**JAVA OO (objektum orientáltság) ELEMEK (5)**

**Attribútumok és metódusok (9)**

**Immutable objektumok**

def, létrehozás (3)

**JavaBeans objektumok**

def, szabályok(3+1), property, eredeti felhasználás

**Metódusok**

fej részei (5), törzs szabályai (5)

**Változó hosszúságú paraméterlista (varargs)**

leírás (4), szabályok(2)

**Metódus hívások láncolása**

def, feltétel, Builder osztályok (3)

**Metódus túlterhelés**

def, paraméter szignatúra def, használat, sorrend, visszatérési típus problem

**Metódus paraméterek és visszatérési értékek**

paraméterek (3), visszatérési értékek (2)

**Statikus attribútumok és metódusok**

hovatartozás – elérés – használat (2-2-2)

**Értékmásolás szerinti paraméterátadás**

def, primitív és osztálytípus; hatókör, visszahatás(2); memória

## Konstruktorok és inicializátorok (3)

## Default és paraméter nélküli konstruktor (3)

## Konstruktor túlterhelés

## def, par. egyeztetés, chaining, telescoping, private konstruktor, builder osztály

## Inicializátor

## def, típusok (2), mikor futnak le, használat, elhelyezés

## Öröklődés (5)

## Öröklődés

## mikor, 2 típus, nevek, jelölés, egyszeri öröklődés, object ősosztály, isa, hasa, melyiket?

## Konstruktorok és az öröklődés viszonya (4 + 2)

## Object ősosztály

## (3), metódusai, override

## Attribútumok öröklődése

## information hiding, elfedés/felülírás, super, láthatósági módosítók

## Metódusok öröklődése

## elérhetőség, override, override szabályai (6), super, elfedés névazonosság esetén, dinamikus kötés szerinti meghívás, override annotáció, statikus szabályok (3)

## Absztrakt osztályok és interfészek (9)

## Absztrakt osztályok

## def, használat, létrehozás, tartalom, implementáció/ override, referencia típus, változó statikus típus

**Interfészek**

jelentés, javaban(2->def), jellemzők(5) - deklarálás, elnevezés, használat, példányosítás, létrehozás, feladat; implementáló osztály, konstans, többszörös öröklődés, uml jelölés, marker, statikus típus, névütközés

**Interfészekkel kapcsolatos szabályok (8)**

**Dependency Inversion**

kompozíció, d.i. def, hasznosság, implementáció cserélés

## Default interface metódusok

## mikor, jelzés, tartalom, csak hol, öröklődés(3), többszörös öröklődésnél névütközés szabályai, private

## Interfészek és az öröklődés viszonya

## interfészek közötti többszörös, hálós kapcsolat, egy osztály, hogy osztály maradjon.., osztályok közötti öröklődésnél ? öröklődik az interface metódus; névütközés több interface implementációnál

## Statikus interfész metódusok

## csak public és no default, öröklődés?, hívás, törzs, cél, list.of, java9

## Absztrakt metódusok implementálása felsorolásos típusokban

## nem maradhat.., ezért absztrakt metódusait..!, absztrakt metódus vagy kiszervezés interfacebe

## Állapotgép

## def, új állapot ? határozza meg és hány lehet; UML-irányított kapcsolat, enum használat, JVM és fordítók

## 

## Haladó OO elvek (3)

## Polimorfizmus

## def, stat és dinamikus típus, variációk, típuskényszerítés

## Dinamikus kötés

## mit határoz meg, mikor dől el, virtuális metódus, overrideolt metódus láthatósága

## Is-a has-a kapcsolatok

## isa(3-statikus/fordítási, példánya.., instanceof)

## hasa – kompozíció, dinamikus/ futási

## számosság, kötelezőség, irány

## preferálás - >tartalmazási/leszármazotti viszony - >újrafelhasználhatóság

## imlementáció kicserélése setterrel