

# Projektování informačních systémů 6a

---

## **Test-driven development**

doc. RNDr. Ing. Roman Šperka, Ph.D.

# Význam testů a testování

---

- Spojení analýzy a implementace do jednoho vývojového cyklu
    - Nárůst chyb a roztržitost softwaru
  - Průběžné testování zároveň s implementací
  - Automatizované testování ještě před dokončením implementace
  - U tradičních přístupů testuje zákazník
-

# Test-driven development

---

- ❑ Pro každou drobnou součást funkcionality ve zdrojovém kódu je nutné nejprve napsat test, který dokáže příslušný kód otestovat a ověřit
  - ❑ Napsání testu před napsáním kódu
  - ❑ Po dokončení zdrojového kódu (testovaná funkce), který projde testem nastává fáze úprav kódu – refaktoring
  - ❑ Zařazení testovacího případu do testovací sady (kolekce)
-

# Testovací moduly

---

- Běží rychle,, nezdržují zdlouhavou konfigurací, rychle vracejí
  - Běží nezávisle
  - Používají pochopitelná data
  - Používají reálna data
  - Splněním testu vzniká nová funkcionality systému
  - Podpůrné nástroje JUnit nebo VJUnit
-

# Testovací případ (test case)

---

- Testovací případ, často se využívá i anglický výraz „test case“, popisuje konkrétní akce prováděné s určitou softwarovou komponentou a jejich očekávané výsledky.
  - <http://testovanisoftware.cz/dokumentace-v-testovani/test-case/>
-

```
import junit.framework.*;

public class ChangeTest extends TestCase {

    private Money old;
    private String newCur1;
    private String newCur2;
    private float exRate;
    Rates rateC = new Rates();

    public ChangeTest(String name) {
        super(name);
    }

    // příprava prostředí na test
    protected void setUp() throws Exception {
        old = new Money(25, "CZK");
        newCur1 = new String("CZK");
        newCur2 = new String("EU");
        exRate = rateC.getRateCZtoEU();
    }

    // úklid prostředí po testu
    protected void tearDown() throws Exception {

    }

    public void testChangeSame() {
        Money expected = new Money(25, "CZK");
        assertEquals(expected, old.change(newCur1));
    }

    public void testChangeDiff() {
        Money expected = new Money(25*exRate, "EU");
        assertEquals(expected, old.change(newCur2));
    }

}
```

---



Děkuji za pozornost.

Otázky?