Hova tovább Android? – OpenJDK és további várható újdonságok

peter.ekler@aut.bme.hu

BME AUT
Automatizálási és
Alkalmazott
Informatikai Tanszék

AutSoft
BME

A félév felépítése

- Java nyelv kialakulása
- Szintaktikai alapok
- Java környezetek
  - Java ME
  - Java SE
  - Java EE
- Java APIk: UI, hálózatkezelés, perzisztencia, stb.
- Java Web

- Ütemezés:
  - ZH1:
    - 7. hét
  - ZH2:
    - 12. hét
  - NagyHF specifikáció:
    - 8. hét
  - NagyHF leadás:
    - 14. hét

Miről beszéljünk?

OpenJDK

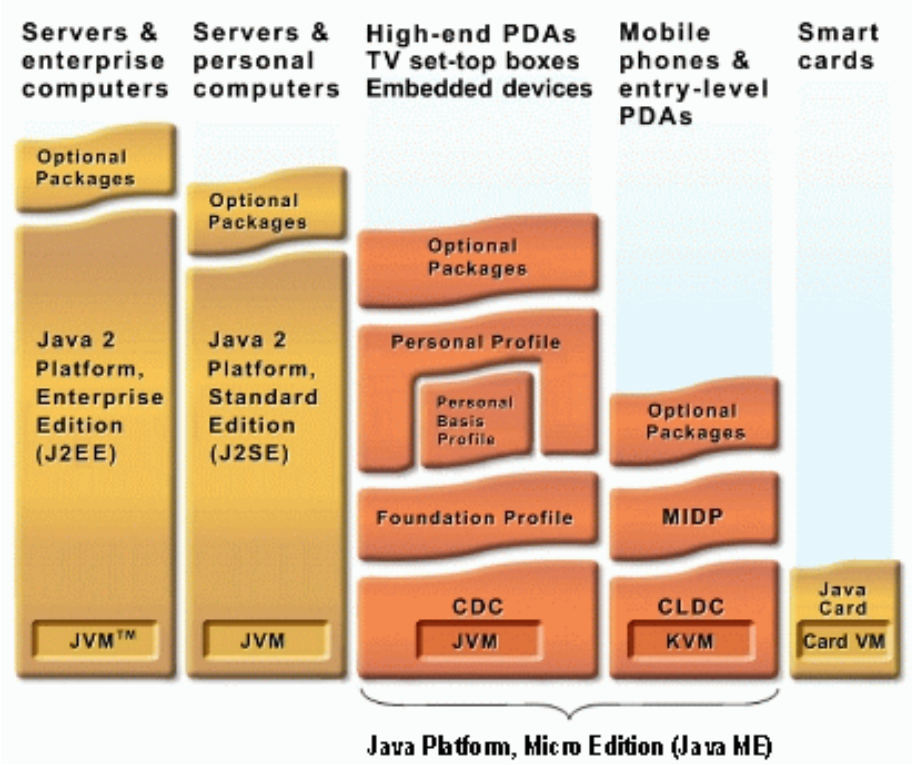
OpenJDK

Java fejlődéstörténet - röviden

- 1996: Development Kit 1 (Oak)
  - 8 package; 212 osztály
- ...
- 2006: Java 6 (Mustang)
  - 203 package; 3793 osztály
- 2011: Java 7 (Dolphin)
  - 209 packages; 4024 osztály
- 2014: Java 8
  - 217 package; 4240 osztály
- ~2017 március: Java 9

Java újdonságok

- Nyelvi újdonságok
  - Be kell hozni a lemaradást
- Fejlődő API
- Fejlettebb virtuális gép
- Fejlődő GC
- Fordítás optimalizálása
- Technológiai újítások:
  - Beágyazott, web, asztali

