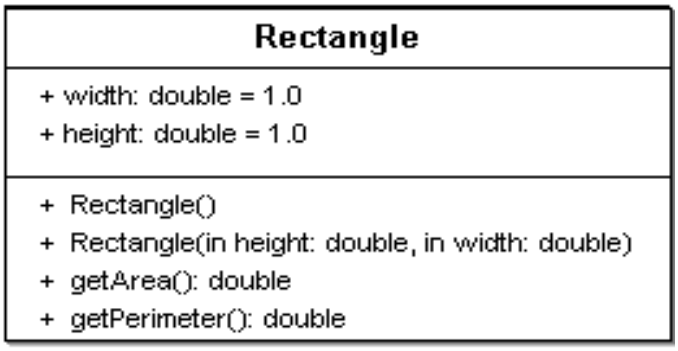
1. (*A **Rectangle** osztály.*) Tervezzen egy **Rectangle** nevű Java osztályt, amely egy téglalapot reprezentál.



   Az osztály tartalmazzon

   - egy width és egy height nevű, double típusú mezőt, amely téglalap szélességét és magasságát adja meg; az alapértelmezés szerint mind a width, mind a height értéke 1 legyen;
   - egy paraméter nélküli konstruktort, amely egy alapértelmezett téglalapot hoz létre;
   - egy konstruktort, amely megadott width és height értékekkel hoz létre egy téglalapot;
   - egy getArea() nevű metódust, amely az aktuális téglalap területének az értékét adja vissza;
   - egy getPerimeter() nevű metódust, amely az aktuális téglalap kerületének az értékét adja vissza!
   - Ne felejtsen el gondoskodni a toString() metódus megírásáról sem!

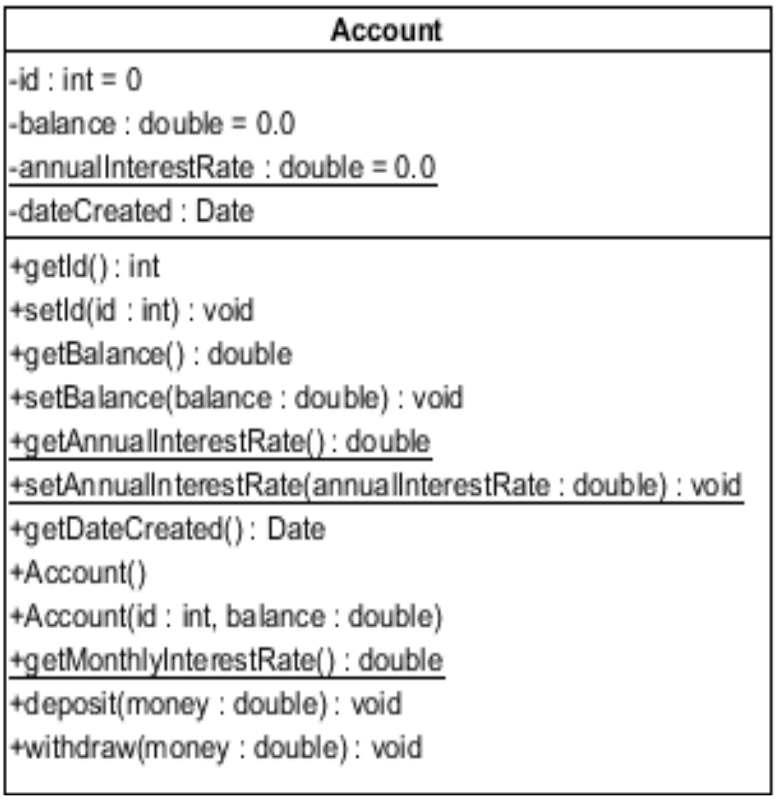
   Implementálja az osztályt! Írjon egy teszt programot, amely létrehoz két Rectangle típusú objektumot — egy 4 szélességű és 40 magasságú, valamint egy 3.5 szélességű és 35.9 magasságú téglalapot!

   Írassa ki az egyes téglalapok szélességét, magasságát, területét és kerületét ebben a sorrendben a standard kimenetre!

2. (*Algebra: másodfokú egyenletek.*) Tervezzen egy QuadraticEquation nevű Java osztályt egy $ax^2 + bx + c = 0$ másodfokú egyenlethez! Az osztály tartalmazzon
    - a, b és c privát adatmezőket, amelyek a három együtthatót reprezentálják;
    - egy konstruktort megadott a, b és c paraméterekkel;
    - a három get metódust a-hoz, b-hez és c-hez;
    - egy getDiscriminant() nevű metódust, amely visszaadja diszkriminánst, amelynek értéke $b^2 - 4ac$.
    - a getRoot1() és getRoot2() nevű metódusokat, amelyek az egyenlet két gyökét adják vissza. Ezek a metódusok csak akkor használhatók, ha a diszkrimináns értéke nem negatív. Ha a diszkrimináns negatív lenne, ezek a metódusok adjanak vissza 0 értéket!
    - Írja meg a toString() metódust is!

    Implementálja az osztályt!

