Západočeská univerzita v Plzni
Centrum informatizace a výpočetní techniky

Informační Bulletin
CIV

JAVA
vývojová prostředí

Západočeská univerzita v Plzni
Centrum informatizace a výpočetní techniky

Informační Bulletin

1

Únor 2005

Příspěvky uvedené v bulletinu jsou dílem kolektivu autorů CIV, autorem kapitoly 3 je Roman Tesař, KIV. Publikace neprošla jazykovou ani grafickou úpravou. Autoři děkují Doc. Pavlu Heroutovi za korekturu a připomínky.

Redakční rada: J. Sitera, J. Valdman a R.Tesař.

Sazba písmy Bitstream Charter a Concrete v systému LaTeX 2_ε.
Vytiskl TYPOS — Digital Print s.r.o., závod Plzeň.

První vydání, náklad 250 výtisků.
Vydala Západočeská univerzita v Plzni.

ISBN 80-7043-357-4

Obsah

1 Úvod 5

2 Java v prostředí Orion 7
2.1 Kde hledat informace a pomoc . . . 7
2.2 Základní poskytované služby . . . 7
2.3 Podpora JRE (Java Runtime Environment) . . . 8
2.4 Vývoj programu v jazyce Java . . . 8
2.4.1 Přeložení a spuštění programu z příkazové řádky . . . 8
2.4.2 Vývojové prostředí pro jazyk Java v editoru Emacs . . . 9
2.4.3 Eclipse v prostředí Orion . . . 10
2.4.4 Učebna UNIXových stanic . . . 11
2.4.5 Vývoj apletů . . . 11
2.4.6 Dokumentace . . . 11
2.5 Řešení problémů při domácí instalaci JDK . . . 11

3 Základy práce v Eclipse 13
3.1 Úvodem . . . 13
3.2 První spuštění . . . 13
3.3 Editory . . . 14
3.4 Pohledy . . . 15
3.5 První Java projekt . . . 18
3.6 Parametry příkazové řádky . . . 20
3.7 Import již existujícího projektu . . . 20
3.8 Krokování programů . . . 21
3.9 Vlastnosti editoru Java kódu . . . 23
3.10 Instalace rozšiřujících modulů, domácí instalace . . . 28
3.11 Závěr . . . 28

4 Nadstavby nad Eclipse 29
4.1 IBM WebSphere Studio . . . 29
4.2 IBM Rational Development Platform . . . 31
4.3 Další nástroje . . . 31

Kapitola 1

Úvod

Sborník, do kterého právě nahlížíte, je další z monotématicky zaměřených publikací Centra informatizace a výpočetní techniky Západočeské univerzity v Plzni (CIV–ZČU). Jeho posláním je oslovit konkrétní zájmovou skupinu uživatelů výpočetní techniky – tentokrát studentů programování v jazyce Java – a pomoci jí řešit některé z mnoha starost. Hlavní část obsahuje základy používání vývojových nástrojů pro tvorbu a ladění programového kódu Java, které jsou k dispozici v prostředí ORION..

Obsah sborníku je užitečný jak pro uživatele výpočetního prostředí ZČU ORION, tak pro čtenáře, kteří chtějí primárně používat vývojové prostředí jazyka Java na svém vlastním počítači doma či na koleji.

Tento sborník vás nemá naučit programovat. Nicméně vznikl díky spolupráci s Katedrou informatiky a výpočetní techniky (KIV) Fakulty aplikovaných věd a je primárně obsahově zamýšlen jako doplněk k výuce programování v jazyce Java, kterou KIV zajišťuje. Chcete-li se naučit programovat, zapište si předmět na KIV a/nebo použijte tomuto předmětu odpovídající literaturu, chcete-li se naučit pro svoji práci používat vývojové nástroje, pak je vám určen obsah tohoto sborníku.

Jádrem sborníku je popis vývojového prostředí Eclipse. Toto prostředí bylo zvoleno jako primárně podporovaný vývojový nástroj pro veřejné laboratoře CIV (ORION LINUX a ORIONXP), přičemž se jedná o moderní a perspektivní řešení, které je k dispozici zdarma, a má tak značnou naději být zvoleno i prostředím pro domácí počítače studentů ZČU.

Sborník začíná kapitolou (2), která vás seznámí s výpočetním prostředím ORION, s tím co vám dává k dispozici CIV pro vaši práci a se základními fakty a postupy; mj. kapitola obsahuje podrobný postup spuštění vývojového prostředí Eclipse v učebnách CIV. Další, největší kapitola (3) obsahuje již vlastní popis Eclipse. Pokud chcete toto prostředí používat na svém vlastním počítači, můžete přeskočit rovnou na tuto kapitolu. V jejím závěru najdete i návod, jak Eclipse nainstalovat. Sborník je uzavřen (kap. 4) nástinem dalších možností, které vývojové prostředí Eclipse skýtá. Ten je určen pro ty čtenáře, jejichž cílem není jen Eclipse použít, ale třeba se i aktivně zapojit do vývoje aplikací na ZČU (v rámci výuky na KIV nebo individuální spolupráce s CIV–PIS) nebo jen získat vodítko pro další rozvoj svých znalostí s Eclipse spojených.

KAPITOLA 2

JAVA V PROSTŘEDÍ ORION

Výpočetní prostředí ORION a jeho služby uživatelům ZČU jsou popsány v předchozích sbornících CIV a v dokumentaci na WWW stránkách. Následující řádky slouží pro základní a rychlou orientaci, nejsou kompletním popisem služeb, které můžete od CIV očekávat. Základní informace o výpočetním prostředí ORION jsou shrnuty v příručce „První krůčky“ (sborník CIV 3/2004). Zde lze najít vše potřebné pro vstup do ORIONu.

2.1 KDE HLEDAT INFORMACE A POMOC

Základním zdrojem informací je WWW stránka `http://support.zcu.cz`. Zde lze najít zejména:

- *Kde jsou učebny CIV* – mapy, popis na které učebně je jaký systém.
  `http://support.zcu.cz/sluzby/ucebny.html`
- *Sborníky CIV* – sborníky CIV v elektronické formě.
  `http://support.zcu.cz/manualy.html`
- *Jak kontaktovat uživatelskou podporu* – HelpDesk CIV je tu pro vás.
  `http://support.zcu.cz/helpdesk.html`

2.2 ZÁKLADNÍ POSKYTOVANÉ SLUŽBY

Pro programování v Javě budete ze služeb které vám CIV nabízí potřebovat:

- *Uživatelské konto a prostor na sdíleném souborovém systému* – toto získáte registrací uživatele. Sdílený souborový systém vám poskytuje domovský adresář přístupný ze všech míst (stanice v učebnách, veřejné servery) a to jak ve Windows tak v Linuxu.
- *Pracovní místo v učebně* – seznam učeben a mapa viz výše. Zde se dozvíte, že na některých učebnách je operační systém založený na Windows NT zatímco na většině ostatních Windows XP. Podpora Javy se liší, starší systém založený na Windows NT (ORIONT) nepodporuje Eclipse. Pracovní místo v učebně si můžete rezervovat na `http://rezervace.zcu.cz/`.
- *Veřejné servery* – z libovolného počítače na Internetu (třeba z kolejí) lze pomocí klienta SSH přistupovat na server `eryx` a zde pracovat. K dispozici není Eclipse, ale pouze příkazová řádka.

- *Elektronická pošta* – možná budete potřebovat pro komunikaci (nejen) se svým vyučujícím. V každém systému na učebně je k dispozici program elektronické pošty Mozilla Thunderbird, nebo lze použít program pine a případně lze ke své poštovní schránce přistupovat z libovolného WWW prohlížeče přes adresu `http://webmail.zcu.cz/`.
- *Tiskové služby* – budete-li potřebovat vytisknout svůj program nebo textovou zprávu, máte k dispozici tiskové služby. Tisk jehličkovou tiskárnou je v tomto případě zdarma, tisk laserovou tiskárnou (lze i barevně či oboustranně) za poplatek.
  Více viz `http://support.zcu.cz/tisk/`.

2.3 PODPORA JRE (JAVA RUNTIME ENVIRONMENT)

Pro spuštění libovolného programu v jazyce Java je potřeba podpora tzv. JRE v operačním systému. V systému obvykle bývá nainstalováno více verzí JRE, které navíc tvoří podmnožinu JDK (Java Development Kit).

JRE, někdy nesprávně zaměňované za JVM (Java Virtual Machine) je navíc k dispozici od různých dodavatelů: "standardní" je verze od firmy SUN, bežně se používá i JRE firmy IBM, které má některé odlišnosti. V prohlížeči Microsoft Internet Explorer, potaženo celém operačním systému Windows bývá nainstalováno JRE od firmy Microsoft, které se nejvíce odchyluje od standardů, nicméně je často vyžadováno produkty typu internet banking.

JRE existuje v různých verzích, aktuální je verze 5 (JRE 1.5), bežně se používá 1.4.2 nebo z důvodů kompatibility i starší verze 1.3. Java existuje v několika verzích, základní je Java2 Standard Edition (J2SE), dále se používá Java2 Enterprise Edition (J2EE) a Java2 Micro Edition (J2ME).

Na veřejných serverech (`eryx`) je k dispozici J2SE 1.4.1. Stejná verze je k dispozici i v instalaci ORION LINUX. V ORIONXP je instalována verze J2SE 1.4.2. Vzhledem k minimálním změnám mezi oběma verzemi J2SE by přenášení mezi ORIONXP a ORION LINUX nemělo činit potíže.

Na začátku školního roku 2005/2006 je plánován přechod na verzi J2SE 5.0 ve všech systémech ORION.

2.4 VÝVOJ PROGRAMU V JAZYCE JAVA

Pro základní kompilaci a spuštění programu stačí příkazová řádka. Pro rozsáhlejší práci zvolte vývojové prostředí Eclipse.

