

OBJEKTUMORIENTÁLT PROGRAMOZÁS 2008.

4. GYAKORLAT

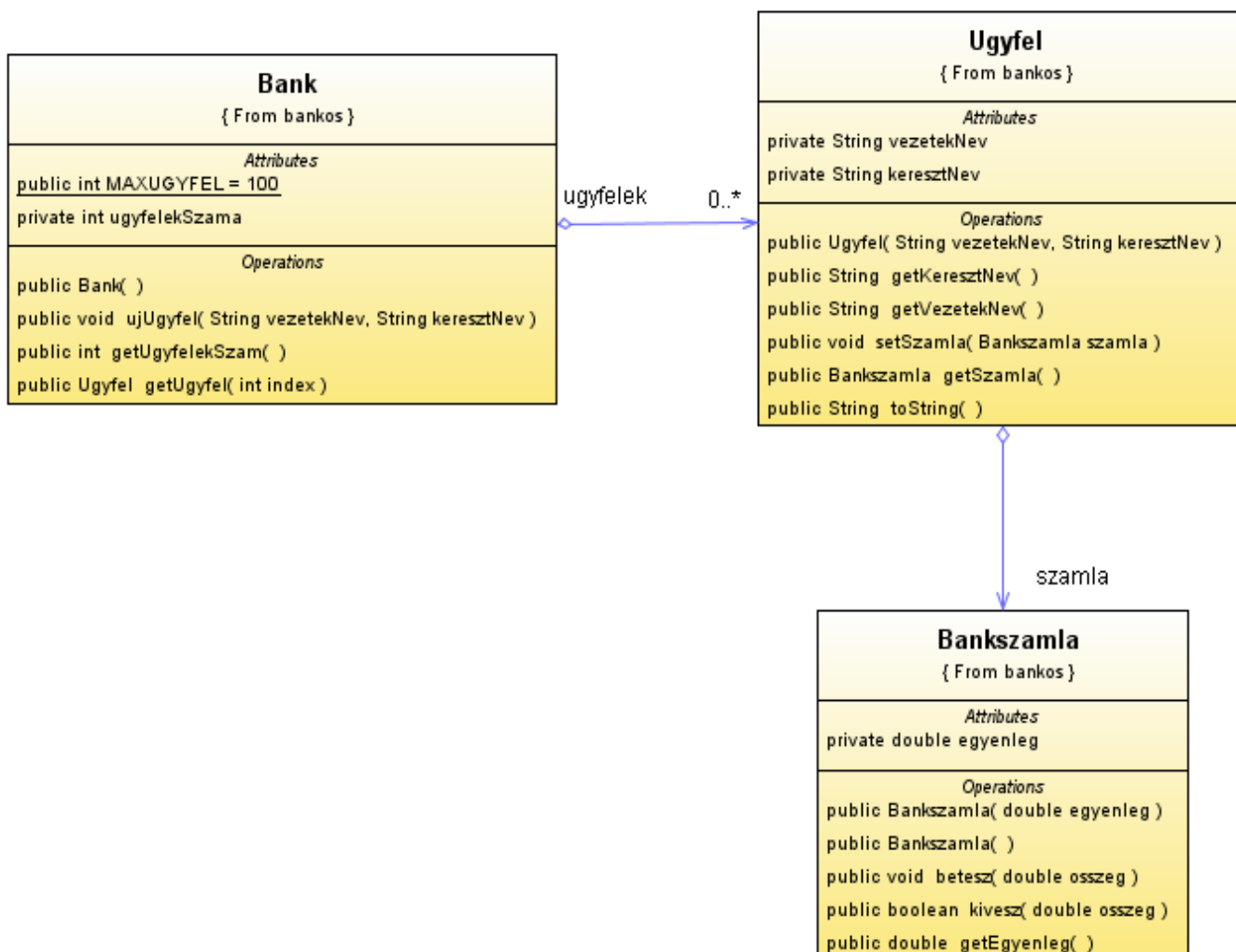
CÉL:

- Műveletek egydimenziós tömbökkel
- Osztályok közötti kapcsolatok: egy-sok kapcsolat megvalósítása

1. Feladat

Készítsen egy másolatot a 3. gyakorlatban készített bankos projektről. Adjon hozzá a projekthez egy **Bank** nevű osztályt.

A **Bank** osztály és az **Ugyfel** osztály között tartalmazási kapcsolat van, mégpedig *egy-sok* kapcsolat, amely megvalósítására tömböt használunk:



OBJEKTUMORIENTÁLT PROGRAMOZÁS 2008.

4. GYAKORLAT

- *egy-sok* kapcsolat megvalósítása

```
public class Ugyfel{
    ...
}

public class Bank{
    private Ugyfel ugyfelek[];
    ...
}
```

- Implementálja a **Bank** osztályt a diagram alapján.
- A **main** metódusban végezze el a következőket:
 - adjon hozzá a bankhoz három ügyfelet, majd mindenik ügyfélhez egy-egy bankszámlát
 - listázza ki a bank ügyfeleit: (vezeteknév keresztnév egyenleg)

Kérdések:

- Hány ügyfele lehet egy banknak?
- Azonosítsa a diagramon az *egy-egy*, illetve *egy-sok* kapcsolatokat !

2. Feladat

Adott egy egész számokat tartalmazó **Matrix** osztály váza. Egészítse ki a metódusokat, majd tesztelje a matrixot. Tesztelje le az osztály minden egyes műveletét.

```
public class Matrix {
    private int elements[][];
    private int rows;
    private int columns;

    public Matrix( int rows, int columns ){
        //
    }

    public int getRows(){
        //
    }

    public int getColumns(){
        //
    }
}
```

OBJEKTUMORIENTÁLT PROGRAMOZÁS 2008.

4. GYAKORLAT

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

public void printMatrix(){
    //
}

public void setElement( int i, int j, int value){
    //
}

public int getElement( int i, int j ){
    //
}

public void randomMatrix(int a, int b){
    //Feltölteni véletlen egész
    //számokkal az [a,b] intervallumból
}
}
```