    Írjon egy Test nevű osztályt, ebbe a **main** metódust, amelyben létrehoz 3 tetszőleges QuadraticEquation objektumot, majd megjeleníti a diszkriminánson alapuló eredményt! Ha a diszkrimináns pozitív, írja a két gyököt! Ha a diszkrimináns 0, írja ki a (közös) gyököt! Egyébként írja ki a „The equation has no roots.” üzenetet!



(*Az **Account** osztály.*) Tervezzen egy **Account** nevű Java osztályt, amely tartalmaz

- egy int típusú, id nevű privát adatmezőt, a számla azonosítására (alapértelmezés szerint 0);
- egy double típusú, balance nevű privát adatmezőt, az egyenleg tárolására (alapértelmezés szerint 0);
- egy double típusú, annualInterestRate nevű privát adatmezőt, amely az aktuális kamatlábat tárolja (alapértelmezés szerint 0). Feltételezzük, hogy minden számlához azonos kamatláb tartozik;
- egy Date típusú, dateCreated nevű privát adatmezőt, amely azt a dátumot tárolja, amikor a számlát megnyitották/létrehozták;
- egy paraméter nélküli konstruktort, amely egy alapértelmezett számlát hoz létre;
- egy konstruktort, amely megadott azonosító és egyenleg adatokkal hoz létre egy számlát;
- lekérdező és beállító metódusokat az id, a balance és az annualInterestRate mezőkhöz;
- lekérdező metódust a dateCreated mezőhöz;
- egy getMonthlyInterestRate() metódust, amely a havi kamatláb értékét adja vissza;
- egy withdraw metódust, amely egy megadott (pénz)mennyiséget kivesz a számláról;
- egy deposit metódust, amely egy megadott (pénz)mennyiséget betesz a számlára.

Tervezzen egy **Fan** nevű Java osztályt! Az osztály tartalmazzon

- három nevesített konstanst (SLOW, MEDIUM és FAST) az 1, 2 és 3 értékekkel, amelyek a ventilátor sebességét jelzik;
- egy int típusú, speed nevű privát adatmezőt, amely megadja a ventilátor sebességét (alapértelmezés szerint SLOW);
- egy boolean típusú, on nevű privát adatmezőt, amely jelzi, hogy a ventilátor be van-e kapcsolva (alapértelmezés szerint false);
- egy double típusú, radius nevű privát adatmezőt, amely a ventilátor sugarát adja meg (alapértelmezés szerint 5);
- egy color nevű adatmezőt, amely egy sztringet tartalmaz, ami megadja a ventilátor színét (alapértelmezés szerint blue);
- lekérdező és beállító metódusokat mind a négy adatmezőhöz;
- egy paraméter nélküli konstruktort, amely egy alapértelmezett ventilátort hoz létre;
- egy toString() metódust, amely a ventilátor sztring reprezentációját adja vissza. Ha a ventilátor be van kapcsolva, a metódus a ventilátor sebességét, színét és sugarát adja vissza egy összetett sztringben! Ha a ventilátor nincs bekapcsolva, a metódus a ventilátor színét és sugarát adja vissza azzal a szöveggel együtt, hogy „fan is off”, egy összetett sztringben!
- equals metódus: két ventilátor egyenlő, ha ugyan az a színük és a sugaruk.

Implementálja az osztályt! Írjon egy teszt programot, amely létrehoz két Fan objektumot! Rendeljen maximális sebességet, 10-es sugarat és yellow színt az első objektumhoz, valamint kapcsolja azt be! Rendeljen közepes sebességet, 5-ös sugarat és blue színt a második objektumhoz, ugyanakkor kapcsolja ki azt! Írassa ki az objektumokat a toStringmetódusaik meghívásával!

sAdottak az alábbi stringek:

String s1 = "Welcome to Java";

String s2 = s1;

String s3 = new String("Welcome to Java");

String s4 = "Welcome to Java";

String s5="semmi";

Mi lesz a következő kifejezések eredmény(egymástól függetlenül)?

System.out.println(s1 == s2);

System.out.println(s2 == s3);

System.out.println(s1 == s4);

System.out.println(s1.equals(s2));

System.out.println(s2.equals(s3));

System.out.println(s1.compareTo(s2));

System.out.println(s2.compareTo(s3));

System.out.println(s2.compareTo(s5));

System.out.println(s1.charAt(0));

System.out.println(s1.indexOf('j'));

System.out.println(s1.indexOf("to"));

System.out.println(s1.lastIndexOf('a'));

System.out.println(s1.lastIndexOf("o", 15));

System.out.println(s1.length());

System.out.println(s1.substring(5));

System.out.println(s1.substring(5, 11));

System.out.println(s1.startsWith("Wel"));

System.out.println(s1.endsWith("Java"));

System.out.println(s1.toLowerCase());

System.out.println(s1.toUpperCase());

System.out.println(" Welcome ".trim());

System.out.println(s1.replace('o', 'T'));

System.out.println(s1.replaceAll("o", "T"));

System.out.println(s1.replaceFirst("o", "T"));

char[] str = s1.toCharArray();