2.4.1 PŘELOŽENÍ A SPUŠTĚNÍ PROGRAMU Z PŘÍKAZOVÉ ŘÁDKY

Náš zdrojový kód v jazyce Java musí být uložen v souboru s příponou `.java` pojmenovaném stejně (včetně malých a velkých písmenek) jako veřejná (`public`) třída (na následujícím příkladu je to třída `Prvni`).

```
public class Prvni {
  public static void main(String[] args) {
     System.out.println("Zdar a silu");
  }
}
```

Nyní je potřeba zdrojový kód přeložit. V systémech Windows spusťte příkazový řádek (command prompt), který naleznete v menu **Start > Programy > Příslušenství > Příkazový**

řádek (v systémech Linux spusťte konzoli). Nastavte se do adresáře, kde máte soubor se zdrojovým Java kódem uložen a proveďte překlad příkazem

```
javac Prvni.java
```

V aktuálním adresáři se objeví (pokud jsme ve zdrojovém kódu neudělali syntaktickou chybu) přeložený soubor Prvni.class obsahující přeložený bytecode. Tento soubor pomocí interpretru spustíme

```
java Prvni
```

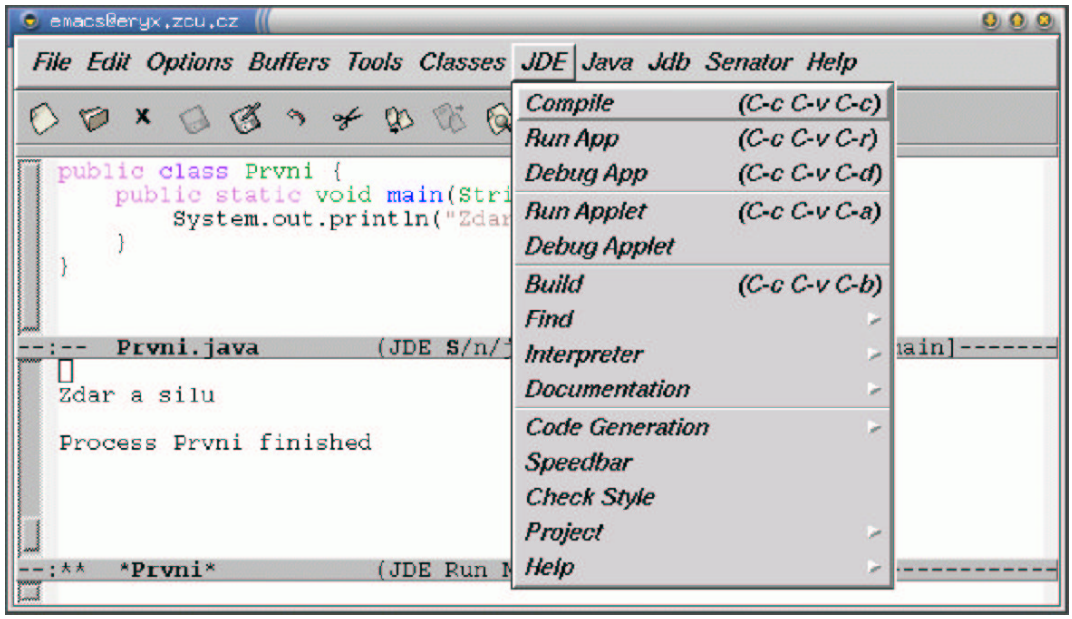
Na obrazovce se objeví výpis "`Zdar a silu`".

! Nezapomeňte, že Java rozlišuje malá a velká písmena (je case sensitive), často je tento fakt zdrojem (dlouhou dobu hledaných a nepochopitelných) chyb.

! Pozor u příkazu `javac <název souboru>`, vyžaduje jako svůj parametr název překládaného zdrojového souboru včetně přípony `.java`! Stejně dejte pozor u příkazu `java <název souboru>`, ten vyžaduje název přeloženého souboru s bytecodem bez přípony `.class`!

2.4.2 VÝVOJOVÉ PROSTŘEDÍ PRO JAZYK JAVA V EDITORU EMACS

V editoru Emacs, hlavním podporovaném editoru v prostředí ORION LINUX, je k dispozici vývojové prostředí pro Javu *Java Development Environment for Emacs*:



Obrázek 2.1: Práce s Java kódem v prostředí Emacs (Xwindows)

Vývojové prostředí pro Javu se nahraje v okamžiku, kdy je rozpoznáno, že editujete javovský soubor. V nabídce přibudou položky `Classes`, `JDE`, `Java`, `Jdb` a `Senator`. Základní klávesové zkratky pro práci s prostředím jsou tyto:

`C-c C-v C-c`	kompilace otevřeného souboru
`C-c C-v C-r`	spuštění přeložené aplikace
`C-c C-v C-d`	spuštění aplikace v debugeru

Další velice užitečnou funkcí je tzv. Speedbar, který spustíte z menu `JDE→Speedbar`. Umožní vám se rychle orientovat ve zdrojových souborech javovských tříd formou stromu, který lze rozbalit až na úroveň parametrů metod jednotlivých tříd a kliknutím myší se přepnout na patřičné místo ve zdrojovém kódu. Stejnou možnost nabízí i menu `Classes`.

Detailní dokumentace k prostředí je na stránkách projektu JDEE[1], na něž se lze také dostat z menu Emacsu `JDE→Help JDEE User's Guide`.

2.4.3 ECLIPSE V PROSTŘEDÍ ORION

Vývojové prostředí Eclipse (verze 3.0.1)lze spustit v prostředí ORIONT přes hlavní systémové menu **Start > Programy > Eclipse** (v Linuxu **Aplikace > Vývoj > Eclipse**).

Pracovní adresář, ve kterém budou ukládány všechny vaše projekty (obsahující zdrojové kódy, přeložené .class soubory a další) vytvořené v prostředí Eclipse, naleznete ve svém domovském adresáři H:\eclipse (v Linuxu $HOME/eclipse). Zde musí být většinou umístěny (pokud nespecifikujete ve svém programu jinak) i případné soubory obsahující vstupní data pro vaše programy.

! Kromě adresáře \eclipse naleznete ve svém domovském adresáři i adresář \.eclipse, který je systémový a obsahuje různá nastavení včetně vzhledu a rozmístění oken v prostředí Eclipse. Tohoto adresáře si doporučuji nevšímat a určitě ho žádným způsobem nemodifikujte.

! Protože adresář s projekty je umístěn ve vašem domovském adresáři, přičemž prostředí Eclipse je na tento adresář automaticky nastaveno, budou vaše vytvořené projekty vždy přístupné a vzhled prostředí bude vždy stejný nezávisle na tom, zda spustíte Eclipse v ORIONT nebo v Linuxu.

Pozor při otevírání zdrojových Java souborů vytvořených v jiných operačních systémech. Linux standardně používá znakovou sadu ISO-8859-2, zatímco Windows CP1250. Komentáře nebo výpisy řetězců s diakritikou by totiž nemusely vypadat přesně tak jak jste původně zamýšleli. Také znaky ukončující řádek (ve Windows je to *CRLF*, v Linuxu *LF*) mohou působit potíže.

! Prostředí Eclipse (veze 3.0.1) nabízí při editaci souborů možnost přepínání mezi znakovými sadami v menu **Edit > Encoding**. V menu **Edit > Convert Line Delimiters To** je potom možné změnit znaky konce řádků.

Dále dejte v Linuxu pozor, pokud budete chtít v Eclipsu editovat soubory nahrané z CD. Nemají totiž nastavené právo zápisu a zdrojový kód není možné editovat. Právo zápisu je nutné u souborů zkopírovaných z CD explicitně nastavit.

Při práci s prostředím Eclipse nemodifikujte vaše projekty jinak než opět v prostředí Eclipse. Pokud ukončíte práci s prostředím Eclipse a smažete nebo modifikujete nějaký soubor, který byl ve chvíli ukončení součástí vašeho projektu v prostředí Eclipse, může se stát, že váš projekt nepůjde při dalším spuštění prostředí přeložit. V tomto případě je vhodné vytvořit nový projekt, zdrojové soubory ze starého nefungujícího projektu do nově vytvořeného překopírovat a nefungující projekt smazat. Samozřejmě zdrojové .java soubory, které jsou součástí vašich projektů, je možné libovolně (a beztrestně) editovat jakýmkoliv

[1] `file:///usr/share/doc/jde/html/jde-ug/jde-ug.html`

jiným editorem přímo i v pracovním adresáři prostředí Eclipse (i spuštěného, Eclipse si změny souboru všimne, pokud ho má otevřený), ale u ostatních typů souborů to raději nezkoušejte.

Pokud možno nespouštějte dvě prostředí Eclipse současně. Při spuštění druhého prostředí budete vyzváni, abyste zadali název nového pracovního adresáře, protože ten přednastavený ve vašem domovském adresáři je již využíván prvním spuštěným prostředím. Pokud zadáte nový pracovní adresář, musíte počítat s tím (především při odchodu od počítače), že zde budou umístěny všechny projekty, které ve druhém prostředí vytvoříte.

2.4.4 Učebna UNIXových stanic

Jak již bylo řečeno, je v učebnách CIV (v systému Linux a Windows) podporováno pouze vývojového prostředí Eclipse, ačkoli pro jazyk Java existuje řada jiných vývojových prostředí. Nicméně CIV dává uživatelům k dispozici také jednu učebnu se stanicemi SUN, která je obecně zaměřena na podporu výuky programování. Zde jsou k dispozici jiná vývojová prostředí, v okamžiku psaní tohoto článku se jedná o JBuilder a některé části řešení SUN One studio. Učebna je v budově CIV, konkrétně UI312.

2.4.5 Vývoj apletů

K vytváření Java apletů je nejvhodnější použít prostředí Elipse, které označí případné chyby v kódu a dovoluje samozřejmě vytvářený aplet v rámci svého prostředí i spustit. Pokud chcete aplety vytvářet a ladit jen s využitím internetového prohlížeče, použijte v prostředí ORION LINUX prohlížeč Opera. Prohlížeč Mozilla prozatím zobrazení apletů nepodporuje (což by se mělo změnit s přechodem na verzi J2SE 5.0). V prostředí ORIONXP podporují aplety prohlížeče Internet Explorer i Mozilla.

2.4.6 Dokumentace

Dokumentace k J2SE je k dispozici nejen online na "standardní" adrese u společnosti SUN `http://java.sun.com/j2se/1.4.2/docs/index.html`, ale i v prostředí ORION LINUX prostřednictvím linku `$JAVA_HOME/docs/`.

Manuálové stránky k příkazům `java`, `javac`, `javadoc` a dalším jsou přístupné programem `man` tak, jak je to v linuxovém světě běžné.

Prostředí Eclipse má integrovaný vlastní systém nápovědy a dokumentace.

2.5 Řešení problémů při domácí instalaci JDK

Pokud jste si nainstalovali Javu (dostupná na adrese `http://www.javasoft.com/`) na svůj domácí počítač a systém hlásí, že nelze spustit překladač, ujistěte se, že máte v systémové proměnné path nastavenu cestu k adresáři, ve kterém se soubor `javac.exe` nachází (ve WinXP možno nastavit přes **Ovládací panely > Systém > Upřesnit > Proměnné prostředí**). Alternativou je překládat zdrojový kód s využitím úplné cesty, například

```
c:\Program Files\Java\jdk1.4.2\bin\javac Prvni.java
```

Dále se v případě problémů ujistěte, že máte nadefinovánu systémovou proměnnou *classpath* a že je v ní přiřazena hodnota "." reprezentující aktuální adresář, aby překladač a interpreter nehlásili chybu při pokusu o nalezení vstupních souborů. Samozřejmě i zde se nabízí alternativa překládat a spouštět program se zadáním úplné cesty v proměnné *classpath*, například

```
e:\prvni>java -classpath e:\prvni Prvni
```

Kapitola 3

Základy práce v Eclipse

3.1 Úvodem

Eclipse je univerzální vývojové prostředí primárně určené pro programovací jazyk Java a je volně k dispozici na adrese `http://www.eclipse.org`. Zde je také možné nalézt mnoho dalších informací, které se tohoto prostředí týkají, diskusní fórum, do kterého přispívají i samotní vývojáři tohoto prostředí a samozřejmě i množství plugin modulů, které umožňují rozšířit prostředí Eclipse o další vlastnosti (například o editor a překladač programovacích jazyků Pascal, C, Python a dalších). Toto prostředí se neustále vyvíjí a přibližně každého půl roku je uvolněna nová verze, která samozřejmě obsahuje nejen nové funkce, ale i opravu případných chyb, které se v předchozích verzích vyskytovaly. Eclipse je k dispozici nejen pro operační systém Windows, ale i pro Linux, Solaris a mnoho dalších.