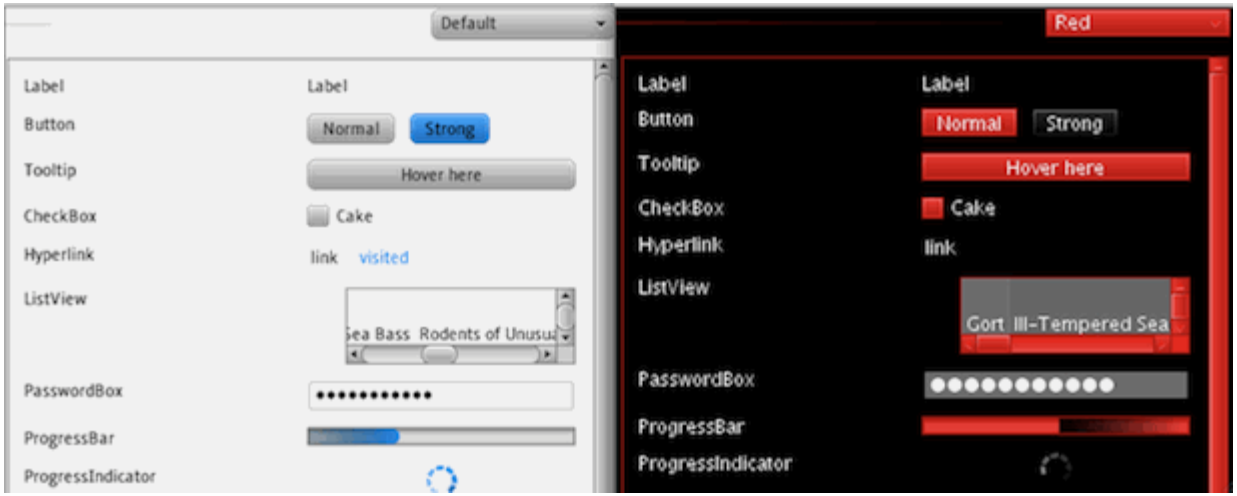


JavaFX áttekintés

- Java SE része, Win/Linux/Mac támogatás
- Teljesen megújult UI API
- FXML alapú felület leírás
- Grafikus gyorsító támogatás
- Multimédia, animációk és effektek
- Web render támogatás
- Gyors eseménykezelés
- Swing „kompatibilitás”

