testowanie programów

Tomasz Kokoszka

JUnit – kilka haseł

- Erich Gamma, Kent Beck
- na SourceForge od 24 listopad 2000
- Framework
- Testy jednostkowe
- Xdoclet, Struts, JOnAS

Jak można testować

```
public static void main(...) throws Exception {
   //wołanie metod, sprawdzenie wyników
   //w razie błędu rzuca wyjątek
   System.out.println("Test OK");
}
```

Wady tego rozwiązania

- Klasa najmniejszą jednostką testowania
- Przerwanie przy pierwszym błędzie
- Kod testowy obecny w kodzie wynikowym
- Brak mechanizmu do uruchamiania
- Brak mechanizmu dla raportów

Co nam daje JUnit

- Metoda najmniejszą jednostką testowania
- Przypadki testowe
- Oddzielenie testów od kodu
- Wiele mechanizmów uruchamiania
- Budowanie raportów
- Integracja z różnymi IDE

Prosty przykład

Czyli z czym to się je...

Program

```
public class DateStr {
   public String dat;
   public DateStr(String dat) {this.dat = dat;}
   public String[] split() {
        return dat.split("\\.");
   }
   public Date toDate() {
        try {
           return (new
                SimpleDateFormat("dd.MM.yyyy")).parse(dat);
        } catch (ParseException e) {}
        return null;
   }
}
```

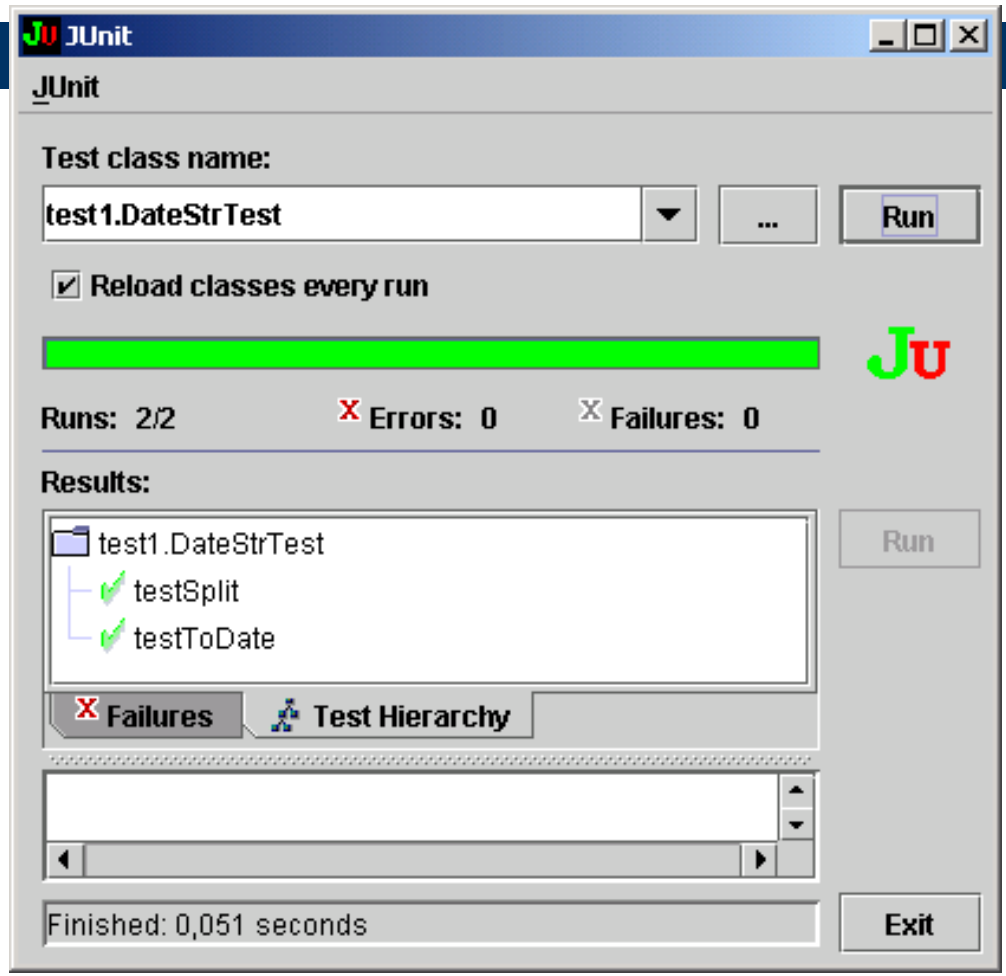
Testy jednostkowe

```
public class DateStrTest extends junit.framework.TestCase {
   private DateStr t1 = new DateStr("06.12.2003");
   public void testSplit() {
        String[] a1 = t1.split();
        assertTrue("Rozmiar>0", a1.length > 0);
        assertEquals("dzień = 06", "06", a1[0]);
   }
   public void testToDate() {
        Date d1 = t1.toDate();
        assertNotNull("Dobra data", d1);
        assertNull("Zła data",
                (new DateStr("06-12-2003")).toDate());
   }
}
```

Uruchamianie

- Konsola SWING
- Konsola tekstowa
- Skrypty budujące ANT
  - testowanie *wsadowe*
  - możliwość budowania raportu
- Integracja ze środowiskiem IDE (np. Eclipse)

