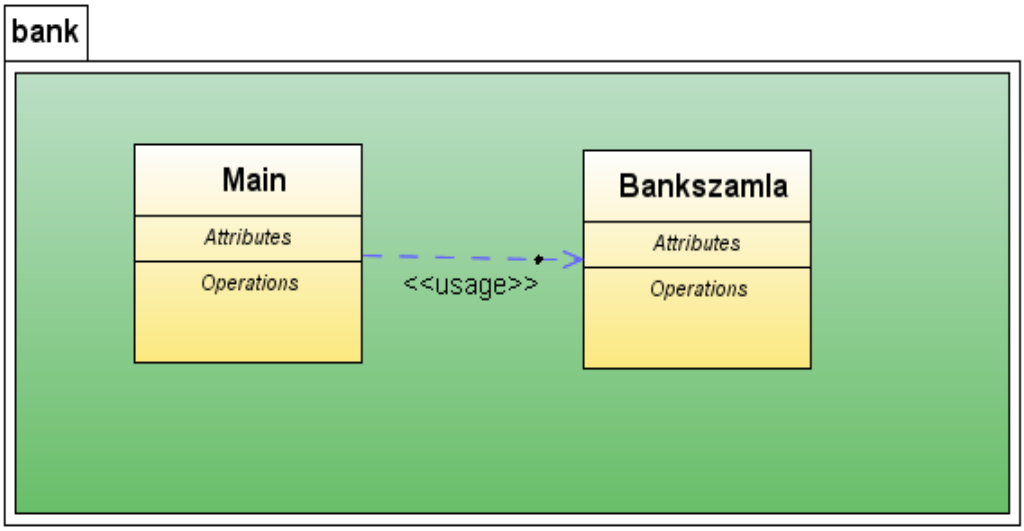
Készítsen egy új **bank** nevű projektet.

Adjon hozzá egy olyan bankszámla osztályt, amelyben az egyenleg nyilvános adattag. A bankszámla kezdőegyenlegének beállítását a konstruktor végzi. (ld. 1. labor)



A bankszámla osztálynak egyetlen üzleti szabályt kell betartania: *Az egyenleg értéke nem lehet kisebb, mint 0.*

OBJEKTUMORIENTÁLT PROGRAMOZÁS 2008.
2. GYAKORLAT



A **Main** osztály **main** metódusában végezzük el a következő műveleteket:

- Készítsünk egy bankszámlát 1000 RON kezdőegyenleggel.
- Írassuk ki az egyenleg értékét.
- Vegyünk ki 1500 RON-t
- Írassuk ki az egyenleg értékét.

Segítség:

```
Bankszamla szamla = new Bankszamla( 1000 );
System.out.println( "Egyenleg: " +szamla.egyenleg);
szamla.egyenleg -= 1500.
System.out.println( "Egyenleg: " +szamla.egyenleg);
```

Kérdések:

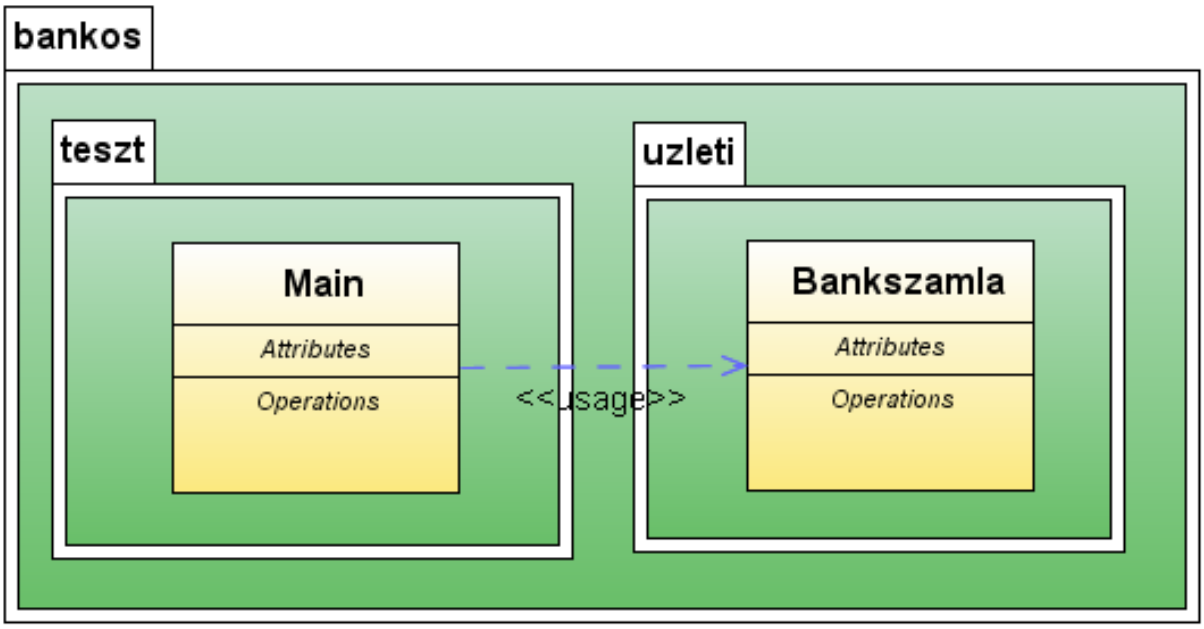
- Betartja ez az osztály az üzleti szabályt?
- Alakítsa át úgy, hogy helyesen működjön.

3. Feladat

- Készítsen másolatot a 2. feladathoz készített projektről. (Jobb egérgomb a projektnévre és Copy)

- Az előző feladatban a két osztály ugyanahhoz a **bank** nevű csomaghoz tartozott. Hozzuk létre az alábbi diagramnak megfelelő csomagstruktúrát és helyezzük át a már meglévő osztályainkat.
- Az osztályok áthelyezését végezze a *Code Refactoring* lehetőséggel (jobb kattintás az osztálynévre –>Refactor -->Move)
- Ellenőrizze a forrásállományokat. Mi lehet az **import** utasítás szerepe?

OBJEKTUMORIENTÁLT PROGRAMOZÁS 2008.
2. GYAKORLAT



4. Feladat

Készítsen egy képernyőpont ábrázolásához szükséges `Pont` osztályt.

Adattagok:

- `x,y: int`

Metódusok:

- Konstruktor: Inicializálja x és y adattagokat úgy, hogy csak 1 és 2000 közötti értékeket fogadjon el.
- `getX(), setX()` metódusok az x koordináta lekérdezése/beállítása.
- `getY(), setY()` metódusok az y koordináta lekérdezése/beállítása.
- `toString()` metódus

A **Main** osztály **main** metódusában végezze el a következő műveleteket:

- Hozzon létre egy p1 és egy p2 nevű **Pont** típusú referenciát és inicializálja ezeket a (10,20), illetve a (30,40) értékekkel.
- Írassa ki a pontokat.
  Végezze el a `p1=p2;` értékadást majd ezt követően írassa ki a pontokat. Mit észlelt?
- Végezze el a Pont p3=p1; értékadást majd írassa ki a p3-t.