

# Angol nyelvű szakmai kifejezések

---

**perzisztencia:** olyan rétege egy alkalmazásnak, ami az alkalmazás által kezelt adatok tartós tárolásáért felel

- legtöbbször adatbázishoz kapcsolódik, és az adatbázissal kommunikál

**DAO:** Data Access Object

**DAO-layer:** adatelérésért felelős réteg

**reszponzív:** (responsive): az alkalmazás felülete a különböző megjelenítési felületeken is jól kezelhető módon jelenik meg

- adott méretekre optimalizált verziói vannak (desktop, tablet, mobil, stb)

**fluid:** az alkalmazás felülete a különböző megjelenítési felületeken is jól kezelhető módon jelenik meg

- nincsenek kategóriák, hanem bármilyen méretben, alakban jól kell, hogy megjelenjen

**karakterkiosztások/ character sets** ([Joel Spolsky cikke](#))

- ASCII: általános 1 byte-on tárolható karakterek ([link a táblázathoz](#))
- UNICODE: kiegészíti az alapkészletet, nyelv/terület függő karaktereket ad hozzá laponként
- UTF-8: változó hosszúságú Unicode karakterkódolási eljárás (van még UTF-16, UTF-32, stb)
- törekedjünk az UTF-8 használatára

**POJO:** Plain Old Java Object

**script:** forráskódokat tartalmazó fájl, ami interpreteres nyelven írt

**interpreter:** (értelmező) soronként értelmezi a forráskódot, és futtatja (pl.: Python, scriptnyelvek)

**compiler:** fordítóprogram - forráskódból futtatható állományt állít elő (pl.: C, C#, stb)

**JVM:** Java Virtual Machine ([mottó](#))

- a Java fordítóprogram (javac) Java forráskódot JVM-en futtatható bytekódra fordít le, majd a JVM-nek átadva az platformfüggetlenül tudja futtatni

**user interface/GUI:** felhasználói felület, grafikus felhasználói felület

**wireframe:** drótváz, egyszerű, általában designelemek nélküli felhasználói felületterv

**algorithm:** algoritmus, utasítások sorozata

**application:** alkalmazás (desktop, web, mobile - asztali, webes, vagy mobilappok)

**architecture:** architektúra - szoftver komponenseinek általános felépítése, részei közötti kommunikáció

**database:** adatbázis

**array:** tömb - adatszerkezet, véges elemszámú adatot tárol

**assembler:** (összeállító): assembly kódból gépi kódot állít elő

**Assembly:** low-level / alacsony szintű programozási nyelv

**inline style:** HTML és CSS nyelveket keverjük, HTML-be oda nem illő CSS formázásokat helyezünk

**thread:** futási szál

**asynchronous:** aszinkron - kizárólag több szált kezelni képes környezetben lehetséges

**binary:** bináris, kettős, kétalakú, kétállapotú

**bit:** adattárolás legkisebb egysége, kétállapotú, 0-t vagy 1-egy tárol

**branch:** (ág) verziókezelő rendszereknél a kód egy más verzióban tárolt állapota

**bug:** (bogár): hiba, szoftverben található hiba

**CPU:** (Central Processing Unit) központi feldolgozó/vezérlő egység - gépünk "agya"

**server:** szerver, kiszolgáló alkalmazás, végpontokon szolgáltatásokat szolgáltat

**client:** kliens, szerverhez képes kapcsolódni, azzal interakcióba tud lépni (adatok küld neki, stb) a végpontokon keresztül, egyszerre több kliens is kapcsolódhat általában egy szerverhez

**MVC:** Model-View-Controller - szoftverarchitektúra

**code review:** az a tevékenység, amikor egy fejlesztő egy másik fejlesztő által írt kódot átnéz, ellenőriz, véleményez

**computer:** számítógép

**documentation:** dokumentáció, leírás

- felhasználói, fejlesztői, telepítési, üzemeltetési, stb.