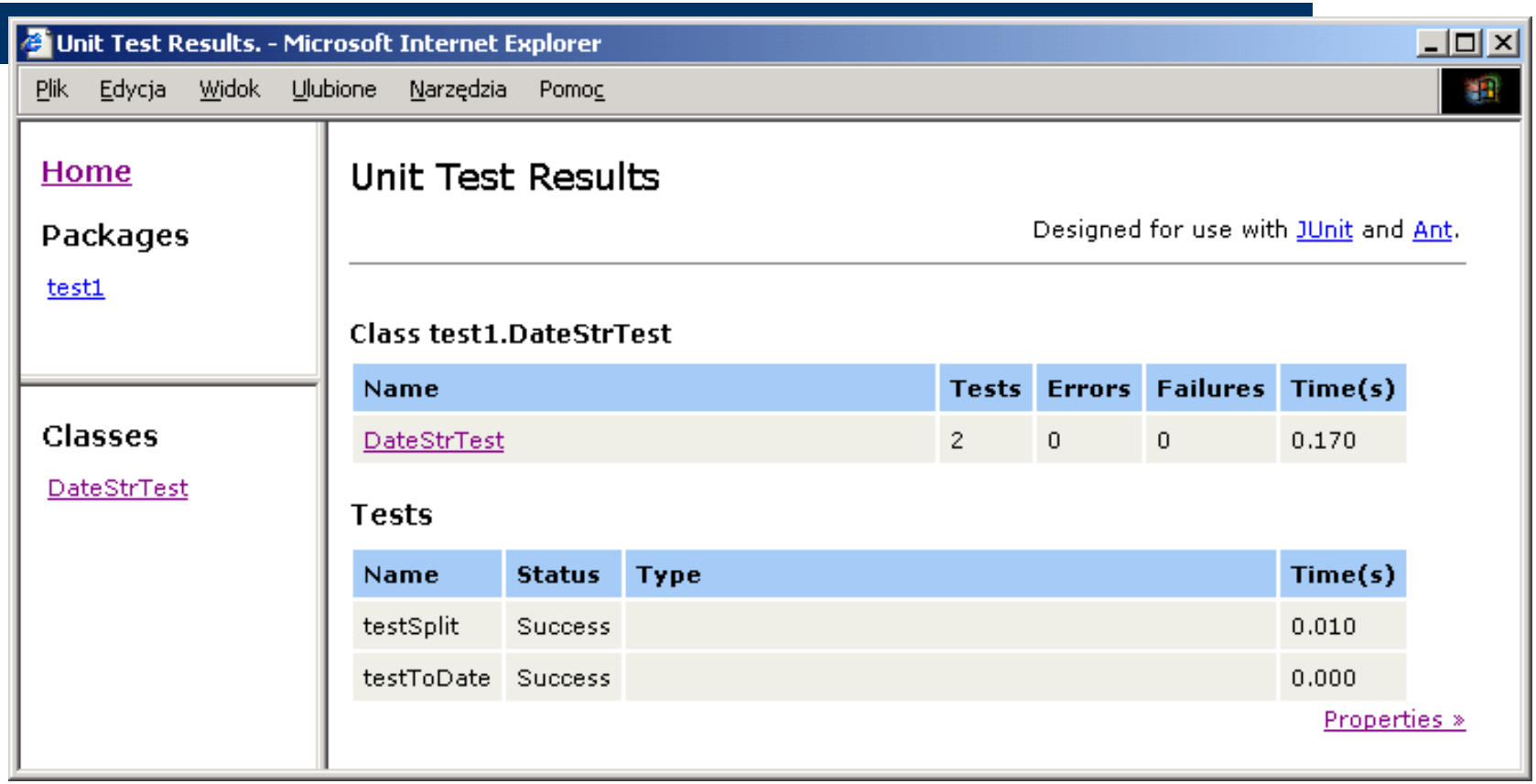
Uruchamianie – konsola SWING



Raport z testów

- Wyniki testów w XML
- Xalan: XML → HTML

Raport z testów



Jak to działa

- Klasa TestCase
- Asercje
- Nazwa klasy: <klasa>**Test**
- Nazwa metody: **test**<metoda>

Trochę więcej...

- junit.framework.TestCase
  - setUp()
  - tearDown()
- junit.framework.TestSuit
  - zbiór testów jednostkowych

Asercje

- assertTrue(b_1), assertFalse(b_1)
- assertNull(o_1), assertNotNull(o_1)
- assertEquals(o_1, o_2)
- assertSame(o_1, o_2), assertNotSame(o_1, o_2)
- fail()
- AssertionFailedError

Zróbmy sobie test

Ach ten Eclipse...

Testy wydajności

- JUnit = testy funkcjonalności
- JUnitPerf
  - na bazie JUnit (korzysta z klas testowych)
  - symuluje wielokrotne wywołanie procedur testowych
  - prędkość działania i zużycie zasobów
  - przydatny tylko w początkowej fazie projektu

Nie samą Javą żyje człowiek...

- CppUnit
- NUnit
- utPLSQL – PL/SQL
- HttpUnit – drzewo obiektów strony
- DUnit – Delphi
- J2EEUnit – obecnie jakarta-Cactus

Dalsze informacje

- www.junit.org
- otn.oracle.com/oramag/oracle/03-may/o33junit.html
  - JUnit krok po kroku
- www.3plus4software.de/eclipse/junit_en.html
  - JUnit w Eclipse krok po kroku
- www.e-urzad.pl/testy_testow.html
  - „*Testy testów*”

Pytania, uwagi

http://rainbow.mimuw.edu.pl/~tk189405/JUnit