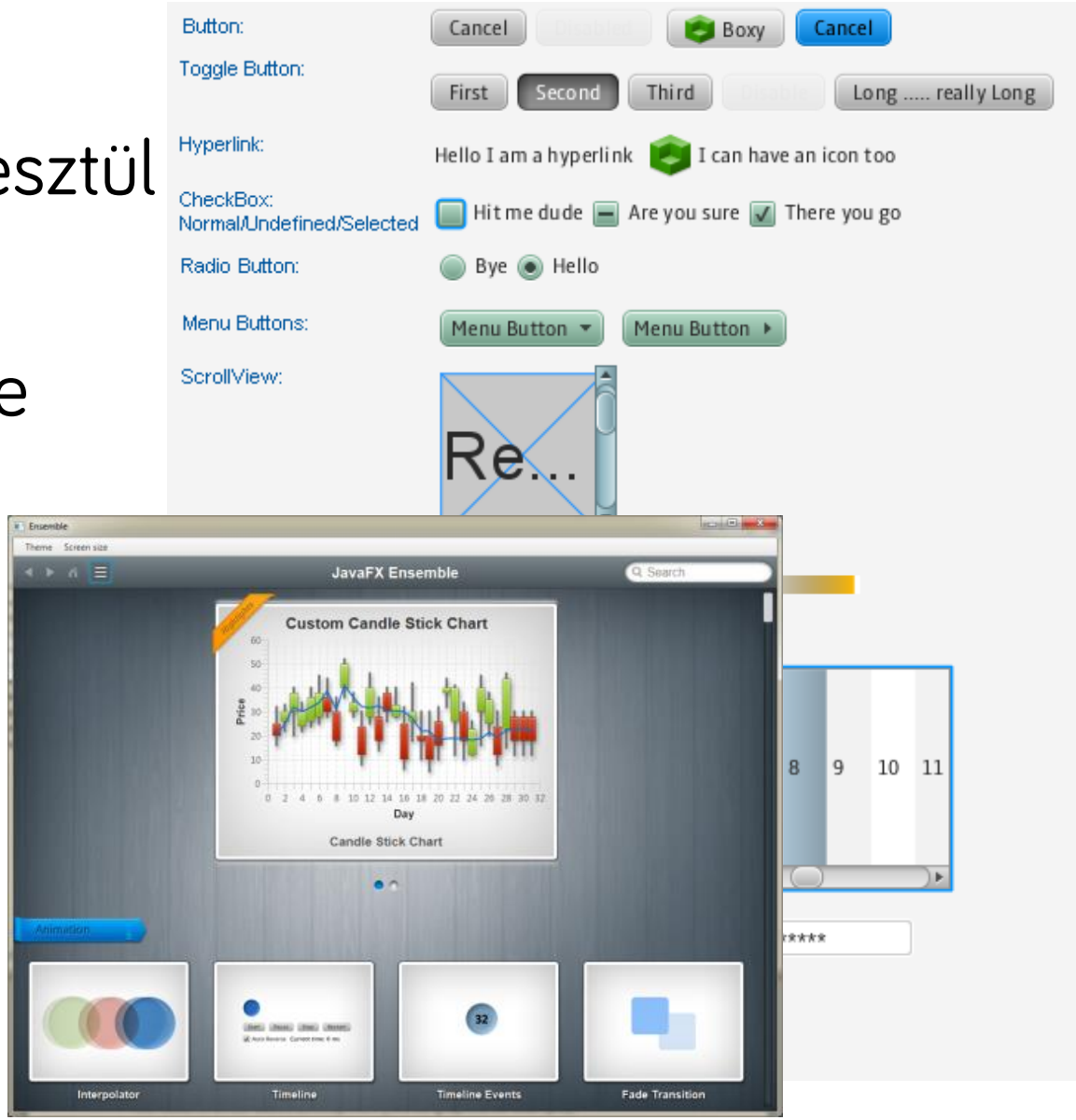
CSS támogatás

- Megjelenítés megváltoztatása a forráskód változtatása nélkül
- Bármelyik Scene Graph Node-on alkalmazható és akár futási időben is hozzárendelhető
- W3C CSS 2.1 alapú



