X36PJC
Proseminář #3

Unit testing v C/C++

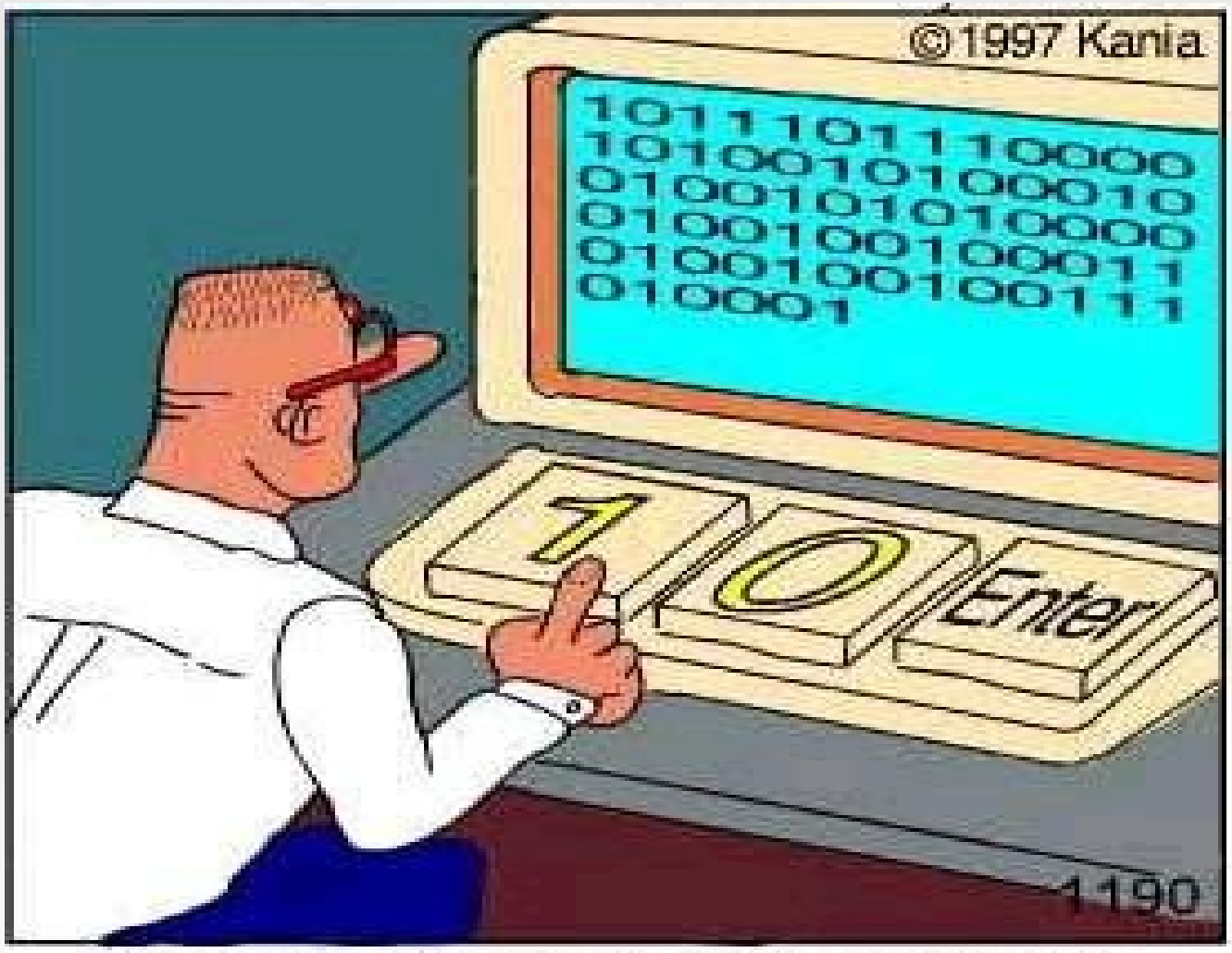
Unit testing

- Testování samostatné části programu
- Testují se procedury nebo objekty
- Každá část aplikace musí být samostatně stoprocentně funkční
- Testuje se (pokud možno) bez návaznosti na ostatní části programu
- Lze provádět ručně nebo použít k tomu vytvořený framework
- Test musí dát jednoznačnou odpověď, zda proběhl úspěšně nebo ne

Test-driven development

- Relativně mladá metodika návrhu softwaru, která značně zkracuje životní cyklus vývoje softwaru
- Způsob tvorby aplikací směrem shora dolů
- Nejdříve se jasně formulují požadavky na vytvářený systém
- Požadavky jsou vyjádřeny vytvořením testovacího prostředí
- Systém se pak tvoří na základě požadavků testovacího prostředí

Extrémní programovaní I.