3.2 První spuštění

Po spuštění Eclipse se zobrazí uvítací obrazovka (viz Obr. 3.1). Kliknutím na ikonu Go To The Workbench vstoupíte do pracovního prostředí, ve kterém bude standardně zobrazena základní perspektiva Resource Perspective, jež obsahuje pohledy *Navigator*, *Tasks* a *Outline* (viz Obr. 3.2).

Každá perspektiva (perspective) se skládá z pohledů (views) a z editorů (editors) umístěných na pracovní ploše. Perspektiva definuje obsah a uspořádání pracovní plochy. Jednu perspektivu můžeme mít svým uspořádáním a složením uzpůsobenu pro vytváření Java projektů a další můžeme mít uzpůsobenu pro jiné účely. Název aktivní perspektivy je zobrazen v nadpise hlavního okna prostředí Eclipse a její aktivní (stlačená) ikona je umístěna v pravém horním toolbaru pracovní plochy současně s ikonami zastupujícími již případně vytvořené perspektivy (viz Obr. 3.2). Je zde také přítomna ikona (viz Obr. 3.2 - Ikona 1) sloužící k vytvoření nové perspektivy.

Pohled (view) je vizuální komponenta umístěná na pracovní ploše. Typicky slouží k zobrazení hierarchie a závislostí mezi daty, k otevírání editorů souborů nebo k zobrazení vlastností aktivního editoru. Změny provedené v pohledu zobrazujícím data se okamžitě projeví i v editorech a naopak.

Editorem rozumíme opět vizuální komponentu na pracovní ploše. Obvykle umožňuje editovat, prohlížet a samozřejmě ukládat data různého typu. Editory a pohledy mohou být aktivní nebo neaktivní, ale vždy může být aktivní pouze jeden pohled nebo editor. Ten má potom zvýrazněnu horní část svého okna světle modrou barvou (může se u různých verzí lišit). Na aktivní okno se vztahují společné operace typu vyjmout (cut), kopírovat (copy) nebo vložit (paste) dostupné z hlavního programového menu. Aktivní okno editoru určovalo u dří-

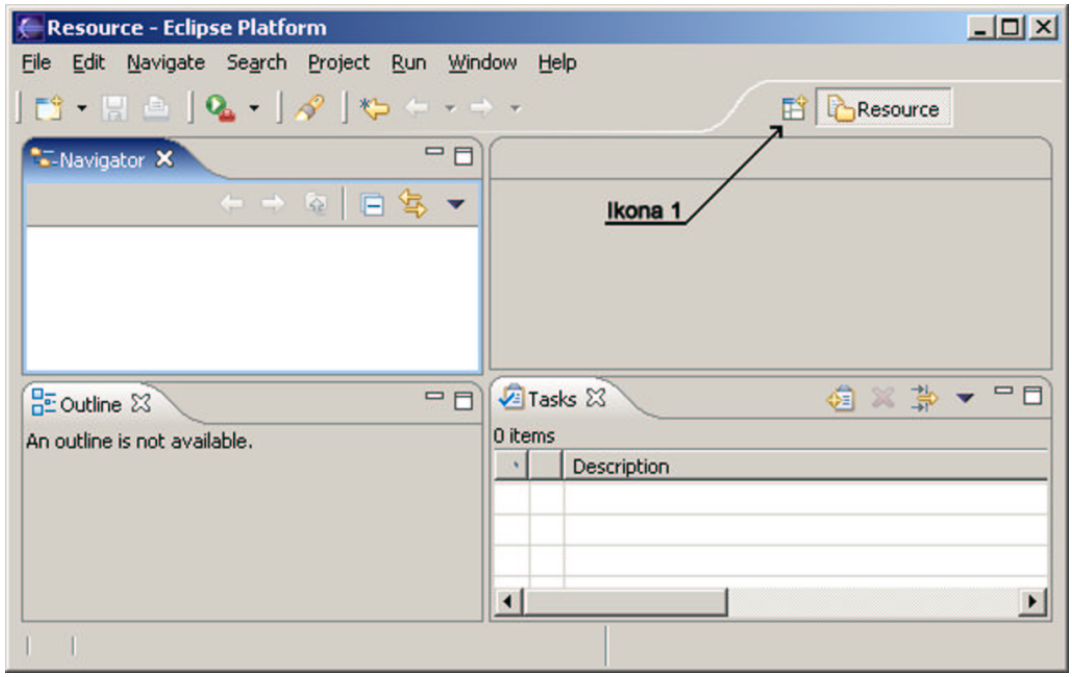
Obrázek 3.1: První spuštění prostředí Eclipse

vějších verzí prostředí Eclipse obsah informačního řádku (status line), který se nacházel v dolní části pracovního prostředí. Sloužil hlavně k zobrazení chybových hlášení při překladu Java kódu, ale v novějších verzí byl nahrazen pohledem *Problems*.

3.3 EDITORY

V závislosti na typu souboru, který chceme editovat, je zobrazen odpovídající typ editoru. Ten pro daný typ souboru disponuje specifickými funkcemi. Prostředí Eclipse obsahuje editory pro mnoho typů souborů, například při editaci souboru s příponou .java, je použit editor disponující funkcemi usnadňujícími editaci Java kódu. Název editovaného souboru je zobrazen na záložce v horní části editoru. Předchází-li názvu souboru znak '*', znamená to, že editor obsahuje neuložené změny.

V závislosti na typu aktivního editoru se v hlavním menu prostředí (menubar) a hlavní nástrojové liště (toolbar) umístěné v horní části pracovního prostředí zpřístupní operace aplikovatelné na aktivní editor. Zároveň se znepřístupní operace aplikovatelné na pohledy. Je samozřejmě možné mít otevřeno více editorů a editovat více souborů i různých typů. Pokud je otevřen soubor s příponou, ke které nemá Eclipse asociován žádný editor, je soubor otevřen a editován implicitně textovým editorem.

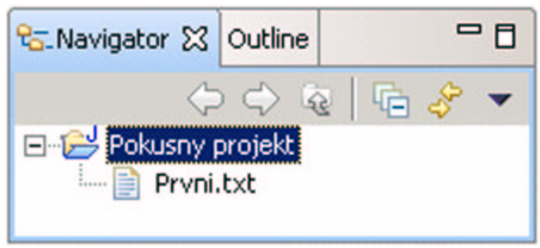


Obrázek 3.2: Perspektiva Resource - implicitní při prvním spuštění

3.4 POHLEDY

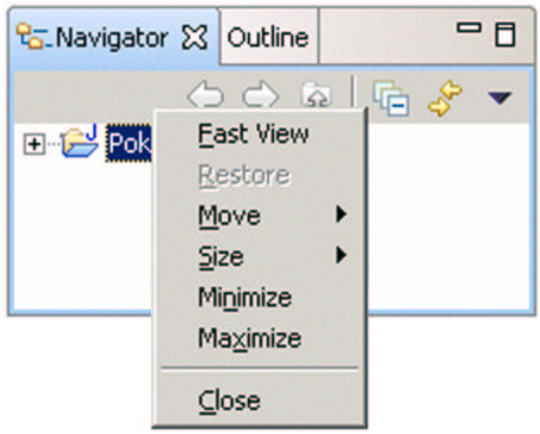
Pohledy většinou poskytují podporu editorům a zajišťují alternativní reprezentaci dat a informací, se kterými se aktuálně pracuje.

Například pohled *Navigator* (viz Obr. 3.3) zobrazuje vytvořené projekty a zdroje, se kterými se pracuje. Dvojitým kliknutím na položku typu soubor v pohledu *Navigator* se otevře odpovídající editor. Vpravo na Obr. 3.3 je zobrazena záložka dalšího pohledu *Outline*, který není aktivní.



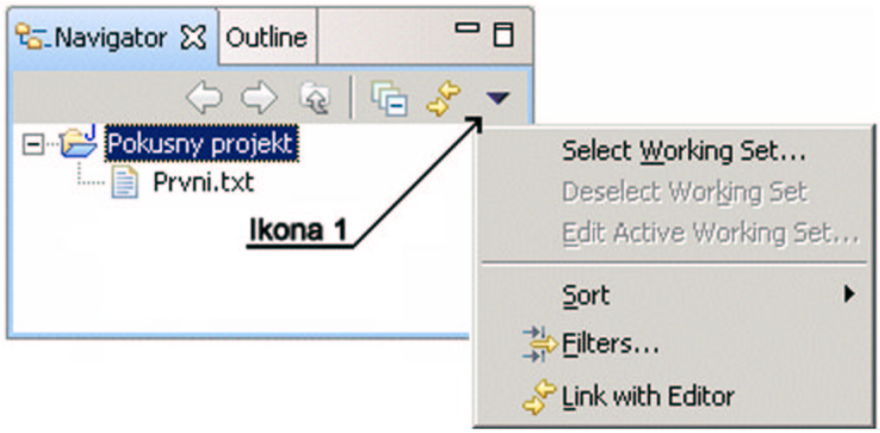
Obrázek 3.3: Pohled *Navigator*

Každý pohled disponuje dvěma menu. První menu lze vyvolat stiskem pravého tlačítka myši na celé horní ploše pohledu (modrá + šedá část). Obsahuje většinou funkce vztahující se k umístění a velikosti konkrétního pohledu na pracovní ploše prostředí (viz Obr. 3.4).



Obrázek 3.4: Menu umístěné v levém horním rohu pohledu

Druhé menu je dostupné v pravém horním rohu (viz Obr. 3.5 - Ikona 1). Obsahuje většinou funkce, které se aplikují na celý obsah pohledu. Samozřejmě jsou zde i další ikony zastupující pohyb po položkách zobrazených v pohledu.

