

1. JAVA

Osnovni pojmovi

Java je platforma za izvršavanje programa i Java je programski jezik. Java kao platforma je dizajnirana tako da što manje zavisi od specifičnih karakteristika konkretnog računarskog sistema, odnosno, da jednom napisan i preveden program može da se izvršava na bilo kojoj platformi koja podržava Javu. Dakle, kod napisan u Java programskom jeziku se ne prevodi u izvršni kod (exe) za određeni hardver, nego u izvršni kod (bajt kod – class fajlovi) za Java Virtuelnu Mašinu (JVM). Java virtuelna mašina izvršava bajt kod na platformi na kojoj je instalirana, koristeći dostupan hardver.

Java kao programski jezik je jezik opšte namene za objektno-orientisano programiranje. Ovo „opšte namene“ znači da se može koristiti i za proceduralno programiranje, što je tema ovog kursa.

Instalacija

Java instalacija dolazi u dva oblika: JRE (Java Runtime Environment) i JDK (Java Development Kit). JRE je podskup JDK i sadrži samo alate neophodne za izvršavanje prethodno razvijenog i prevedenog Java koda (dakle, uključuje Java virtuelnu mašinu - JVM). JDK pored alata za izvršenje prevedenog Java koda (bajt koda), sadrži i alate za razvoj aplikacija pisanih u Javi (uključuje i Java kompajler).

Provera da li je postoji java na računaru. Otvorite konzolu („cmd“) i ukucajte: `java -version`

Provera da li je postoji java kompajler na računaru. Otvorite konzolu i ukucajte: `javac -version`

Ukoliko nije instaliran **Java Development Kit**, neophodno je otići na Download sekciju Oracle sajta i preuzeti odgovarajuću instalacionu verziju JDK-a. Važno je napomenuti da treba preuzeti verziju koja odgovara **vašem operativnom sistemu** (Windows, Linux, Mac OS...) i **njegovoj bitnoj verziji** (**x86** - za 32-bitni ili **x64** - za 64-bitni operativni sistem). Ispod je dato uputstvo i linkovi za preuzimanje odgovarajućih verzija JDK-a.

Za početak je neophodno da utvrdite koja je bitna verzija vašeg operativnog sistema. Na Windows operativnom sistemu ovo možete uraditi tako što u Search polje upišete Control Panel, pokrenete Control Panel, kliknete na ikonicu System i tamo pročitate podatak o verziji operativnog sistema, naveden iza stavke „System type:“.

1. Ukoliko imate 64-bitni Windows, preuzmite **JDK 17 (Java SE Development Kit 17.0.2)** sa sledećeg linka:

<https://www.oracle.com/java/technologies/downloads/#java17>

Kada pristupite stranici, treba da odaberete karticu Windows, stavku x64 Installer, pod sledećim nazivom „**jdk-17_windows-x64_bin.exe**“ (ovo je druga stavka od tri na spisku sa instalacionim verzijama, a veličina fajla je 152,43 MB). Pre preuzimanja će se možda od vas tražiti da kreirate Oracle nalog (account).

2. Ukoliko imate 32-bitni Windows, pošto JDK 11 ne postoji u verziji za 32-bitne sisteme, preuzmite **JDK 8 (Java SE Development Kit 8u321)** sa sledećeg linka:

<https://www.oracle.com/java/technologies/downloads/#java8-windows>

Kada pristupite stranici, treba da odaberete karticu Windows, stavku x86 Installer, pod sledećim nazivom „**jdk-8u321-windows-i586.exe**“ (ovo je druga stavka na spisku sa verzijama, a veličina fajla je 157,99 MB). Pre preuzimanja će se možda od vas tražiti da kreirate Oracle nalog (account).

Nakon preuzimanja potrebno je izvršiti instalaciju preuzete verzije JDK-a.

U nastavku slede primeri određenih instalacija:

< -----Windows ----->

Za Windows OS duplim klikom pokreniti instalaciju jave i pratiti instalacioni vizard. Instalacioni folder Jave za Windows operativne sisteme zavisi od njegove verzije (x86 ili x64) i najčešće je sličan sledećem:

- C:\Program Files\Java\jdk1.8.0_271- za 32 bitne sisteme ili
- C:\Program Files\Java\jdk-17.0.2 - za 64 bitne sisteme.