**relation:** reláció, kapcsolat, táblák és a köztük megfogalmazható kapcsolat

- database: olyan adatbázis, amiben az adatokat tároló egységek(táblák) között lehet kapcsolatokat kezelni

**NOSQL database:** az adatokat tároló egységek(gyűjtemény, lista, dokumentum, stb) között **nem** lehet kapcsolatokat kezelni

- előnye a relációs adatbázisokkal szemben, hogy a adatok között általában sokkal gyorsabban tud keresni, illetve nem kötött típusú adatokat is tud tárolni

**OOP:** Object-Oriented Programming, olyan paradigma, objektumorientált programozás, ahol létezik az osztály és az osztályokhoz tartozó egyéb nyelvi elemek (pl.: Java)

**refactoring:** refaktorálás - (általában) működő kód újrastrukturálására, újraírása, kódminőség, kódhatékonyság érdekében - ugyanazt a funkcionalitást kell, hogy biztosítsa a kód a refaktorálás után is

Mindig hagyd a kódot szebben, mint ahogy kaptad!

**waterfall model:** vízesés modell - szoftverfejlesztési módszertan

**spagetti code:** nem jól strukturált, nem jól szervezett, átláthatatlan, nehezen értelmezhető kód(részlet) - tipikusan refaktorálásra szorul

**smoke testing:** kódmódosítások, hibajavítások után is elvárt, hogy a rendszer összességében, még mindig helyesen működjön és minden korábban megfogalmazott tesztet lefusszon

**SDK:** (software development kit) - előre definiált kódok, struktúrák, (osztályok), amit az adott nyelven történő fejlesztéshez elengedhetetlenek csomagba foglalva

**JDK:** (Java Development Kit) - ennek része a JRE - fejlesztéshez szükséges, és mivel része a JRE így futtatni is tud

**JRE:** (Java Runtime Environment) - fejleszteni nem lehet vele, kizárólag előre lefordított Java kódokat tud futtatni

**pseudocode:** olyan nyelv, ami nem egy programozási nyelv, de programot tudsz vele leírni, tervezéshez (algoritmusok tervezéséhez) használatos, nem lehet futtatni, előre definiált "szabályzat" amiben meg van határozva, hogy adott nyelvi elemeknek hogyan kell megjelennie

**boilerplate:** olyan kód, ami a kódbázisban sokszor ugyanolyan szerkezetben, ugyanolyan mintával megjelenik, cél, hogy ezektől megszabaduljunk, (mert amúgy időt vesznek el) és ezzel kódDuplikációt csökkentünk (pl.: refaktorálással, segédcsomagokkal)

tipikusan boilerplate kódok : getter, setter, stb

**random:** véletlenszerű, pszeudorandom/valós véletlenszerű

**class:** osztály, azonosos tulajdonságokkal rendelkezik, (tervrajz az objektumok létrehozásához)

**package:** csomag, valamilyen logika alapján összetartozó osztályok összesége (konkrétan mappaként valósul meg a fájlrendszerben)

**getter:** osztály (private, protected, stb) változóinak értékeinek lekérésére, hogy az osztályon kívül is elérhető legyen az értéke

**setter:** osztály (private, protected, stb) változóinak értékeinek beállítására, vagy módosítására, hogy az osztályon kívülről is változtathatók legyenek

**hashCode:**

- hashCode metódus: az objektumnak egyedi azonosítót generál
- hash: egyedi és egyértelmű azonosító

**markdown:** leíró nyelv, megkönnyíti a dokumentumok formázását, széleskörben használt fejlesztési világában

**WYSIWYG editor:** what you see is what you get szerkesztő

**ArrayList:** egy olyan osztály, ami megvalósítja a List interface-t, azaz egy adatszerkezet, amiben egyforma típusú elemeket tárolhatunk, és fontos az elemek sorrendje

**framework:** keretrendszer, ami előre elkészített, újrafelhasználható elemeket tartalmaz, adott funkcionalitás fejlesztését megkönnyíti/ fejlesztési időt lecsökkenti, pl.: Bootstrap (CSS), .NET (C#), Spring (Java)

**recursive:** rekurzív, önmagát hívja meg pl.: algoritmus)

**entity:** entitás, szoftver (azon belül egy adott osztályt) és adatbázis közötti kapcsolatot teremti meg ((mapping(leképzés) db - software))

((((HBR))))): HáromBetűs Rövidítések

```
//  
// Dear maintainer:  
//  
// Once you are done trying to 'optimize' this routine,  
// and have realized what a terrible mistake that was,  
// please increment the following counter as a warning  
// to the next guy:  
//  
// total_hours_wasted_here = 42  
//
```