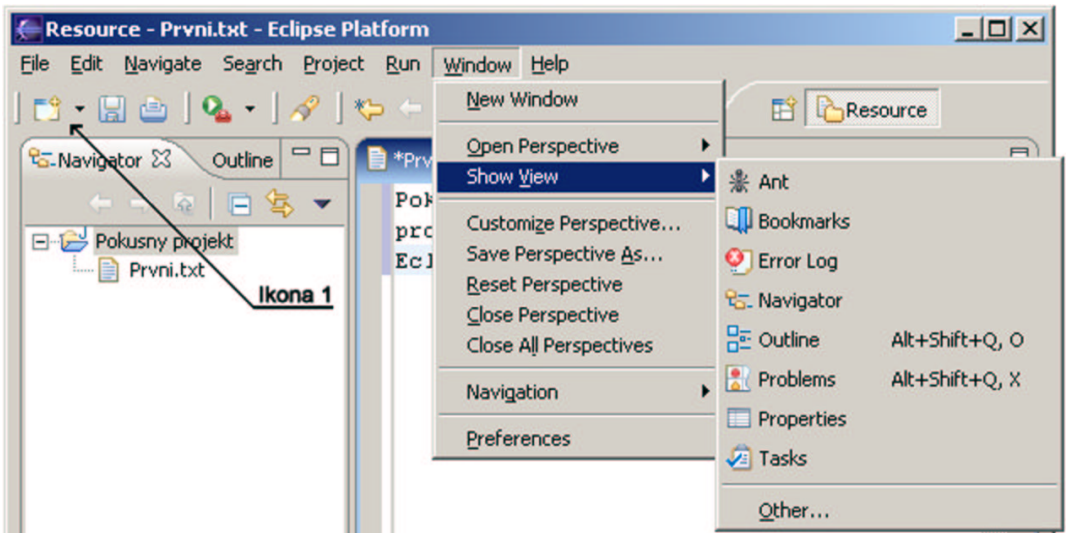


Obrázek 3.5: Menu umístěné v pravém horním rohu pohledu

Pravé tlačítko myši stisknuté na ploše některých pohledů může aktivovat pomocné pop-up menu, jehož položky vždy zastupují funkce dostupné i z hlavního menu prostředí (menubar). Dvojitým kliknutím na položku v pohledu *Navigator* se otevře příslušný editor. V případě souboru Prvni.txt (viz Obr. 3.5) se otevře textový editor umožňující obsah tohoto souboru editovat.

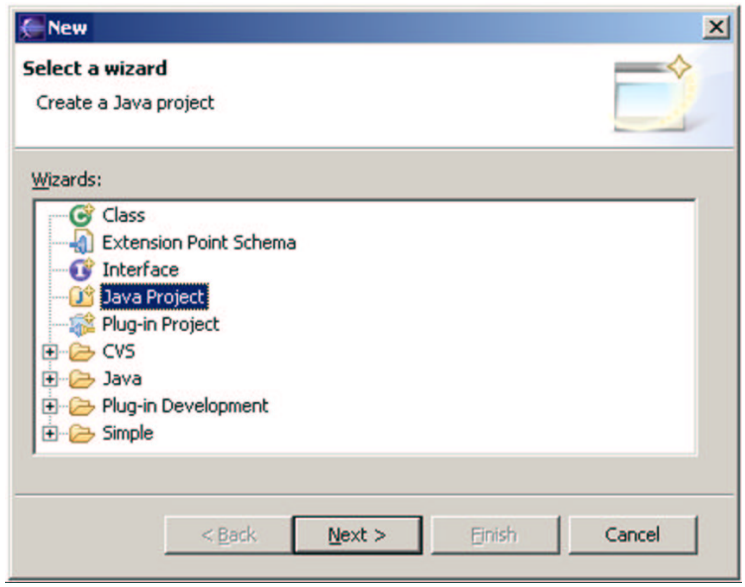
Užitečná operace se ukrývá v hlavním menu **Window > Reset Perspektive**, která zajistí zobrazení implicitních pohledů pro danou perspektivu. V počátcích práce s prostředím a v průběhu experimentování se tato operace poměrně hodí.

Pohled *Navigator* není jediný, který máme k dispozici, řadu dalších pohledů je možné nalézt v menu **Window > Show View** a také v přidruženém podmenu **Other...** (viz Obr. 3.6). Pomocí pohledů je možné si vytvořit vlastní pracovní prostředí, které bude zobrazovat jen ty informace, které uživatel požaduje. K dispozici jsou zde například pohledy *Console* reprezen-



Obrázek 3.6: Menu obsahující všechny pohledy

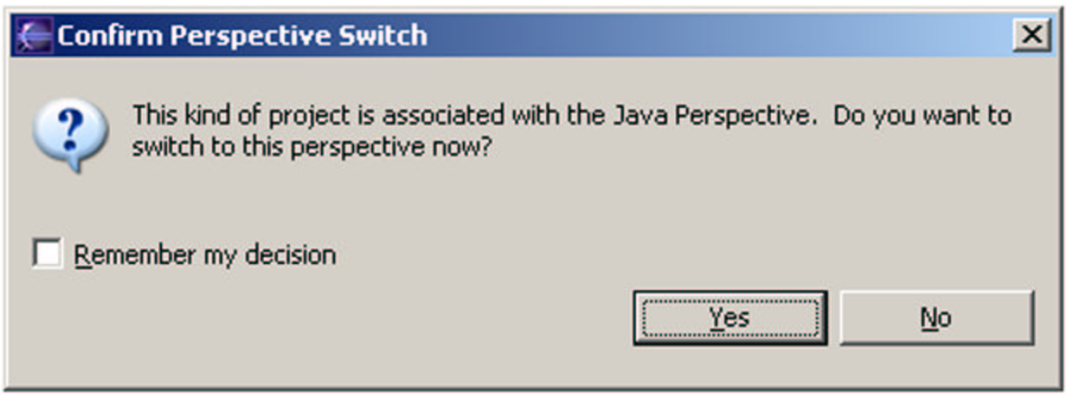
tující konzolový výstup, pohled *Search* umožňuje vyhledávání jak řetězců v souborech, tak i názvů souborů kromě jiného i pomocí regulárních výrazů, nebo pohled *Outline* v případě editování Java kódu zobrazuje jeho strukturu včetně balíků, tříd a jejich metod.



Obrázek 3.7: Dialog pro vytvoření nového projektu

3.5 PRVNÍ JAVA PROJEKT

Nové projekty, složky nebo soubory je možné vytvořit různými způsoby, my si ukážeme jeden základní. V levém horním rohu pracovního prostředí (v hlavním toolbaru) klikneme na ikonu 1 - viz Obr. 3.6. Zobrazí se dialog znázorněný na Obr 3.7.
Vybereme položku Java Project. Klikněte na tlačítko *Next*, zadejte název projektu, na který budete dotázáni (nazvěme jej třeba "Prvni") a klikněte opět na tlačítko *Next*. Zobrazí se dialog umožňující nastavit, jaké knihovny mají být importovány (standardně jsou importovány knihovny aktuální verze JDK), jaké projekty mají být zahrnuty do tohoto nově vytvářeného projektu a spousta dalšího nastavení. Pro začátek doporučuji nic neměnit a rovnou kliknout na tlačítko *Finish*. Objeví se pravděpodobně dialog zobrazený na Obr. 3.8.

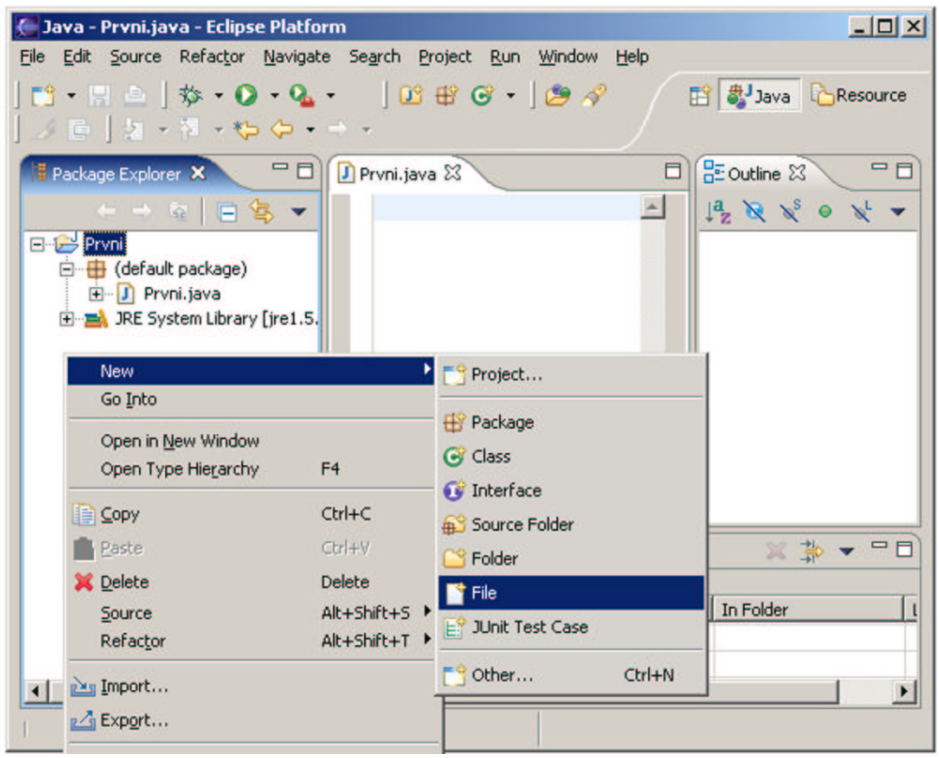


Obrázek 3.8: Dotaz zobrazený při vytváření Java projektu z Resource Perspective

Dialog zobrazený na Obr. 3.8 upozorňuje, že vytvářený Java Project je asociován s perspektivou Java Perspective a ptá se, zda se do této perspektivy chceme přepnout. Protože máme momentálně otevřenou perspektivu Resource Perspective, doporučuji kliknout na tlačítko **Yes**. Objeví se nová pracovní plocha již obsahující pohledy *Outline*, *Problems* (obsahuje seznam a popis chyb při editaci Java kódu) a *Package Explorer*, který zobrazuje ke každému vytvořenému projektu seznam balíků, ze kterých se skládá a ke každému balíku seznam souborů, které obsahuje. Pochopitelně si takovouto plochu můžeme vytvořit sami i v implicitní perspektivě Resource Perspective aktivováním potřebných pohledů, ale takto máme ušetřenou práci a máme vytvořenu novou pracovní plochu určenou pro vytváření Java projektů. Ostatní pracovní plochy (perspektivy) můžeme používat pro jiné účely, což zpřehlední celkovou práci s prostředím. Máme vytvořen Java projekt a nyní v projektu vytvoříme soubor. V pohledu *Package Explorer* vybereme námi vytvořený projekt "Prvni", klikneme pravým tlačítkem myši a vybereme **New > File** (viz Obr. 3.9). Objeví se dialog, do kterého zadáme název souboru (nazveme jej třeba "Prvni.java") a potvrdíme stlačením tlačítka **Finish**. Automaticky se otevře editor Java kódu (v jeho horní záložce je uveden název editovaného souboru).