Real programmers code in binary.

Extrémní programování I.

- Není nápní předmětu, takže jen stručně...
- Unit testing a Test-driven development jsou jedny ze součástí extrémního programování
- Životní cyklus vývoje softwaru:
  - Návrh testovacího kódu
  - Tvorba softwaru dle požadavků testovacího kódu
  - Úspěšný průchod testovacím kódem
  - Úprava softwaru
  - Úspěšný průchod testovacím kódem
  - ...

Boost C++ Libraries I.

- Jedná se o soubor „modulů“ pro C++
- Rozšiřují funkcionalitu jazyka
- Některé z těchto modulů pomalu pronikají do standardu jazyka
- Všechny moduly jsou kompilovány do jedné statické nebo dynamické knihovny

Boost C++ Libraries II.

- Kompilace na Solarisu je poměrně náročná na výkon a místo na disku (60 MB zdrojových kódů)
- Pro kompilaci je třeba nastavit správně cesty ke k hlavičkovým souborům a ke knihovně:

```
-I/opt/sfw/include/boost_1_44_0/
-L/opt/sfw/lib/boost_1_44_0/
```

Boost – The Unit Test Framework I.

- Součástí rozšiřujících knihoven

  pro jazyk C++ - Boost C++ Libraries
- Dostupná prakticky pro všechny platformy
- Knihovna pro zjednodušení testovaní samostatných procedur a objektů (unit testing)
- Nabízí kategorizaci testů a přehledný výstup výsledků
- Při selhání testu přímo vypíše příslušný řádek!
- Při kompilaci linkovat s libboost_test_exec_monitor:

```
g++ … -lboost_test_exec_monitor
```

Boost – The Unit Test Framework II.

- Ukázkový kód (test.cpp)

```
#define BOOST_TEST_MODULE example

#include
<boost/test/included/unit_test.hpp>

#include „MyClass.hpp“


BOOST_AUTO_TEST_CASE( test_case1 ){
   MyClass myClass;
   BOOST_CHECK(myClass.empty());
}
```

Boost – The Unit Test Framework III.

- Ukázkový výstup

```
#>./test
Running 1 test case...

*** No errors detected
```

Boost – The Unit Test Framework IV.

- Ukázkový výstup s detaily

```
./test --report_level=detailed
Running 1 test case...

Test suite "example" passed with:
  1 assertion out of 1 passed
  1 test case out of 1 passed

  Test case "test_case1" passed with:
    1 assertion out of 1 passed
```

Boost – The Unit Test Framework V.

- Přehled organizačních maker
  - BOOS_TEST_MODULE nazev
    - Definuje název celého unit testu
  - BOOST_AUTO_TEST_CASE(nazev){}
    - Případový test - používá se pro testování jednoho problému (případu), např. funkce konstruktoru
  - BOOST_AUTO_TEST_SUITE(nazev)
  - BOOST_AUTO_TEST_SUITE_END()
    - Párová makra, seskupují makra pro případové testy do sad. Jednotlivé případové testy jsou uvedeny v kódu mezi nimi.

Boost – The Unit Test Framework VI.

- Testovací makra BOOST_<level>_<type>
  - *level* – existují 3 možnosti

    WARN – ve výstupu zobrazí pouze varování

    CHECK – ve výstupu vypíše chybovou hlášku

    REQUIRE – ve výstupu vypíše chybovou hlášku a zastaví průběh testování
  - *type* – udává typ testu

Boost – The Unit Test Framework VII.

- Vybrané typy testovacích maker
- EQUAL(vyraz1, vyraz2)
  - Stěžuje si pokud si nejsou výrazy rovny
- EQUAL_COLLECTION(...)
  - Stěžuje si pokud si nejsou rovna dvě pole hodnot
- NE(vyraz1, vyraz2)
  - Stejné jako EQUAL, jen naopak (= NOT EQUAL)
- THROW(vyraz, vyjimka)
  - Stěžuje si pokud výraz navyhodí výjimku

Boost – The Unit Test Framework VIII.

- Vybrané typy testovacích maker
- BOOST_<level>(vyraz)
  - Testuje pouze pravdivost vyrazu
- BOOST_MESSAGE(vyraz, text)
  - Při nepravdivosti výrazu zobrazí na výstup požadovaný text
- BOOST_<level>_NO_THROW(...)
- BOOST_IS_DEFINED(makro)
- Více info: User guide – Testing tools - reference

Použité zdroje a užitečné odkazy

- http://www.boost.org/doc/libs/1_44_0/libs/test/doc/html/index.html
  - Hlavní stránka Boost – The Unit Test Framework