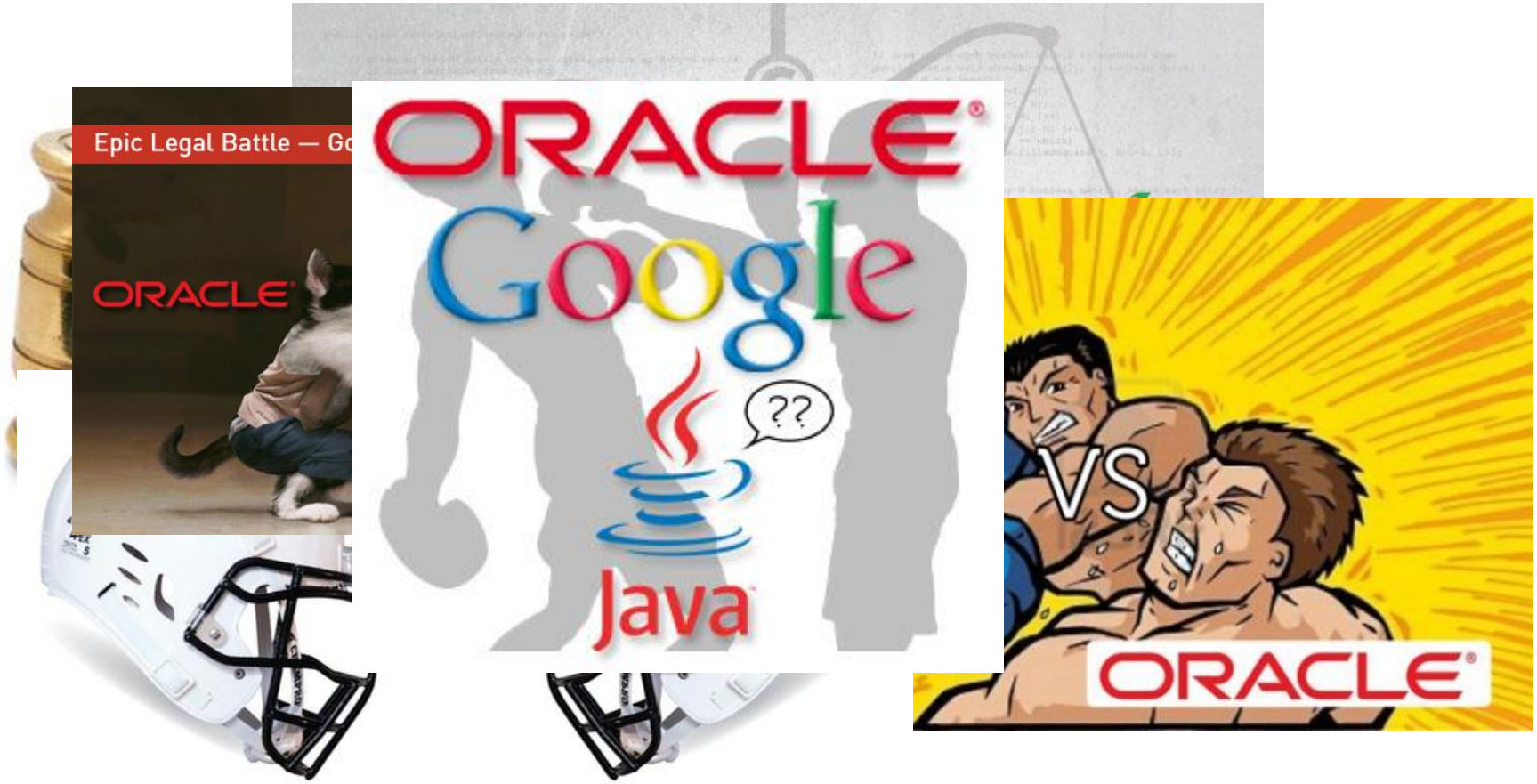
UI vezérlők

- JavaFX API-n keresztül elérhetők
- Scene Graph Node elemek
- CSS-el testreszabható megjelenítés
- Animációk és effektek



Oracle vs. Google

- Mi a pereskedés tárgya?



Háttér

- 2007: **Android** megjelenése
  - Java a SUN tulajdonában volt
  - Nyitottak voltak az együttműködésre, de nincs szerződés
  - **Apache Harmony** Java implementáció (2011-ben leállt a fejlesztése)
- ~2010: Oracle felvásárolja a SUN-t
  - Nem jutottak egyezségre
  - Folyamatos pereskedés
- 2015-16: Áttérés **OpenJDK-ra**
  - Érdekesség: OpenJDK is Oracle fejlesztés

Milyen Java-t is támogat most az Android?

- **Java 7** a 19-es Build Tool-tól
  - 2011-ben jelent meg
  - 2012-13-tól kezdték el támogatni Androidon
- **~Java 8** bizonyos részei API level-től függően
  - 2014-ben jelent meg
  - 2016-tól kezd részben támogatott lenni Androidon

Java 7 - emlékeztető

- String alapú switch
- Bináris formátumok és ‚_'-es elválasztás számokban:

```
int bitfield = 0b10101001;
int value = 100_200;
```

- Multi-catch és pontosabb kivétel tovább dobás

```
try {
    …
}catch (ConnectionPendingException | IOException  ex) {
    ex.printStackTrace();
}
```

- Improved type inference for generic instance creation (diamond)

```
MyClass<Integer> myObject = new MyClass<>("");
```

- try-with-resources statement

Try-with-resources statement

```
try (
    Socket socket = new Socket("192.168.1.130", 5040);
    InputStream is = socket.getInputStream();
) {
    …
} catch (Exception e) {
    e.printStackTrace();
}
```