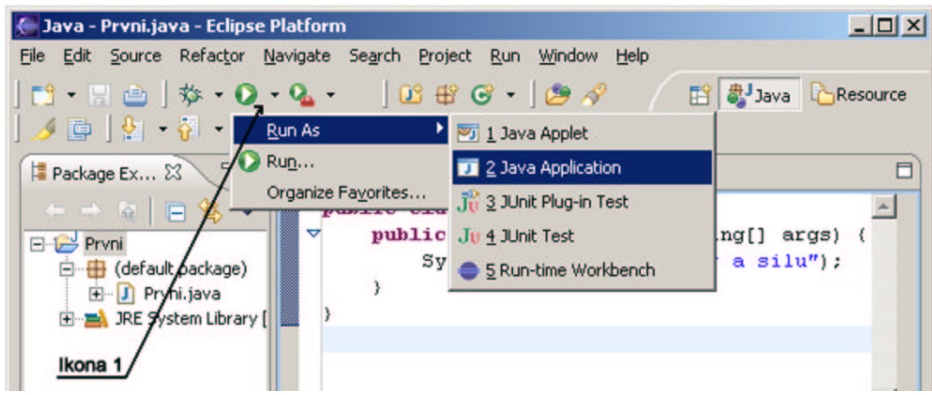
Jak je vidět na Obr. 3.9, v projektu je možné přímo vytvořit třídu (class), rozhraní (interface) nebo například balík (package). V editoru napíšeme jednoduchý kód, třeba tento:

```
public class Prvni {
    public static void main(String[] args) {
        System.out.println("Zdar a silu");
    }
}
```



Obrázek 3.9: Vytvoření souboru do projektu

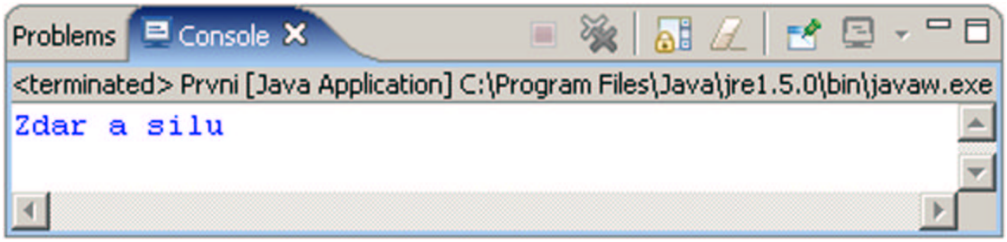
V hlavním toolbaru umístěném v horní části pracovního prostředí klikneme na šipku vedle ikony č. 1 – viz Obr. 3.10. Vybereme volbu **Run As > Java Application** (viz Obr. 3.10). Pokud jste soubor předem neuložili, budete vyzváni k jeho uložení.



Obrázek 3.10: Spuštění Java kódu

Program je přeložen a spuštěn. Protože zapisuje do konzole, automaticky se otevře pohled *Console*, ve kterém se zobrazí výstup programu (viz Obr. 3.11). V pravé horní části

pohledu *Console* jsou k dispozici ikony umožňující zastavit probíhající program, vrátit se k výstupu dříve spouštěných programů, uzamknout nebo vymazat výstup konzole a samozřejmě uzavřít konzoli. Pokud se v kódu programu vyskytne chyba, otevře se pohled *Problems* s popisem nalezených chyb.



Obrázek 3.11: Výstup programu

3.6 PARAMETRY PŘÍKAZOVÉ ŘÁDKY

Obsah souboru `Prvni.java` modifikujeme následujícím způsobem:

```
public class Prvni {
    public static void main(String[] args) {
        System.out.println(args[0]);
    }
}
```

Program spustíme stejně jako v předchozím případě, jen místo volby **Run As > Java Application** zvolíme volbu **Run...**. Objeví se dialog umožňující nadefinovat jméno projektu, který má být spuštěn, jméno třídy, jejíž metoda `main()` má být vyvolána, v dalších záložkách je možné nadefinovat cestu ke třídám (záložka **Classpath**), lze nadefinovat, které JRE má být pro překlad použito (pokud jich je v systému instalováno více - záložka **JRE**) a také je možné v záložce **Arguments** nadefinovat parametry příkazové řádky. Do pole **Program arguments:** zadáme například řetězec "Zdar a silu" (včetně uvozovek) a klikneme na tlačítko **Run**. Opět dostaneme stejný výstup jako v předchozím případě (viz Obr. 3.11). Pokud bychom chtěli zadat více vstupních parametrů, stačí je oddělit mezerou.

3.7 IMPORT JIŽ EXISTUJÍCÍHO PROJEKTU

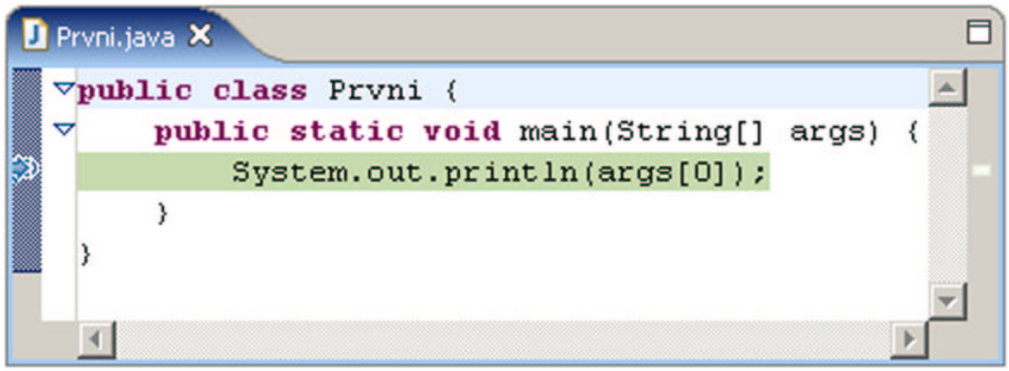
Pokud již máme vytvořen soubor nebo soubory obsahující Java kód, lze je jednoduchým způsobem přidat do našeho projektu v prostředí Eclipse. Jak již bylo řečeno v úvodu, v adresáři *\eclipse\workspace* jsou umístěny adresáře představující naše vytvořené projekty. Stačí tedy jednoduše již dříve vytvořené soubory nahrát do požadovaného adresáře a v pohledu *Package Explorer* po kliknutí pravým tlačítkem na požadovaný projekt vybrat volbu **Refresh**. Přidané soubory se potom objeví jako součást projektu. V případě, že přidané soubory obsahují definici balíku (package), je na tuto skutečnost samozřejmě brán zřetel a v projektu se vytvoří nový balík, v němž budou tyto soubory obsaženy. Aby bylo vše v pořádku, je potřeba pro tyto soubory v adresáři projektu vytvořit adresářovou strukturu, která odpovídá názvu balíku v souladu s pravidly programovacího jazyka Java.

Pokud by bylo potřeba importovat již existující projekt prostředí Eclipse, případně jiný typ projektu, je k dispozici průvodce importem dostupný přes hlavní menu **File > Import**.

Stejně jako je k dispozici průvodce importem, je k dispozici i průvodce exportem **File > Export**, který umožňuje export do jar souboru, do zip archívu nebo například umožňuje generovat dokumentaci pomocí nástroje Javadoc.

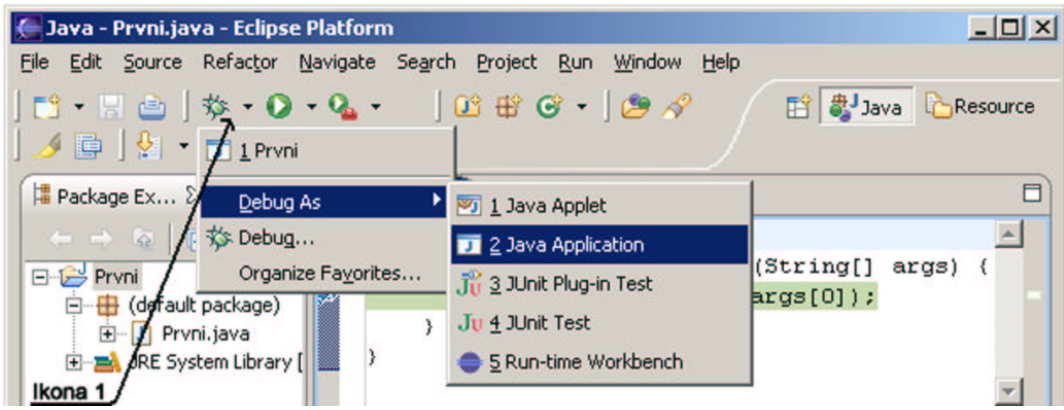
3.8 KROKOVÁNÍ PROGRAMŮ

Pokud chceme ladit náš program v jazyce Java, je nejprve potřeba do našeho kódu umístit body přerušení běhu programu (breakpoints), na jejichž pozici dojde k zastavení běhu programu a které umožní dále program krokovat po jednotlivých příkazech. To lze provést jednoduše dvojitým kliknutím levým tlačítkem myši v levé krajní části editoru kódu, kde se následně zobrazí modrá kulička (Obr. 3.12).



Obrázek 3.12: Umístění breakpointu v editoru kódu

V hlavním toolbaru umístěném v horní části pracovního prostředí klikneme na šipku vedle ikony č. 1 – viz Obr. 3.13. Vybereme volbu **Debug As > Java Application** (viz Obr. 3.13).

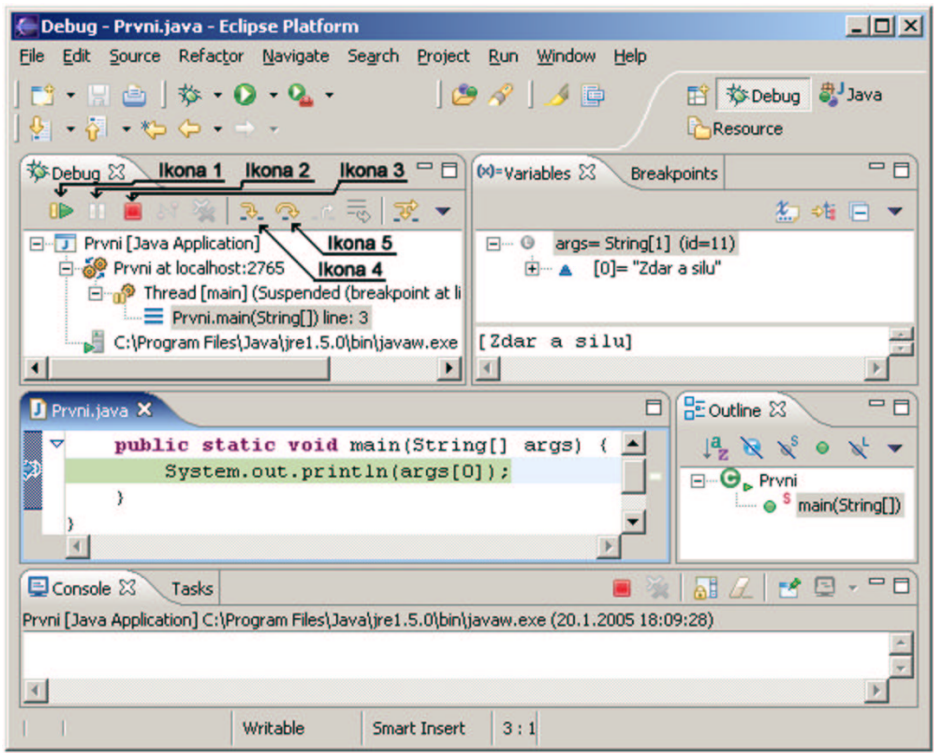


Obrázek 3.13: Spuštění krokování Java kódu

Po aktivaci této volby bude uživatel pravděpodobně dotázán, zda chce otevřít novou perspektivu Debug Perspective obsahující všechny potřebné pohledy pro ladění a editor Java kódu. Samozřejmě je vhodné nabízenou možnost přijmout. V pravé horní části pracovní plochy je umístěn pohled *Variables* (viz Obr. 3.14), který obsahuje seznam všech aktuálních

proměnných programu včetně detailního popisu jejich momentálních hodnot. Vlevo nahoře je potom umístěn pohled Debug (viz Obr. 3.14), který zobrazuje všechny aplikace, které jsou v současné chvíli prostředím Eclipse spuštěny, případně zobrazuje aplikace, které již svou činnost ukončily. V pravé horní části tohoto pohledu jsou k dispozici ikony pro pokračování běhu programu (Resume) , pro jeho pozastavení (Suspend) nebo pro jeho ukončení (Terminate) , dále jsou zde k dispozici ikony pro zanoření (Step Into) , pro vykonání příkazu v kódu (Step Over) a další.

