

Správa projektů a testování

doc. Ing. Miroslav Beneš, Ph.D.
katedra informatiky FEI VŠB-TUO
A-1007 / 597 324 213
<http://www.cs.vsb.cz/benes>
Miroslav.Benes@vsb.cz



Programátorské nástroje

- Příprava zdrojových textů
- Překlad a sestavení programů
- Údržba verzí
- Řízení postupu při sestavování projektu
- Ladění programů
- Testování funkcí
- Testování výkonnosti
- Generování dokumentace
- Lokalizace

Správa projektů

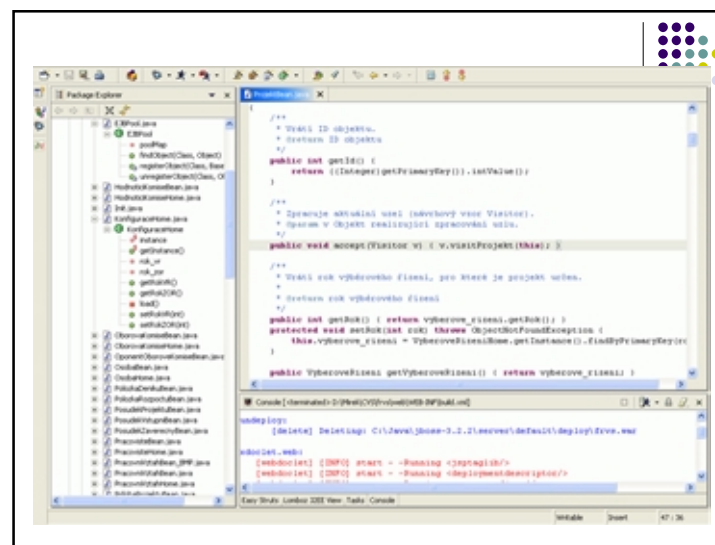
2

Integrovaná vývojový prostředí (IDE)

- Poskytují většinu uvedených funkcí současně
- Univerzální nebo specializovaná na konkrétní programovací jazyk
- Příklady:
 - Borland Pascal
 - JBuilder, Eclipse, NetBeans, JCreator, Sun ONE
 - MS Visual Studio, KDeveloper, SharpDevelop

Správa projektů

3

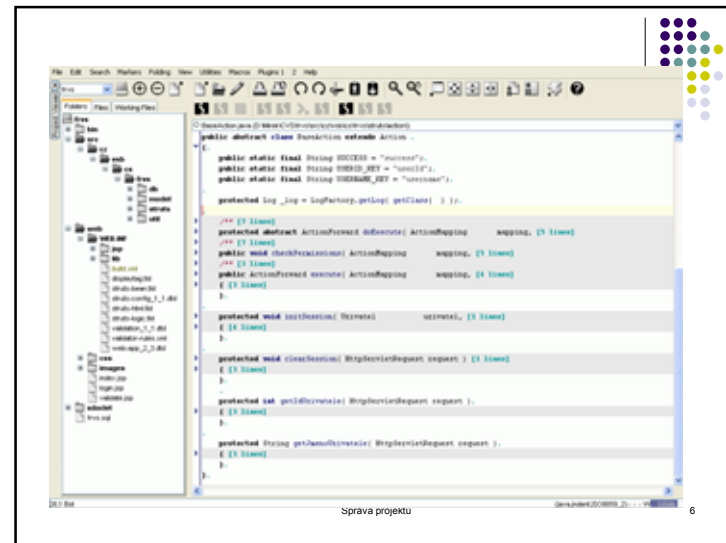


Příprava zdrojových textů

- Textové editory
 - zvýraznění syntaxe (syntax highlighting)
 - kontrola závorek
 - vyhledávání a nahrazování – soubor, projekt
 - šablony a makra
 - sbalení textu (folding)
 - spolupráce se správou verzí
- Příklad: gvim, jEdit

Správa projektů

5



Správa projektu

6

Překlad a sestavení programů

- Překladače
 - generování binárních modulů
 - kontrola syntaxe
 - sémantické kontroly
 - upozornění na možné problémy
 - inkrementální překlad
 - podpora ladění
- Sestavovací programy
 - spojení binárních modulů do spustitelného programu
 - knihovny modulů

Správa projektů

7

Údržba verzí

- Vývoj v týmu
 - sdílení zdrojových textů
 - ochrana před současnými modifikacemi
- Vývoj různých verzí
 - Verze pro různá cílová prostředí
 - Vývojové verze – 0.1, 1.1beta, 1.2RC3, ...
- Archivace
 - Možnost návratu po nevhodných změnách
 - Zjištění rozdílů ve verzích

