

# Szoftverfejlesztés (INBPM0420E) – az 1. zárthelyi dolgozat témái

2023. március 21.

## 1. Szoftverfejlesztés

- Mi a szoftverfejlesztés?
- Fogalmak: programkönyvtár, alkalmazásprogramozási interfész (API), keretrendszer

## 2. Java technológiai áttekintés

- Java virtuális gép fogalma
- Platform fogalma
- Java platformok: Java SE, Java EE/Jakarta EE
- Java Runtime Environment (JRE), Java Development Kit (JDK)
- OpenJDK és Oracle JDK, kapcsolatuk
- A Java SE aktuális kiadási modellje

## 3. Rendszerépítés

- Mi a rendszerépítés?
- *Build* automatizálás, automatikus *build* eszközök, *build* eszközök funkciói
- Folyamatos integráció fogalma
- DevOps fogalma

## 4. Apache Maven

- Mi az Apache Maven?
- A szabványos könyvtárszerkezet
- Parancssori használat
- Alafogalmak: termék, Project Object Model (POM), szuper POM, effektív POM, Maven koordináták, bővítmények, tárolók (távoli és lokális)
- Életciklusok, életciklus fázisok
- A default életciklus és főbb fázisai (`generate-sources`, `process-resources`, `compile`, `package`, `install`, `deploy`)
- Tulajdonságok, hivatkozás tulajdonságokra
- Függőségek megadása
- Verziószámok
- Verzió követelmények
- Transzitiv függőségek, tranzitiv függőségek kizárása
- Öröklés
- Profilok, profil aktiválás
- Bővítmények használata

## 5. Szoftver sebezhetőségek

- Sebezhetőség fogalma
- Sebezhetőségi adatázis fogalma
- A CVE és NVD

## 6. Szemantikus verziószámozás

- verziópokol/függőségi pokol
- Mi a szemantikus verziószámozás?
- Normál verziószámok, verziószámok növelése

## 7. Annotációk a Java programozási nyelvben

- Annotáció fogalma
- Lehetséges felhasználások
- Annotációk szintaxisa
- Annotációk fajtái
- Hol alkalmazható annotáció?
- Előre definiált annotáció interfészek: `@Deprecated`, `@SuppressWarnings`, `@Override`, `@FunctionalInterface`
- Meta-annotációk: `@Documented`, `@Inherited`, `@Repeatable`, `@Retention`, `@Target`
- Annotáció interfészek deklarálása

## 8. A Java SE/JDK új lehetőségei

- Előzetes lehetőségek
- Lokális változó típus kikövetkeztetés
- `switch` kifejezések
- Szövegblokkok
- Mintaillesztés az `instanceof` operátorhoz
- Rekord osztályok
- Lezárt osztályok

## 9. A Java haladó szintű lehetőségei

- Nem absztrakt (alapértelmezett, statikus, privát) interfész metódusok
- `java.util.Optional`
- Funkcionális interfészek
- Beépített funkcionális interfészek: `Consumer`, `Function`, `Predicate`, `Supplier`
- Lambda kifejezések
- Metódus referenciák
- Streamek
- Stream műveletek: köztes és terminális műveletek, állapotmentes és állapotörző műveletek

- Streamek: viselkedési paraméterek kívánatos jellemzői (interferencia-mentesség, állapotmentesség)
- Stream csővezetékek, műveletek kiértékelése, csővezeték végrehajtás
- Streamek: redukciós műveletek (`reduce`, `collect`)
- Streamek: párhuzamosság

## 10. Git

- Tároló létrehozása (`git init`)
- Állományok figyelmen kívül hagyása `.gitignore` állománnyal
- A munkakönyvtár állapotának vizsgálata (`git status`), nem véglegesített módosítások megjelenítése (`git diff`)
- Változások véglegesítése (`git commit`), állományok hozzáadása tárolóhoz (`git add`), állományok törlése tárolóból (`git rm`)
- Verziótörténet megjelenítése (`git log`)
- Ágak használata: ág létrehozása, váltás ágak között, ágak összefésülése, ágak törlése
- Módosítások „visszacsinálása”: az utolsó *commit* üzenet megváltoztatása, nem véglegesített módosítások elvetése, véglegesített módosítások „visszacsinálása”