Aktuální pozice v kódu programu při krokování je znázorněna v editoru kódu zvýrazněným pozadím textu aktuální řádky a šipkou v levé části editoru (viz Obr. 3.14).



Obrázek 3.14: Perspektiva Debug sloužící ke krokování Java aplikací

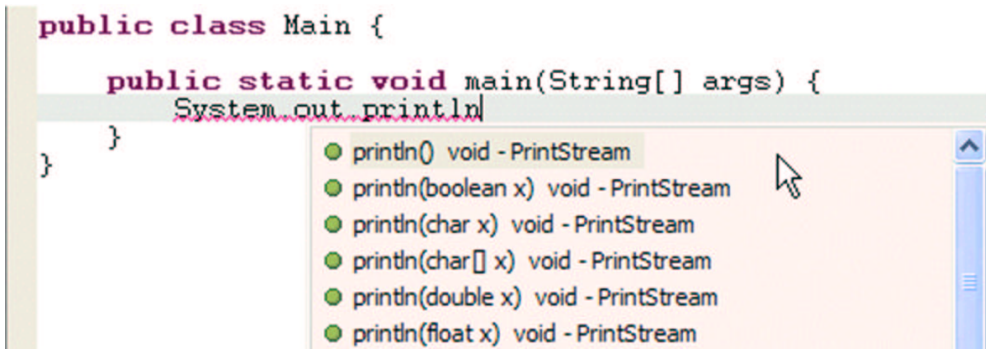
Jak je vidět z Obr. 3.13, je kromě čistokrevných Java aplikací možno krokovat a ladit i Java aplety nebo vlastní pluginy, kterými je možné prostředí Eclipse rozšířit o další funkčnost. Ladění takových pluginů potom probíhá tím způsobem, že je spuštěno druhé prostředí Eclipse, které již obsahuje zakomponovaný vytvářený plugin a první prostředí krokuje a ladí plugin v druhém spuštěném prostředí.

3.9 VLASTNOSTI EDITORU JAVA KÓDU

Java editor Eclipse disponuje mnoha funkcemi, které usnadňují, zrychlují a zpříjemňují práci programátora:

- **Asistent obsahu** (Content assist)

  Asistent obsahu poskytuje uživateli seznam automatických doplnění, které pro daný případ přicházejí v úvahu. Tento seznam je automaticky aktivován zadáním znaku '.' nebo ho lze kdykoli vyvolat stiskem kláves **Ctrl+Space**. Nastavení je možné upravit v menu **Edit > Content Assist**.

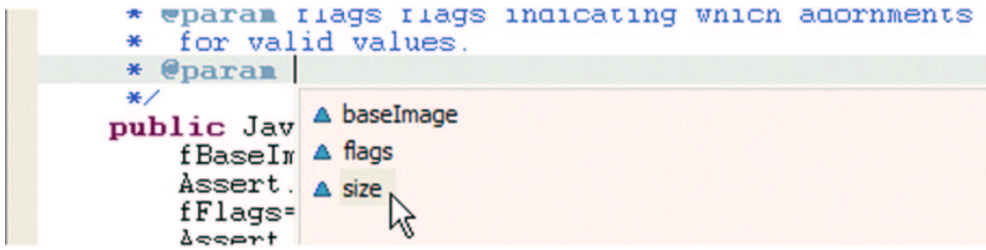


- **Automatické zajištění potřebných importů** (Organize Imports)

  Prostředí Eclipse nabízí automatické přidání potřebných importů do vytvářeného Java kódu. Tato funkčnost je dostupná z hlavního menu **Source > Organize Imports** nebo přes pravé tlačítko myši. Je k dispozici nejen pro jednotlivé soubory, ale i pro celé balíky nebo složky.

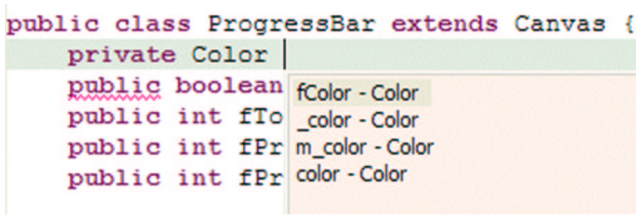
- **Asistent obsahu pro Javadoc komentáře** (Content Assist in Javadoc comments)

  Asistent obsahu je dostupný i pro Javadoc komentáře.



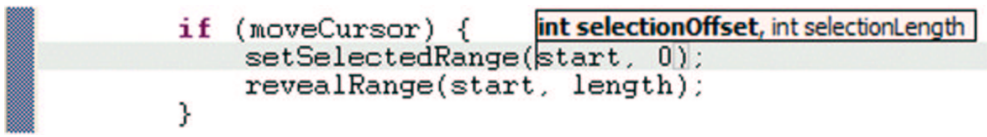
- **Asistent obsahu pro proměnné, parametry metod a názvy polí** (Content assist for variable, method parameter and field name completions)

  Asistent obsahu může být použit i pro zrychlení vytváření polí, parametrů metod a lokálních proměnných nastavením kurzoru za název nově vytvářeného typu a stlačením **Ctrl+Space**. Asistent obsahu pro jednotlivé datové typy nabídne předem nadefinovaná jména. Nadefinování vlastních názvů, prefixů nebo sufixů proměnných je možné provést v menu **Window > Preferences > Java > Code Generation > Names**.



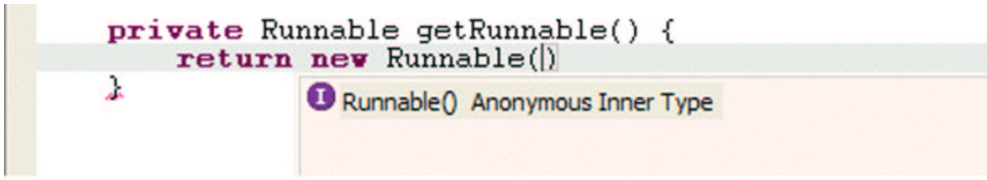
- **Plovoucí nápověda** (Parameter Hints)

  Nastavením kurzoru na argument metody se automaticky zobrazí plovoucí nápověda obsahující všechny parametry metody včetně popisu jejich typu. Nápovědu je také možné vyvolat kombinací kláves **Ctrl+Shift+Space**. Nastavení je možné upravit v menu **Edit > Parameter Hints**.



- **Asistent obsahu pro anonymní třídy** (Content assist on anonymous classes)

  Asistent obsahu také poskytuje automatickou pomoc pro anonymní třídy nastavením kurzoru za otevírací závorky vytvářené instance třídy nebo stlačením kombinace kláves **Ctrl+Space**. Nastavení je možné upravit v menu **Edit > Content Assist**. Asistent umožňuje automaticky vytvořit tělo anonymní třídy, které bude zahrnovat všechny metody, které je nutné implementovat.



- **Nahrazení nebo vložení Asistentem obsahu** (Inserting and replacing in code assist)

  Pokud je aktivován Asistent obsahu nad existujícím identifikátorem, je možné zvoleným doplněním identifikátor nahradit (overwrite) nebo provést jen prosté vložení (insert). Nastavení je možné změnit v menu **Window > Preferences > Java > Editor > Code Assist**. Volbu je také možné dočasně změnit v již aktivovaném Asistentu obsahu při výběru doplnění stlačením klávesy **Ctrl**.

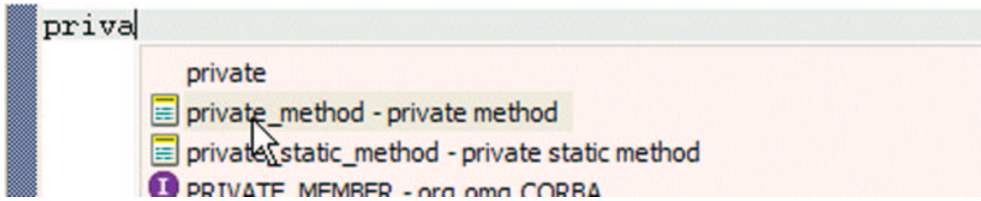
- **Změna v kódu pomocí Drag & Drop** (Use Drag & Drop for refactoring)

  Jednotlivé soubory s Java kódem lze jednoduše přenést pomocí Drag&Drop (například v pohledu **Package Explorer**) mezi různými programovými balíky (packages), přičemž všechny chybějící importy budou automaticky přidány a všechny reference budou upraveny tak, aby bylo vše v pořádku.

- **Použití návrhových vzorů k vytváření metod** (Use Templates to create a Metod)

  V menu **Preferences > Java > Editor > Templates** lze nalézt předdefinované vzory. Je zde samozřejmě možné nadefinovat si i vzory vlastní. Návrhové vzory jsou zobrazeny

společně v nabídce Asistentu obsahu (**Ctrl+Space**). Klávesou **Tab** je možné se přepínat mezi zadávanými hodnotami (návratovým typem, názvem a argumenty).

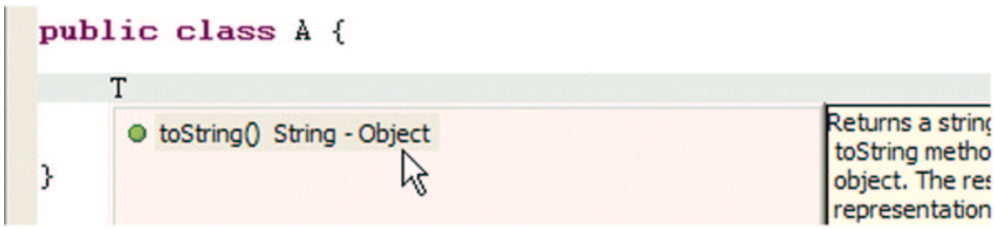


Například vzor pro zpětné procházení pole by mohl vypadat následovně:

```
for (int ${index} = ${array}.length - 1; ${index} >= 0; ${index}--) {
    ${cursor}
}
```

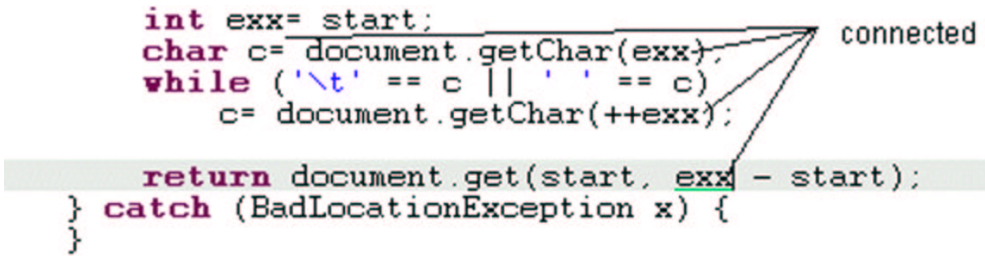
- **Použití Asistentu obsahu k překrytí metody** (Use Content Assist to override a Metod)

  Asistent obsahu (**Ctrl+Space**) vyvolaný uvnitř třídy, kam má být přidána nová metoda, nám nabídne hlavičky všech metod, které mohou být překryty a automaticky vytvoří tělo zvolené metody.



