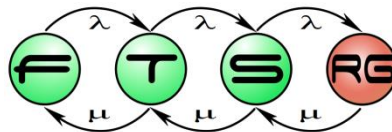


# Automatikus tesztelés

**Kövér Márton**

**Konzulensek: Honfi Dávid, Micskei  
Zoltán**

**Budapesti Műszaki és Gazdaságtudományi Egyetem  
Hibatűrő Rendszerek Kutatócsoport**



# Kitűzött feladatok

- Java tesztelési eszközök megismerése
  - JUnit
  - Mockito
  - Randoop
- Módszertan:
  - Tesztek implementációja manuálisan
  - Tesztek izolációja
  - Tesztgenerálás eszköz segítségével

# Vizsgált projekt - Algorithms

- Nyílt forráskódú projekt
- Nyelv: Java
- Egyszerű problémák megoldása
  - Adatszerkezetek
  - String műveletek
  - Rendezések
  - Matematikai műveletek
  - Mátrix műveletek

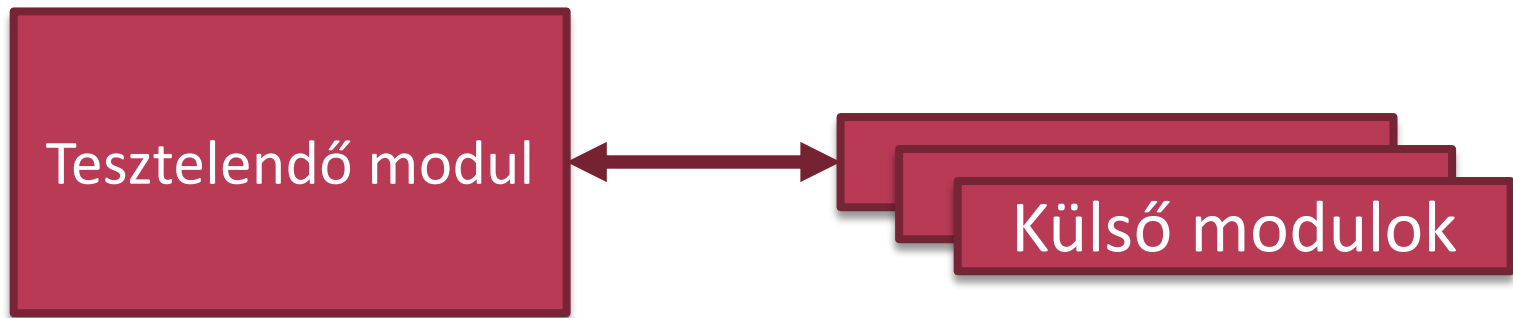
# Unit tesztelés

- Meglévő tesztek elemzése
  - Átlátható
- Kiegészítés saját tesztekkel
  - Kiírás tesztelése
  - Osztás kivonással stb..

	Lefedett osztályok	Lefedett Metódusok	Lefedett sorok
Előtte	97% (87/89)	97% (270/277)	97% (1676/1713)
Utána	98% (88/89)	98% (272/277)	98% (1693/1713)