Správa projektů

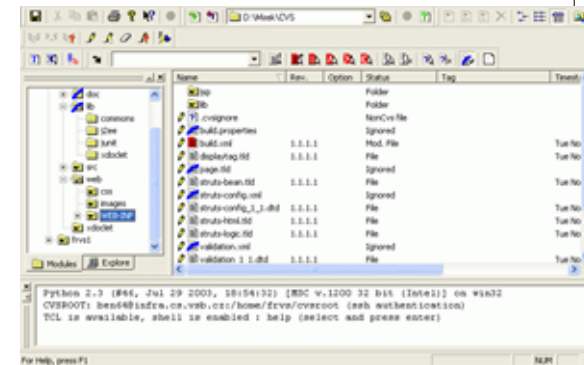
8

Údržba verzí

- Repository
 - originální soubory
 - změnové soubory + komentáře (diff)
 - může být na sdíleném serveru nebo v síti
- Postup
 - získání souborů ze serveru (checkout)
 - práce v lokálním prostředí
 - aktualizace změněných souborů (commit)

Správa projektů

9



Správa projektů

10

Správa projektů

- Nezávislost na prostředí
 - umístění knihoven, programů
 - verze a varianty nástrojů (např. javac/jikes)
- Udržení konzistence
 - sledování závislostí
- Optimalizace budování projektu
 - vyhledání nejkratší cesty – zpracování pouze změněných a na nich závislých souborů

Správa projektů

11

Správa projektů

- Typické činnosti
 - inicializace prostředí (adresáře, parametry, ...)
 - překlad a sestavení programů
 - systematické testování
 - generování dokumentace
 - odstranění pracovních souborů
 - vytváření archivů a distribucí
 - instalace

Správa projektů

12

Správa projektů

- Nástroje
 - dávky (.bat, .sh) – nevhodné
 - make
 - řízení pomocí skriptu Makefile
 - Ant
 - řízení pomocí XML souboru (build.xml)
 - možnost rozšiřování – plugins (Java)

Správa projektů

13

Dávkové zpracování

preloz.sh

```
yacc -o synt.cpp -d synt.y  
lex -o lex.cpp lex.l  
gcc -o prekl synt.cpp lex.cpp main.cpp
```

- opakovaný překlad nezměněných souborů
- obtížná údržba při rozšiřování projektu

Správa projektů

14

Program make

- Sestavení *cílových objektů* na základě *předpokladů*
- Využití implicitních pravidel
 - prog.cpp -> prog.o -> prog
- Makrodefinice
 - SRCS = prog.cpp lex.cpp synt.cpp
 - prog: \$(SRCS)
gcc -o prog \$(SRCS)

Správa projektů

15

Příklad

```
LFLAGS = -l  
  
all: p4  
par.o: par.c lex.c  
p4: par.o  
    $(CC) -o p4 par.o  
clean:  
    $(RM) par.c par.o lex.c  
allclean: clean  
    $(RM) p4  
dist:  
    tar -czf p4.tgz Makefile lex.l par.y
```

Správa projektů

16

Program Ant

- Implementován v prostředí Java
- Možnost rozšiřování
 - definované programátorské rozhraní
 - přidávání dalších akcí
- Činnost se řídí souborem v XML
 - build.xml
- Integrovan do mnoha vývojových prostředí
 - jEdit, Eclipse, ...

Správa projektů

17

Příklad

```
<?xml version="1.0"?>

<project name="p3" basedir="." default="compile">
  <property name="build.compiler" value="jikes"/>
  <target name="compile">
    <javac srcdir="." destdir="classes"
      debug="true" deprecation="true" />
  </target>

  <target name="jar" depends="compile">
    <jar jarfile="uloha3.jar" manifest="uloha3.mf"
      compress="true" basedir="classes"/>
  </target>
</project>
```

Správa projektů

18

Příklad

```
<target name="all" depends="compile"
  description="Build everything.">
  <echo message="Application built."/>
</target>

<target name="run" depends="all">
  <java classname="Test" fork="true"
    failonerror="true">
    <classpath>
      <pathelement location="classes"/>
    </classpath>
    <arg file="g1.txt"/>
  </java>
</target>
```

