

Objektum orientált programozás
9. gyakorlat
Beágyazott osztályok, enum

1. Definiáljon egy `math.Polynomial` nevű interfészt.

Metódusai:

- Egy valós számot kap és egy valós számot ad vissza. Szerepe, egy megadott x értékre visszaadni a kiszámolt y -t.
- Paraméter nélküli és egy egészet ad vissza. Szerepe: a polinom fokát visszaadni.

Definiáljon egy `polynomial.LinearPolynomial` osztályt, amely egy elsőfokú egyváltozós polinomot reprezentál (képlete: $y=ax+b$).

Adattagjai: a polinom együtthatói (a és b), valós számok

Konstruktor: két paraméterben megadott értékkel inicializálja az adattagokat

Metódusai:

- Implementálja a `Polynomial` interfész metódusait.
- Definiálja felül a `toString` metódust, amely a polinom képletét adja vissza egy string-ben a konkrét együtthatókkal (Pl.: $y=2x-3$).

Definiáljon egy futtatható osztályt `polynomial.Test` néven.

a) Ebben hozzon létre egy osztályszintű metódust, amely paraméterként megkap egy polinomot (`Polynomial` típusú referenciát) és kiírja a kapott polinom függvénytáblázatát 0 és 1 között 0.1 lépéssel.

b) A `main`-ben hozzon létre 5 lineáris polinomot beolvasott együtthatókkal és tárolja egy tömbbe. Majd írja ki az összes polinom képletét. Ezután az előző metódus segítségével irassa ki az utolsó polinom függvénytáblázatát.

2. Készítsen osztályt új csomagban `BookWithEnum` néven a korábbi könyv osztály leszármazottjaként. Legyen benne `public` beágyazott felsorolás típus `Style` néven a könyv stílusának megadásához a következő konstansokkal: `SCIFI`, `COOK`, `ROMANCE`, `OTHER`.

Adattagja: `Style` típusú stílus adattag

Konstruktor: minden adattagot a paraméterben kapott értékekkel inicializálja

Metódusok:

- Definiálja felül a `toString` metódust úgy, hogy az ősbelihez még fűzze hozzá a stílust.
- Getter metódus a stílus lekérdezésére.

Ugyanebben a csomagban készítsen egy futtatható osztályt, amelyben

a) legyen egy osztályszintű metódus, amely megkap egy könyv tömböt (`Book`) és minden könyv adatát kiírja.

b) legyen egy osztályszintű metódus, amely megkap egy `BookWithEnum` tömböt és egy `Style`-t és visszaadja azon könyvek tömbjét (`BookWithEnum` tömb), amelyek a megadott stílusúak.

c) legyen `main` metódus, amelyben létrehoz egy könyv tömböt az adatok billentyűzetről történő beolvasásával, kiírja a könyvek adatait, majd kiválogatja és kiírja a sci-fi stílusúakat.

Házi feladat

Definiáljon egy `polynomial.QuadraticPolynomial` osztályt, amely egy másodfokú egyváltozós polinomot reprezentál (képlete: $y=ax^2+bx+c$).

Adattagjai: a polinom együtthatói (a , b és c), valós számok

Konstruktor: három paraméterben megadott értékkel inicializálja az adattagokat. Végezzen ellenőrzést: ha $a=0$, akkor elsőfokú a polinom.

Metódusai:

- Implementálja az 1. feladatban definiált Polynomial interfész metódusait.
- Definiálja felül a toString metódust, amely a polinom képletét adja vissza egy string-ben a konkrét együtthatókkal (Pl.: $y=2x^2 - 3x + 5$).
- Kiszámítja a diszkriminánst ($b^2 - 4ac$).
- Visszaadja a megoldás változatokat (2 komplex gyök, 1 valós gyök, 2 valós gyök). A megoldás változatokat Enum-ban tárolja!
- Kiszámítja az egyes megoldás változatokhoz tartozó gyököket. Alkalmazza a metódus túlterhelést!

Ahhoz, hogy két gyököt vissza tudjon adni, definiálni kell a 2 valós gyököt ill. a 2 komplex gyököt leíró beágyazott osztályokat. A beágyazott osztályokban definiálja felül a toString metódust, ami a két gyököt összefűzve adja vissza.

Egy komplex szám leírásához definiálni kell a Complex osztályt (külön csomagban).

```
public final class Complex {
    private final double re;    // the real part
    private final double im;    // the imaginary part

    // create a new object with the given real and imaginary parts
    public Complex(double real, double imag) {
        re = real;
        im = imag;
    }

    // return a string representation of the invoking Complex object
    public String toString() {
        if (im == 0) return re + "";
        if (re == 0) return im + "i";
        if (im < 0) return re + " - " + (-im) + "i";
        return re + " + " + im + "i";
    }

    // return the real or imaginary part
    public double re() { return re; }
    public double im() { return im; }
}
```

A polynomial.Test futtatható osztály main függvényében hozzon létre 5 másodfokú polinomot beolvasott együtthatókkal és tárolja egy tömbbe. Majd írja ki az összes polinom képletét és számítsa ki a megoldásukat.