Java BGN

Feladatok

Alapok

1. Készíts egy programot, mely kiírja a konzolra hogy „Hello world”! Fordítsd le és futtasd!
2. Készítsd el a „Hello world” programot IDE használatával! Használj Maven build eszközt!

Adatok

3. Készíts egy `Multiplier` osztályt! A `main()` metódusban deklarálj két egész számot, majd írd ki a szorzatukat az alábbi formában: 5 * 4 = 20!
4. Hozz létre egy `Client` osztályt, melynek három attribútuma van: név (`name`), születési év (`year`) és cím (`address`). Típusaik rendre `String`, `int` és `String`. Hozz létre egy `main()` metódust a `ClientMain` osztályba, amelyben kipróbálod a `Client` osztály működését! Példányosítani kell egy objektumot a `Client` osztály alapján, majd adj értéket az attribútumoknak! Ellenőrzésképp írd ki minden attribútumának értékét a konzolra!
5. Módosítsd a `Multiplier` osztályt úgy, hogy a két számot a felhasználótól kéred be!
6. Módosítsd a `ClientMain` osztályt úgy, hogy a `Client` adatait a felhasználótól kéred be!

Vezérlési szerkezetek

7. Készíts egy programot, amely a felhasználótól bekért két szám közül a nagyobbat írja ki!
8. Készíts egy programot, amely a felhasználótól bekéri egy hónap sorszámát, és kiírja a hónap nevét! Amennyiben olyan sorszámot kap, amely nem 1 és 12 közötti, jelezze a hibát a felhasználó felé!
9. Módosítsd az előző programot úgy, hogy a hónap sorszáma alapján az évszakot írja ki!
10. Készíts egy programot, amely beolvas egy betűt és egy számot a felhasználótól, és a betűt annyiszor írja ki egymás után, amennyi a szám!
11. Készíts egy programot, amely csillagokból kirajzol egy téglalapot. A téglalap méreteit a felhasználótól kapja meg! (Profiknak: Csak 1 és 20 közötti értéket fogadj el szélességnek és magasságnak!)

    Pl. 3 magas és 5 széles esetén:
    *****
    *****
    *****

Tömb, lista

12. A `Numbers` osztály `main()` metódusában hozz létre egy 5 elemű egész számok tárolására alkalmas tömböt! Írd ki a tömb 2. elemét! Mi van benne?
    a. Töltsd fel a tömböt tetszőleges számokkal!
    b. Listázd ki a képernyőre az összes elemet!

13. Készíts egy lottószelvény kitöltő programot! A felhasználó 5 darab egymástól különböző 1-90 közötti számra tippelhet. Tárold el ezeket egy tömbben, majd listázd ki a tippeket a képernyőre!

14. Módosítsd a lottószelvény kitöltő programot úgy, hogy a megtippelt számokat növekvő sorrendben írja ki!

15. Készítsd el a lottószelvény kitöltő programot `ArrayList` hazsnálatával! Módosítsd úgy a programot, hogy mind 5-ös, min 6-os lottó tippeket lehessen megadni! (A 6-os lottón 1-45 közötti számokra lehet tippelni.) (Profiknak: Hogyan lehet a listát berendezni?)

Algoritmizálási alapok

16. Készíts egy programot, amely egy egész számokból álló lista elemeit adja össze és átlagolja!

17. Egészítsd ki az előző programot, hogy megszámlálja, hány páros szám van az elemek között!

18. Egészítsd ki az előző programot úgy, hogy az eldöntse, hogy van-e a számok között 100-nál nagyobb, illetve hogy mind nagyobb-e, mint 50!

19. Egészítsd ki az előző programot, hogy megkeresse az első negatív számot! Ha nincs benne negatív, akkor írja ki, hogy „Nincs a számok között egyetlen negatív sem.”!

20. Készíts programot, amely szavak listájából írja ki a legrövidebb és a leghosszabb szót!

21. Készíts programot, amely szavak listájából kiválogatja az összes „A” betűvel kezdődőt, és alfabetikus sorrendben kiírja őket egymás után a konzolra!

Osztályok

22. Készíts egy `Car` osztályt!
    Attribútumai:
    `registrationNumber`: String
    `positionX`: int
    `positionY`: int
    `speed`: int
    Minden részfeladat után teszteld az osztály működését a `CarMain` osztály `main()` metódusában!

    a. Egészítsd ki konstruktorral, amelyben minden attribútumot megkap paraméterként!

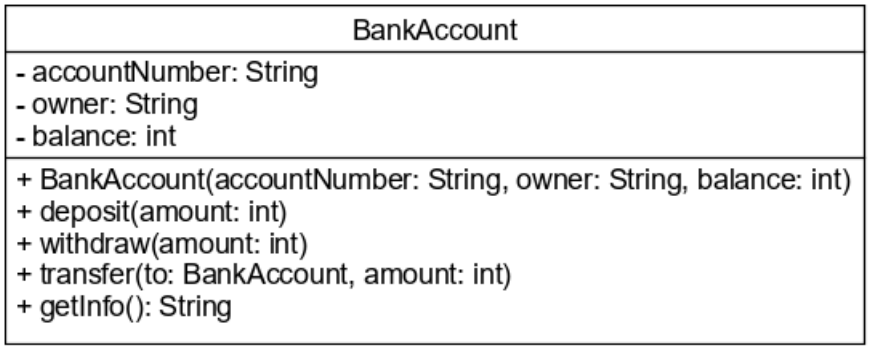
    b. Készíts olyan konstruktort, amely csak a rendszámot kapja meg! Minden más attribútum legyen 0!

    c. Módosítsd az előbb megkapott konstruktort, hogy az autó kezdő pozíciója (100, 100) legyen!

    d. Egészítsd ki a `Car` osztályt getterekkel, setterekkel és az alábbi metódusokkal!
    `accelerate()`: egy egységgel növeli az autó sebességét
    `move(deltaX: int, deltaY: int)`: megváltoztatja az autó pozícióját a paraméterként kapott értékekkel

    e. Egészítsd ki az osztályt úgy, hogy ha az `accelerate()` metódus kap paramétert, akkor annyival változtatja a sebességet!

23. Az UML diagram és a leírás alapján készítsd el a `BankAccount` osztályt!



Számlanyitáshoz mindhárom attribútum értékét meg kell adni. A számlára lehet befizetni (`deposit()`), lehet róla pénzt kivenni (`withdraw()`), illetve másik számlára át lehet utalni összeget (`transfer()`). Ez utóbbi esetben a számla saját egyenlege csökken, de a másik számla egyenlegére jóváírás történik.

A `getInfo()` metódus a számla adatait az alábbi formában adja vissza Stringként:

```
10073217-12000098-67341590
Tóth Kálmán
103400 Ft
```

Sortörést a szövegbe a `\n` karakterrel tudsz elhelyezni.

Készíts egy `Bank` osztályt, amely `main()` metódusában létrehozol két bankszámlát! Próbáld ki az összes elkészített metódust, hogy jól működik-e! Átutalásnál ellenőrizd mindkét számla új egyenlegét!