Správa projektů

19

Příklad

```
<target name="javadoc" description="Building documentation.">
  <mkdir dir="apidoc"/>
  <javadoc packageNames="grammar.*" destdir="apidoc">
    <sourcepath>
      <pathelement location="."/>
    </sourcepath>
  </javadoc>
</target>

<target name="clean" description="Clean all build products.">
  <delete>
    <fileset dir=".">
      <include name="**/*.class,**/*.java-"/>
    </fileset>
  </delete>
  <delete file="uloha3.jar"/>
  <delete dir="apidoc"/>
</target>
</project>
```

Správa projektů

20

Ladění programů

- Ladicí výpisy
 - Tisk do souboru
 - Logování (log4j, ...) – možnost konfigurace
- Sledování stopy programu (trace)
- Analýza obsahu paměti po chybě

Správa projektů

21

Ladění programů

- Ladění běžícího programu
 - na zdrojové / binární úrovni
 - prohlížení a změna obsahu proměnných
 - body zastavení (breakpoint)
 - krokování výpočtu (step into / step over)
 - podmíněné zastavení programu
- Ladění s inkrementálním překladem
 - lokální změna programu a pokračování v ladění

Správa projektů

22

Testování programů

- „Ruční“ testování
 - náročné
 - není opakovatelné
 - snadno se přehlédnou chyby
- Automatické testování
 - sada jednotkových testů (unit tests)
 - systematické testování
 - rozšiřování testů o reklamace
 - generování testů

Správa projektů

23

Testování programů

- JUnit
 - prostředí Java
 - programátorské rozhraní pro tvorbu testů
 - nástroje pro automatické provádění testů
 - grafické rozhraní
 - příkazový řádek, akce pro Ant
 - využití reflexe pro vyhledání testů uvnitř tříd
 - rozšíření pro testování webových aplikací, servletů, ...

Správa projektů

24

JUnit

- `import junit.framework.TestCase;`
- `public class TestXYZ extends TestCase { }`

- `void setUp() {...}` *inicializace*
- `void tearDown() {...}` *finalizace*
- `void testX() {...}` *krok testu*

- `assertTrue(podmínka), assertFalse(...)`
- `assertEquals(x,y), assertNotNull(x)`
- ...

Správa projektů

25

Příklad

```
package cviceni3;
import junit.framework.TestCase;

public class TestZlomek extends TestCase {
    protected Zlomek z1 = new Zlomek(1, 3);
    protected Zlomek z2 = new Zlomek(2, 6);
    protected Zlomek z3 = new Zlomek(2, 3);

    protected void setUp() { }
    protected void tearDown() { }
    public void testEquals()
    {
        assertEquals(z1, z1);
        assertEquals(z1, z2);
    }
    public void testAdd()
    {
        Zlomek result = Zlomek.plus(z1, z2);
        assertEquals(result, z3);
    }
}
```

Správa projektů

26

Testování výkonnosti

- Sledování počtu volání určitých funkcí
- Sledování času stráveného výpočtem různých částí programu
 - podklad pro optimalizaci změnou algoritmu
- Nástroje: profilers
 - gprof
 - součást vývojových prostředí

Správa projektů

27

Generování dokumentace

- Oddělená dokumentace
 - problémy s aktualizací
 - nutnost uvádět kompletní specifikace
- Dokumentace jako součást zdrojového textu
 - snadnější údržba (např. včetně verzování)
 - literární programování (D. Knuth)
 - dokumentační značky - javadoc

Správa projektů

28

Program javadoc

- Dokumentace ve speciálních poznámkách

```
/**
 * Dokumentační poznámka
 */
```
- Dokumentační značky + HTML

```
@author <a href=mailto:joe@mit.edu>Joe</a>
@param x Popis parametru x
```
- Rozšíření – generování zdrojových textů pomocí šablon - XDoclet

Příklad

```
/**
 * Konstruktor zlomku.
 * Naplní čitatele a jmenovatele a převede
 * zlomek do normalizovaného tvaru.
 * @param citatel Čítatel zlomku.
 * @param jmenovatel Jmenovatel zlomku.
 */
public Zlomek(int citatel, int jmenovatel)
{
    this.citatel = citatel;
    this.jmenovatel = jmenovatel;
    normalizuj();
}
```

Lokalizace

- úprava programů pro mezinárodní prostředí
- překlad textových řetězců
 - extrakce textů – překlad – vložení zpět
 - oddělení textů od programu
 - např. soubory .properties v různých jazykových verzích
- knihovny tříd pro formátování
 - přizpůsobení prostředí uživatele (Locale)