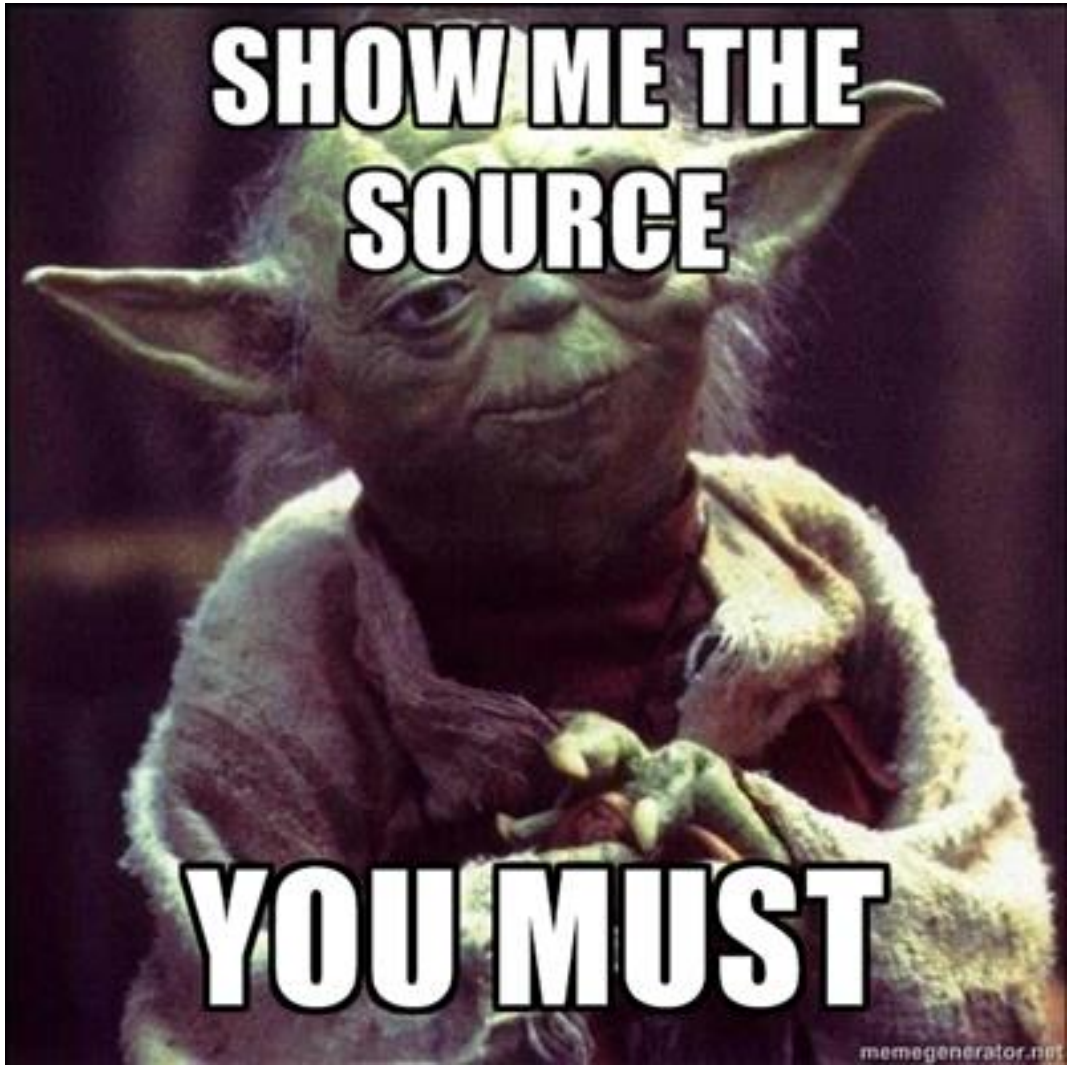
- *AutoCloseable-t* implementáló objektumok *close()* metódusa automatikusan meghívódik
  - Készíthetünk saját implementációkat is
- Ettől függetlenül lehet catch és finally ág
  - Ezek az erőforrás bezárás hívás után hajtódnak végre

Megérkett: Android Nougat (7.0 – 24)

