

### Alapok

1. Készíts egy programot, mely kiírja a konzolra hogy „Hello world”!
2. Módosítsd az előbb elkészített programot, hogy a futtatáskor paraméterként átadott felhasználónévvel üdvözlje a felhasználót!

### Adatok

3. Készíts egy `Multiplier` osztályt! A `main()` metódusban kérj be a felhasználótól 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 kipróbáld a `Client` osztály működését. Példányosítani kell egy objektumot a `Client` osztály alapján, majd kérd be az attribútumok értékét a felhasználótól! Ellenőrzésképp írd ki minden attribútumának értékét a konzolra!

### Vezérlési szerkezetek

5. Készíts egy programot, amely a paraméterként átadott két szám közül a nagyobbát írja ki! Ha nincs elég paraméter akkor jelezze a hibát! (A paraméterek típusával még nem kell foglalkoznod.)
6. Készíts egy programot, amely paraméterként egy hónap sorszámát várja, és kiírja a hónap nevét! Amennyiben olyan sorszámot kap, amely nem 1 és 12 közötti, jelezze a hibát a felhasználó felé!
7. Módosítsd az előző programot úgy, hogy a hónap sorszáma alapján az évszakot írja ki!
8. Készíts egy programot, amely paraméterként egy betűt és egy számot kap, és a betűt annyiszor írja ki egymás után, amennyi a szám!
9. Készíts egy programot, amely csillagokból kirajzol egy téglalapot. A téglalap méreteit paraméterként kapja meg!

Pl: Teglalap 3 5 esetén:

```
*****
*****
*****
```

### Algoritmizálási alapok

10. Készíts egy programot, amely egy egész számokból álló tömb elemeit adja össze és átlagolja!
11. Egészítsd ki az előző programot, hogy megszámolja, hány páros szám van az elemek között!

12. 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!
13. 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.”!
14. Készíts programot, amely szavak tömbjéből írja ki a legrövidebb és a leghosszabb szót!
15. Készíts programot, amely szavak tömbjéből kiválogatja az összes „A” betűvel kezdődőt, és alfabetikus sorrendben kiírja őket egymás után a konzolra!
16. Rendezd be a 8. feladatban létrehozott tömböt!

## Osztályok

17. Készíts egy `Car` osztályt egyszerű szövegszerkesztővel!  
Attribútumai:  
`registrationNumber: String`  
`positionX: int`  
`positionY: int`  
`speed: int`  
Fordítsd le konzol ablakból, majd fordítsd vissza!
18. Importáld IDE-be a `Car` osztályt és egészítsd ki konstruktorral, amelyben minden attribútumot megkap paraméterként!
19. Készíts olyan konstruktort, amely csak a rendszámot kapja meg! Minden más attribútum legyen 0!
20. Módosítsd az előbb megkapott konstruktort, hogy az autó kezdő pozíciója (100, 100) legyen!
21. 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
22. 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!

BankAccount
- accountNumber: String - owner: String - balance: int
+ BankAccount(accountNumber: String, owner: String, balance: int) + deposit(amount: int) + withdraw(amount: int) + transfer(to: BankAccount, amount: int) + getInfo(): String

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!