

## Informatika 4, 1. zárthelyi (2019-10-06)

1	2	3	4	$\Sigma$
---	---	---	---	----------

A feladatok megoldása elfér a feladat alatt, ha külön lapra írjuk, ezt jelöljük egyértelműen!

1. Írjunk olyan függvényt, ami kiszámolja az  $n$ -edik Fibonacci számot. Ne rekurzívan! (5 pont)

```
public static int fibonacci(int n) {
```

```
}
```

2. Írjunk *Complex* nevű osztályt, mely komplex számokat reprezentál. (5 pont)

A lent megadott metódusokat kell megírunk, más publikus adattag vagy metódus ne legyen.

```
public class Complex {  
    float[] t; // kettő hosszú tömb,  
              // ami tárolja a valós és képzetes részt  
    public Complex() {
```

```
    }  
    public Complex(float a, float b) {
```

```
    }  
    public Complex add(Complex other) {
```

```
    }  
    public Complex multiply(Complex other) {
```

```
    }  
    public boolean equals(Complex other) {  
        // adja vissza, hogy a két szám megegyezik-e
```

```
    }  
}
```

A *Complex* osztálynak működésére példa:

```
Complex x = new Complex();  
Complex y = new Complex(3, 4);  
Complex z = x.add(y);  
System.out.println(y.equals(z));
```

3. Válaszoljunk az alábbi kérdésekre! (5 pont)

a) Hogyan kell `main` függvényt írni? Mi a szignatúrája (az első sora)? (1 pont)

b) Mit jelent a `static` kulcsszó a metódusok előtt? (2 pont)

c) Mely metódus(oka)t nevezük *konstruktor*nak? (2 pont)

4. Írjunk olyan függvényt, ami egy számhoz hozzárendeli a prímosztóinak tömbjét.

```
public static int[] primosztok(int n){
```

```
}
```

Egyéb segéd függvényt is írhatunk ha szükséges.

Példa:

$700 \mapsto [2, 5, 7]$