- Biztonság fejlesztése
- „Láthatatlan” frissítések
- Multi-window multitasking
- Fejlettebb értesítések
- Gyorsabb és energiatakarékosabb (Project Doze)
- Kevesebb mobil adatforgalom
- Hozzáférés vezérlése könyvtáranként
- Fejlesztőknek: adott Java 8 nyelvi elemek

Főbb Java 8 újdonságok - Android

- Alapértelmezett és statikus metódusok Interface-ben (API 24-től)
- Lambda kifejezések (API 9-től felfele)
- Ismétlődő/azonos annotációk
- Metódus referenciák (API 23-tól és annál kisebb is)
- Típus annotációk (API 23-tól és annál kisebb is)
  - Fordítási időben érhetők csak el
- Új nyelvi API-k
- Utility APIk:
  - java.util.function
  - java.util.stream


SHOW ME THE SOURCE
YOU MUST
memegenerator.net

Alapértelmezett és statikus metódusok Interface-ben

```
public interface ResponseHandler {
    void handleOk(byte[] data);
    void handleEmpty();

    static String convertHttpStatusToString() {
        ...
    }

    default void handleError(String status) {
        Log.d("TAG_ERROR", status);
    }
}
```

Lambda kifejezések

```
btnDemo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(MainActivity.this, "Demo",
          Toast.LENGTH_SHORT).show();
    }
});
```

Helyett:

```
btnDemo.setOnClickListener(v -> Toast.makeText(
  MainActivity.this,
  "Demo", Toast.LENGTH_SHORT).show());
```

„Lambda expressions let you express instances of single-method classes more compactly.”

Ismétlődő/azonos annotációk

```
@Schedule(dayOfMonth="last")
@Schedule(dayOfWeek="Fri", hour="23")
public void doPeriodicCleanup() { ... }
```

```
@Repeatable(Schedules.class)
public @interface Schedule {
    String dayOfMonth() default "first";
    String dayOfWeek() default "Mon";
    int hour() default 12;
}
```

```
public @interface Schedules {
    Schedule[] value();
}
```

Metódus referenciák

Típus	Példa
Referencia statikus metódusra	`ContainingClass::staticMethodName`
Referencia egy konkrét példány metódusára	`containingObject::instanceMethodName`
Referencia gy konkrét típus példányának metódusára	`ContainingType::methodName`
Konstruktor referncia	`ClassName::new`

Lambda:

```
Arrays.sort(rosterAsArray,
        (a, b) ->
          Person.compareByAge(a, b)
);
```

Metódus referencia:

```
Arrays.sort(rosterAsArray,
  Person::compareByAge);
```

További Java 8 kiegészítések

- Házi feladat! ☺
- Felhasználható források:
  - http://www.oracle.com/technetwork/java/javase/8-whats-new-2157071.html
  - http://docs.oracle.com/javase/8/docs/technotes/guides/language/enhancements.html#javase8
- Érdekességek:
  - Stream API
  - Security
  - JavaFX újítások
  - Concurrency
  - java.lang / java.util csomagok
  - Java Mission Control 5.3
  - Nashorn Javascript Engine
  - Dinamikus JavaScript kód JVM-en

Mit is hoz az OpenJDK Android esetében?

- Apache Harmony elhagyása
- Nyelvi újítások gyorsabb megjelenése
- Egységesebb közösség
- Talán a Google is kap szeletet az OpenJDK fejlesztésében
- API bővülése

Mi várható a Java 9-ben? (~2017 március)

- Jshell
- Java Microbenchmarking Harness
- GC optimalizáció
- HTTP 2.0
- Process API jelentős bővítése
- És még:
  - moduláris forrás/packaging, HTML5 Javadoc, unified GC log, JavaFX újítások, multi-resolution képek, stb.
- További részletek:
  - http://openjdk.java.net/projects/jdk9/

Kitekintés

Kotlin

Köszönöm a figyelmet!

http://blog.autsoft.hu

http://www.autsoft.hu/

peter.ekler@aut.bme.hu