

## OBJEKTUMORIENTÁLT PROGRAMOZÁS 2009.

### 3. GYAKORLAT

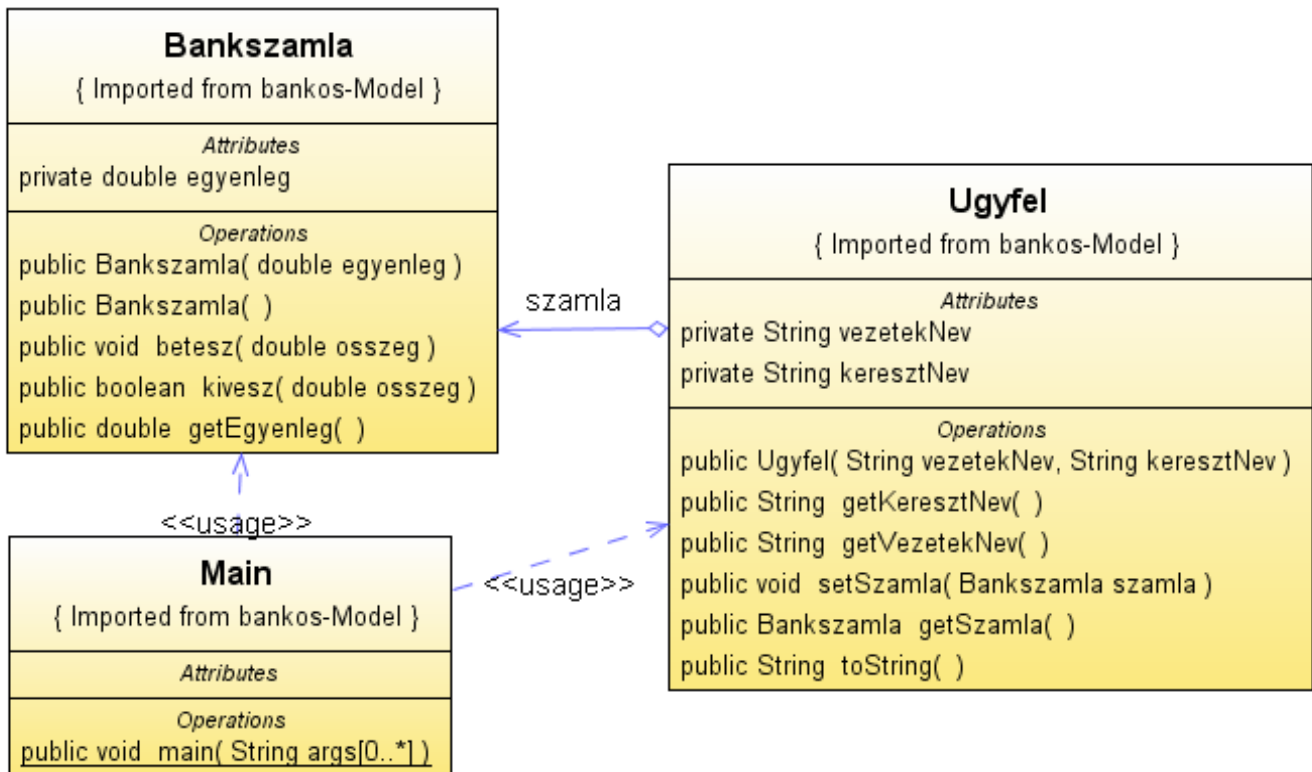
#### CÉL:

- Osztályok közötti kapcsolatok: tartalmazási kapcsolat, használati kapcsolat
- Metódusok hívása - üzenetküldés

#### 1. Feladat

Készítsen egy bankos alkalmazást. Az alkalmazásnak *számlákkal* és *ügyfelekkel* kell dolgoznia. A Bankszamlá osztály adott (ld. 1. gyakorlat).

- Készítse el az *Ugyfel* osztályt az alábbi diagram alapján.
- Adott egy utasítássorozat, ezt másolja be a *Main* osztály *main* metódusába, majd ellenőrizze a programja kimenetét. Ha helyesen dolgozott a megadott kimenetet kell kapnia.



**//Bankszamlá típusú referencia, mindkét ügyfélnél ezt használjuk a példány készítéséhez**

```
Bankszamlá szamlá;
```

**//Első ügyfel**

```
Ugyfel ugyfel1 = new Ugyfel("Demeter", "Ilona");
szamlá = new Bankszamlá( 1000 );
ugyfel1.setSzamlá(szamlá);
System.out.println( ugyfel1 );
```

**//Második ügyfel**

```
Ugyfel ugyfel2 = new Ugyfel("Mate", "Janos");
szamlá = new Bankszamlá();
ugyfel2.setSzamlá( szamlá );
```

## OBJEKTUMORIENTÁLT PROGRAMOZÁS 2009.

### 3. GYAKORLAT

---

```
System.out.println( ugyfel2 );

//Első ügyfél műveletek
System.out.println(ugyfel1.getVezetekNev()+" "+
    ugyfel1.getKeresztNev()+" betesz 1450 RON-t");
ugyfel1.getSzamla().betesz(1450);
System.out.println( ugyfel1 );

//Második ügyfél műveletek
System.out.println(ugyfel2.getVezetekNev()+" "+
    ugyfel2.getKeresztNev()+" betesz 2000 RON-t");
ugyfel2.getSzamla().betesz(2000);
System.out.println( ugyfel2 );

System.out.println(ugyfel2.getVezetekNev()+" "+
    ugyfel2.getKeresztNev()+" kivesz 450 RON-t");
ugyfel2.getSzamla().kivesz(450);
System.out.println( ugyfel2 );
```

#### Kimenet:

```
Demeter Ilona egyenlege 1000.0 RON
Mate Janos egyenlege 0.0 RON
Demeter Ilona betesz 1450 RON-t
Demeter Ilona egyenlege 2450.0 RON
Mate Janos betesz 2000 RON-t
Mate Janos egyenlege 2000.0 RON
Mate Janos kivesz 450 RON-t
Mate Janos egyenlege 1550.0 RON
```

#### Kérdések:

- Hány számlája lehet egy ügyfélnek?
- Azonosítsa a diagramon a használati, illetve a tartalmazási kapcsolatokat!

## 2. Feladat

Adott egy veremosztály váza. Egészítse ki a metódusokat, majd tesztelje a vermet!

```
public class Stack {
    private char elements[];
    private int size; //maximalis meret
    private int sp; //verem tetejének indexe

    public Stack( int size ){
        this.size = size;
        this.elements = new char[ size ];
        this.sp = -1;
    }
}
```

## OBJEKTUMORIENTÁLT PROGRAMOZÁS 2009.

### 3. GYAKORLAT

---

```
public void push( char c ){
    //
}

public char pop(){
    //
}

public char top(){
    //
}

public boolean isEmpty()
    //
}

public boolean isFull(){
    //
}
}
```