- **Lokální přejmenování** (Local rename)

  K rychlému přejmenování, které nevyžaduje úplnou analýzu závislostí v ostatních souborech projektu, je možné použít funkci lokálního přejmenování (Local rename). Volbu vyvoláme nastavením kurzoru na identifikátor proměnné, metody nebo typu a stiskem **Ctrl+1**. Editor se přepne do spojeného módu (linked mode) a všechny provedené změny se okamžitě projeví i v ostatních výskytech modifikované proměnné, metody nebo typu.

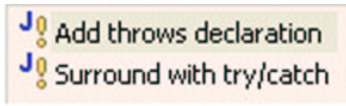


- **Použití rychlé opravy k ošetření výjimky** (Use Quick Fix to handle exceptions)

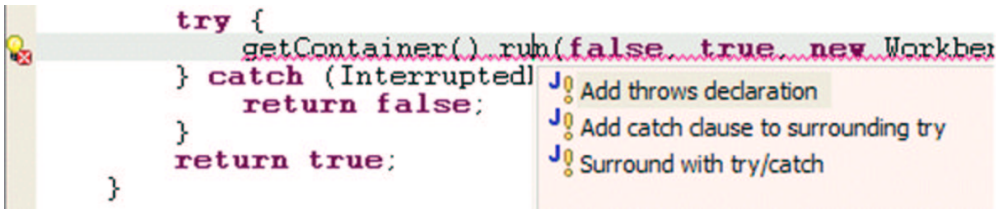
  Neošetřené výjimky (Unhandled Exceptions) jsou detekovány (stejně jako veškeré ostatní chyby) již během zadávání Java kódu a jsou okamžitě označeny červeným podtr-

žením a varovným symbolem v levé krajní části editoru. Prostředí Eclipse umožňuje jednoduše výjimky ošetřit:

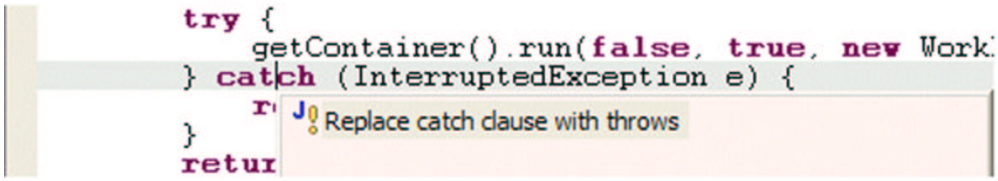
a) Nastavením kurzoru na chybový řádek a stisknutím **Ctrl+1**. Objeví se nabídka, ve které je možné si zvolit, jakým způsobem má být výjimka ošetřena.



b) Pokud chceme do try-catch bloku umístit více příkazů, stačí tyto příkazy označit a použít volbu v hlavním menu **Source > Surround With try/catch Block**.



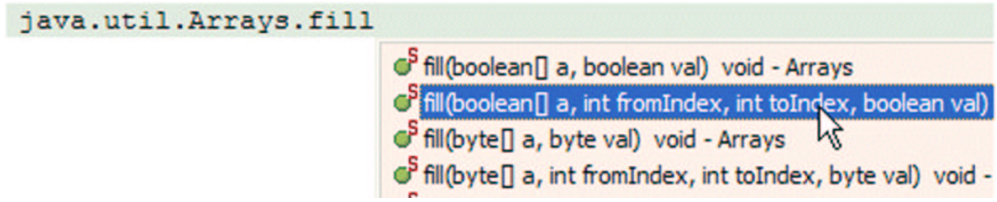
Kdykoli chceme změnit try-catch blok na throws exception, stačí nastavit kurzor na tento blok, použít kombinaci **Ctrl+1** a vybrat příslušnou volbu. Volbu je také možné



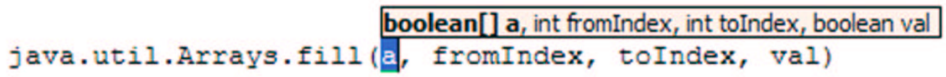
vyvolat v hlavním menu **Edit > Quick Fix**.

- **Asistent obsahu umožňuje automaticky vkládat názvy argumentů** (Code assist can insert argument names automatically)

  V hlavním menu **Java > Editor > Code Assist** lze zaškrtnout volbu **Fill Argument Names on method completion**. Pokud si potom podobně jako v uvedeném příkladu v aktivním Asistentu obsahu zvolíme pro metodu `java.util.Arrays.fill()` druhý řádek, dojde tím k vložení jak hlavičky metody `fill()`, tak i všech názvů jejích argumentů.

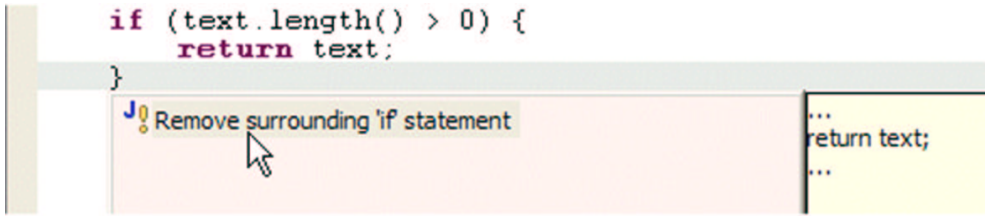


Potom lze použít klávesu **Tab** k přepínání mezi jednotlivými vloženými názvy argumentů.



- **Odstranění okolních programových konstrukcí** (Remove surrounding statement)

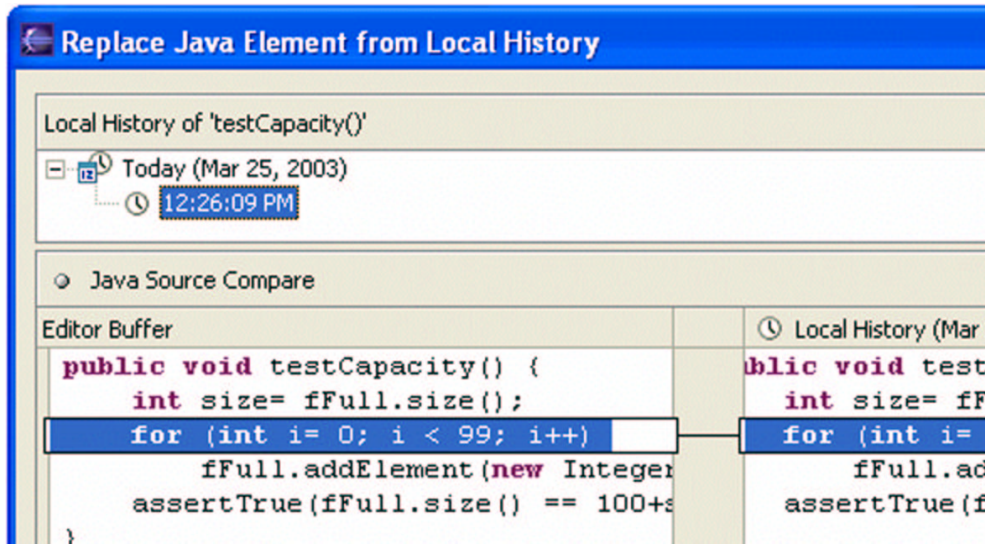
  K odstranění okolních programových konstrukcí nebo bloků stačí nastavit kurzor na otevírací nebo uzavírací složenou závorku daného bloku a stlačit klávesy **Ctrl+1**. Nastavení je možné upravit v menu **Edit > Quick Fix**.



- **Návrat k předchozím verzím pomocí Lokální historie změn** (Use the local history to revert back to a previous edition of a metod)

  Během editování kódu dochází k průběžnému ukládání změn do Lokální historie. Místo celkového obsahu souboru se ukládají jen jednotlivé provedené změny.

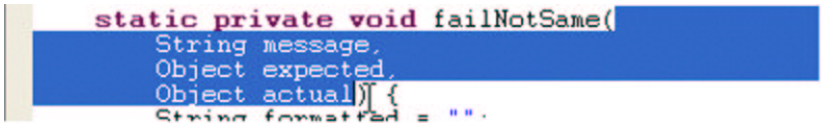
  Pro návrat k předchozímu kódu se stačí nastavit kurzorem na požadovaný úsek, kliknout pravým tlačítkem myši a vybrat v menu volbu **Local History > Replace With**.



- **Nalezení odpovídající závorky** (Find the matching bracket)

  K nalezení uvozujících závorek bloku stačí vybrat jednu ze závorek (typu {}, () nebo []) a druhá odpovídající závorka se automaticky zvýrazní. Nastavení je možné upravit v hlavním menu **Navigate > Go To > Matching Bracket**.

Dále je možné označit celý blok příkazů dvojitým kliknutím před otvírací nebo uzavírací závorkou uvozující příslušný blok.



3.10 INSTALACE ROZŠIŘUJÍCÍCH MODULŮ, DOMÁCÍ INSTALACE

Budete-li někdy potřebovat nainstalovat celé prostředí Eclipse, je to velmi jednoduché. Stačí pouze rozbalit stažený zip archiv (z adresy `http://www.eclipse.org`, velikost je asi 85 MB) obsahující adresář *\eclipse* a nakopírovat jej kamkoli na pevný disk, nejlépe asi do adresáře *\program files* (velikost rozbaleného zip archívu je kolem 100 MB). V adresáři *\eclipse* se nachází soubor *eclipse.exe*, kterým je možné aplikaci Eclipse spustit. Ke spuštění je potřeba mít v systému nainstalováno JDK a JRE (nejlépe aktuální verze), které jsou volně k dispozici na adrese `http://www.javasoft.com`.

Po prvním spuštění je v adresáři *\eclipse* vytvořen nový pracovní adresář *\workspace*, který bude obsahovat všechny uživatelem vytvořené projekty (uživatel je na název a umístění tohoto adresáře dotázán - doporučujeme ponechat výchozí nastavení).

Na adrese `http://www.eclipse.org/tools/index.html` je dále možné získat nově vyvinuté nástroje pro prostředí Eclipse. Zajímavé jsou například následující:

- **VE (Eclipse Visual Editor)** Zahrnuje v sobě prostředí pro vizuální vytváření GUI s podporou Swing/JFC i SWT
- **UML2** Poskytuje nástroje pro návrh modelu aplikace pomocí standardu UML 2.0.
- **C/C++ IDE** Tento nástroj v sobě zahrnuje kompletní vývojové prostředky pro návrh aplikací v jazyce C/C++. Je možné využít editor, debugger, launcher, parser, makefile generator a další služby.