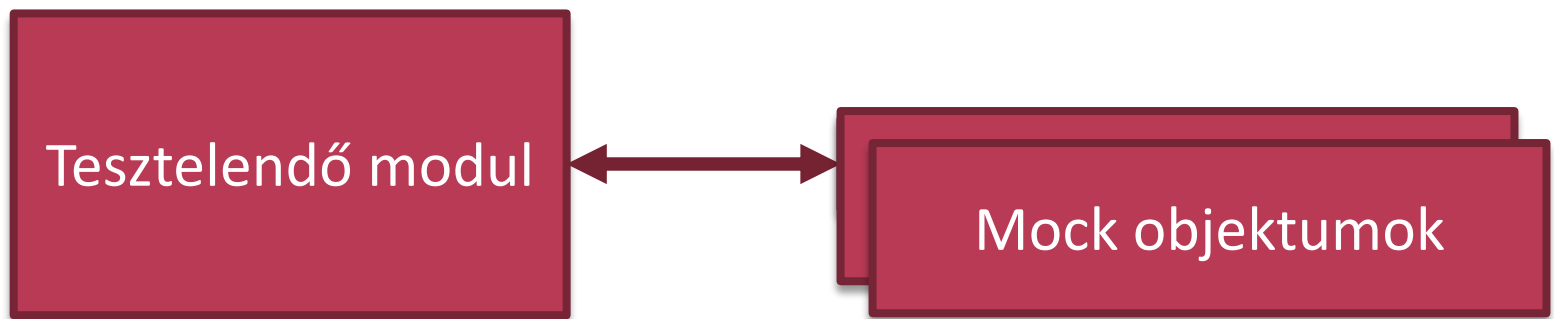
# Függőségek kezelése

- Függőségek: modulok közti kapcsolat
- Külső függőség miatti nem várt viselkedés



# Függőségek kezelése

- Függőségek: modulok közti kapcsolat
- Külső függőség miatti nem várt viselkedés
- Környezet modellezhető
  - Mock objektumok segítségével
  - Megadható visszatérési érték, átadott argumentumok ellenőrzése



# Izolációs keretrendszer

- Megismert eszköz Mockito
- Új projekt
  - Régiben egyszerűség miatt nincs függőség
  - Űrhajó
- Feladat
  - Projekt átalakítása
    - Tesztelhetőségi probléma
  - Torpedó lövés leválasztása



# Unit tesztek generálása

- Több eszköz, több módszer is létezik
- Randoop
  - Java környezethez
  - Sok teszt generálása véletlenszerű értékekkel
  - Regressziós tesztelésre alkalmas
- Kezdeti tapasztalatok
  - Futtatás nem egyértelmű
  - Nehezen olvasható a generált kód



**Randoop**

Automatic unit test generation for Java



# Randoop

- Algorithms projekt
- Talált hiba
  - NullPointerException

```
public BinaryNode getIterative(BinaryNode root, BinaryNode n1, BinaryNode n2) {  
    validateInput(root, n1, n2);  
  
    List<BinaryNode> pathToA = getPathTo(root, n1);  
    List<BinaryNode> pathToB = getPathTo(root, n2);  
    BinaryNode result = null;  
    int size = Math.min(pathToA.size(), pathToB.size());
```

- Futtatási paraméterek
  - Futási idő
  - Tesztelendő osztályok

# Randoo generált tesztek 1

	Generált tesztek	Hibát adó tesztek	Osztály lefedettség	Metódus lefedettség	Sor lefedettség
<b>Futtatás 1</b>	112	1	67	145	667
<b>Futtatás 2</b>	112	1	69	152	692
<b>Futtatás 3</b>	114	1	69	154	727
<b>Futtatás 4</b>	113	1	69	152	693
<b>Futtatás 5</b>	112	1	67	145	667
<b>Átlag</b>	112,6	1	68,2	149,6	689,2
<b>Szórás</b>			1,095445	4,27785	24,68198

Osztályok száma	Metódusok száma	Sorok száma
89	277	1713

# Randoop generált tesztek 2

Futási Idő (mp)	Generált tesztek	Hibát adó tesztek	Osztály lefedettség	Metódus lefedettség	Sor lefedettség
5	18	0	20%	1%	1%
10	54	0	51%	19%	12%
30	70	0	61%	31%	21%
60	96	1	71%	45%	31%
100	144	2	88%	64%	49%
180	182	4	91%	75%	59%
600	368	6	96%	88%	73%
1800	788	10	97%	92%	80%
3600	2161	17	96%	92%	84%

# Összefoglalás

- Elért eredmények
  - Unit tesztek készítése egy nyílt forráskódú projekthez
  - Izoláció megvalósítása egy másik projekten
  - Randoop eszközzel tesztgenerálás
- Jövőbeni tervek
  - Randoop eszköz futtatás több projekten
  - Randoop paramétereinek hatásvizsgálata