Instalace nového nástroje (pluginu) je velmi jednoduchá, stačí vždy nakopírovat adresáře obsažené ve staženém zip souboru do odpovídajících adresářů v místě instalace vývojového prostředí.

3.11 ZÁVĚR

To byl tedy přehled jen těch nejzákladnějších funkcí a vlastností, kterými vývojové prostředí Eclipse disponuje. K dispozici je dále nástroj Ant, podpora práce v týmu pomocí CVS, velká podpora práce s vlákny a samozřejmě možnost dále rozšiřovat schopnosti prostředí pomocí rozšiřujících modulů typu plugin, které je možné si naprogramovat. Mnoho již hotových je však také volně k dispozici například na adrese `http://eclipse-plugins.2y.net`. Podrobný popis všech možností prostředí Eclipse lze nalézt v hlavním menu **Help > Help Contents**.

V případě potřeby (ztratíte-li se v prostředí) je možné se vždy vrátit k úvodnímu menu (viz Obr. 3.1), kde jsou k dispozici tutoriály a vzorové příklady aplikací pro prostředí Eclipse pomocí volby hlavního menu **Help > Welcome**.

Jak lze vysledovat z historie vývoje prostředí Elipse, budoucí verze budou pravděpodobně drobně graficky pozměněny, ale popsané postupy se již příliš lišit nebudou.

Kapitola 4

Nadstavby nad Eclipse

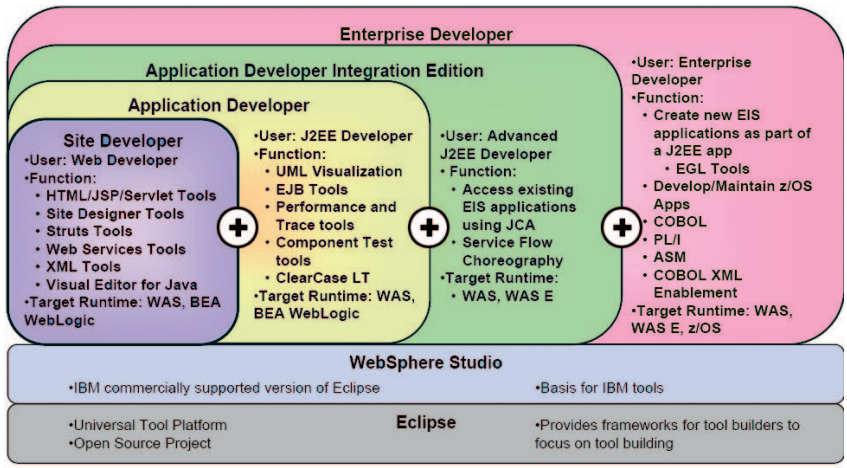
Kromě bohaté nabídky desítek rozšiřujících modulů plug-in (komerčních i opensource) pro Eclipse existuje i několik komerčních produktů, která rozšířením funkcí opensource jádra Eclipse tvoří specializovaná vývojová prostředí. Někdy se hovoří o platformě Eclipse, nad níž jsou vytvořeny specializované a komerčně podporované vývojové nástroje.

Z pohledu studenta je jistě výhodné, že se může naučit pracovat s opensource nástrojem, který získá zdarma, a například po nástupu do zaměstnání se bude v profesionálních odvozeninách Eclipse „cítit jako doma“. Odpadá tak čas na zapracování a sžití se s novým prostředím, které trvá týdny až měsíce.

V této kapitole si ukážeme dvě prostředí z dílny IBM, se kterými se na ZČU můžete potkat.

4.1 IBM WebSphere Studio

Rodina produktů IBM WebSphere studio je typickou ukázkou komerční varianty Eclipse. Dá se říci, že IBM upravila a doplnila Eclipse, aby jej mohla dodávat jako vlastní vývojové prostředí pro své produkty (zejména WebSphere Application Server, Lotus, DB2 a další). Následující obrázek ukazuje rodinu WebSphere studio a její vztah k Eclipse:



Obrázek 4.1: Rodina WebSphere Studio

Jednotlivé produkty jsou odstupňované podle zaměřené nástroje a množství zahrnutých funkcí.

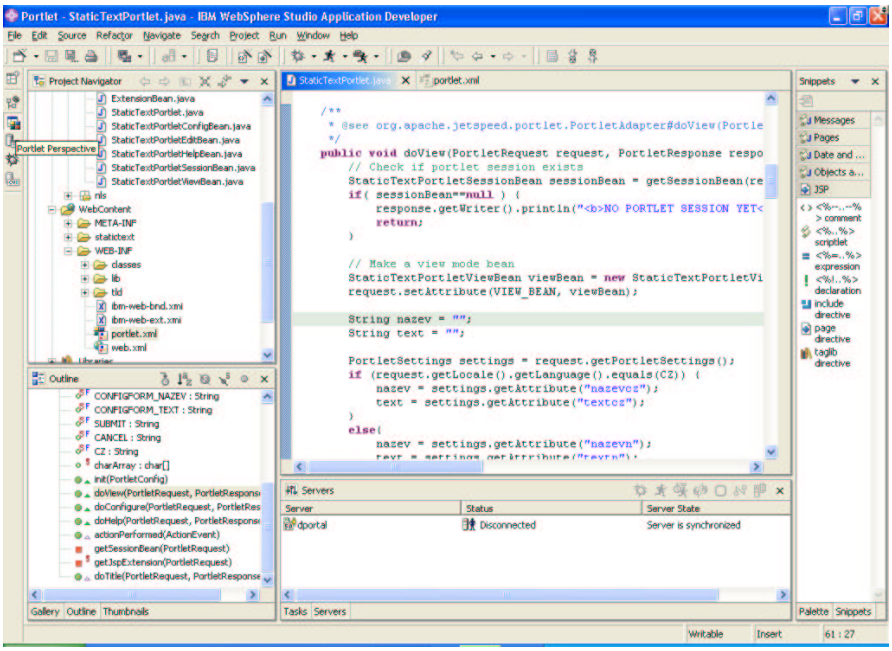
- **Site Developer – WSSD**
  Nejjednušší nástroj z rodiny je zamečen na webdesignera. Předpokládá se tvorba statických HTML stránek nebo JSP. WSSD umožňuje správu celých website (nebo chcete-li projektů), použití šablon a další nástroje podobně jako například Macromedia Dreamweaver. WSSD lze použít i k běžnému programování v jazyce Java (J2SE – Java2 standard edition).

- **Application Developer – WSAD**
  S tímto balením WebSphere studia se setkáte nejčastěji. Obsahuje v sobě WSSD plus nástroje pro J2EE (Java2 Enterprise Edition). Nástroj je zaměřen na návrh a vývoj webových nebo obecně klient/server aplikací, vývoj EJB a testování. V rámci WSAD lze spustit několik různých aplikačních serverů a v nich běžící aplikace přímo ladit.

- **Application Developer Integration Edition**
  Jedná se o ještě větší verzi WSAD obsahující nástroje pro EAI (enterprise application integration), zejména technologii JCA (Java connector architecture).

- **Enterprise Developer**
  Největší balení WebSphere studia kromě jazyka Java podporuje celé portfolio programovacích jazyků (i starších) používaných IBM mj. i na mainframech.

Na ZČU používáme WSAD s rozšířením Portal Toolkit pro vývoj nové generace IS/STAG a portletů pro univerzitní portál.

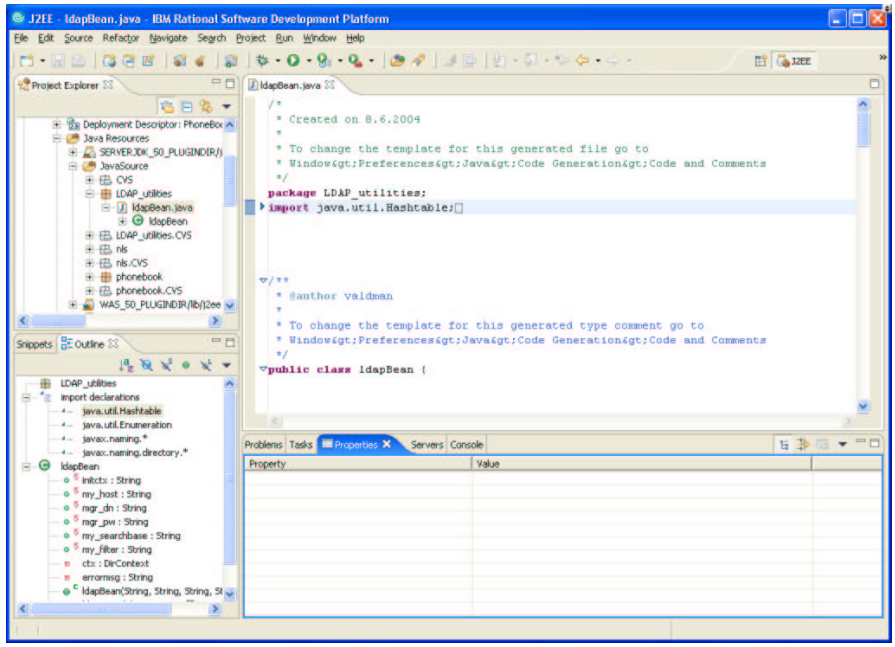


Obrázek 4.2: Portletová perspektiva WSAD

4.2 IBM RATIONAL DEVELOPMENT PLATFORM

Rational Software Development Platform je nástupcem WebSphere studia; víceméně se jedná o přeskupení produktů v rámci IBM, kde vývojové nástroje nyní patří do divize Rational, která pokrývá celou škálu produktů pro oblast analýzy, návrhu, vývoje, testování a instalace software.

V současné době je k dispozici Rational Application Developer 6.0, který od Eclipse 3 prakticky nerozeznáte, jak ukazuje následující obrázek 4.3.



Obrázek 4.3: J2EE perspektiva v Rational Application Developer

Z praktického hlediska se RAD liší od WSAD lepší podporou nových Java standardů a celé studio nyní běží nad Eclipse verze 3. Tento produkt lze pro potřeby výuky získat zdarma, a proto předpokládáme, že se v nejbližší době objeví v učebnách KIV.

4.3 DALŠÍ NÁSTROJE

Bylo by špatné, kdyby ve čtenáři vznikl dojem, že Eclipse slouží pouze k programování, jeho možnosti jsou mnohem širší. Na Eclipse se lze dívat jako na univerzální rozšiřitelný editor, kterým můžete tvořit prakticky *cokoli*. Existují desítky softwarových firem, které se specializují na tvorbu zásuvných modulů do Eclipse, například pro oblast modelování, simulací, testování ...

Jako výchozí bod pro zkoumání dalších možností Eclipse můžete použít následující webové stránky nebo si zakoupit nějakou z desítek knih:

- Opensource rozšiřující moduly plug-in
  `http://www.eclipse.org/community/osplugins.html`
- Komerční rozšiřující moduly a nadstavby
  `http://www.eclipse.org/community/commercialplugins.html`

ISBN 80-7043-357-4