Za potebe ovog kursa neophodno je našem operativnom sistemu naznačiti putanju do foldera u kom je instalirana Java. Ovaj korak se na novijim verzijama Windowsa izvršava automatski pri instalaciji JDK-a, pa ga možete preskočiti dok ne instalirate Eclipse i proverite da li sve radi bez problema. Isključivo ukoliko bude problema sa pokretanjem Eclipse, neophodno je postupiti po uputstvu datom ispod.

Definisanje JAVA_HOME i Path promenljive u Windows 10 operativnom sistemu (uraditi ukoliko se ne pokrene Eclipse):

1. Pokrenuti **Control Panel** i kliknuti na ikonicu **System**.
2. Kliknuti na **Advanced System Settings**
3. Kliknuti na karticu **Advanced**.
4. Kliknuti na dugme **Environment Variables**.
5. Pod stvakom **System Variables**, kliknuti nadugme **New**.
6. Za naziv nove promenljive (Variable name) uneti **JAVA_HOME**.
7. Uneti vrednost promenljive kao putanje do foldera u kom je instaliran Java Development Kit.
Na primer: C:\Program Files\Java\jdk-17.0.2
8. Kliknuti na dugme potvrde **OK**.
9. Zatim treba izmeniti vrednost sistemske promenljive **Path**, tako što ćete u prozoru sa spiskom sistemskih promenljivih kliknuti na promenljivu **Path**, a zatim na dugme **Edit**. U prozoru za editovanje vrednosti sistemske varijable, u prvi slobodan red ispod spiska postojećih vrednosti, treba dodati vrednost putanje do **bin** foldera JDK-a:
%JAVA_HOME%\bin
10. Kliknuti na dugme potvrde **OK**.
11. Ponovo pokrenuti Windows. (obavezno)

Provera da li je JAVA_HOME postavljen: Otvoriti command prompt i ukucajte **SET JAVA_HOME**. Ukoliko vidite odgovarajuću putanju varijabla je postavljena. Ukoliko vidite naziv "Environment Variable JAVA_HOME not defined", ponovite proceduru za unos odgovarajuće varijable.

Provera da li je PATH postavljen da vidi %JAVA_HOME%\bin folder: Otvoriti command prompt i ukucajte **SET PATH** i proverite da li je PATH dobar.

< -----Linux ----->

Za razliku od Windows platforme gde se na x64 bitnom OS može instalirati 32 bitna Java, za Linux platforme to nije slučaj. Kod Linux platforme obavezno se mora koristiti instalacija jave prilagođena verziji OS (x86 ili x64).

Prvi korak u instalaciji Jave je preuzimanje odgovarajuće tar.gz arhive sa Oraklovog sajta i njeno smeštanje u okviru Home direktorijuma u folderu Downloads (~/.Downloads). Pokrenuti komandnu liniju i pozicionirati se u okviru pomenutog foldera. Ekstrahovati sadržaj tar.gz arhive koja sadrži instalaciju Jave komandom tar:

(32bit)

```
tar -zxvf jdk-8uXY-linux-i586.tar.gz
```

ili

(64bit)

```
tar -zxvf jdk-8uXY-linux-x64.tar.gz
```

Posle ekstrakcije u okviru Downloads foldera biće kreiran folder jdk1.8.0_XY u kome se nalazi instalacija Jave.

Zatim je potrebno kreirati folder jvm/jdk1.8.0 na putanji /usr/lib u kome će se biti smeštena Java. Za kreiranje foldera potrebno je poznavanje administratorskih privilegija (šifre). Kreiranje foldera sa administratorskim privilegijama postiže se komandom:

```
sudo mkdir /usr/lib/jvm/jdk1.8.0
```

Sledeći korak zahteva prebacivanje instalacije Jave u novokreirani folder.

```
sudo mv ./jdk1.8.0_XY /usr/lib/jvm/jdk1.8.0
```

Da bi se koristila java u Linux sistemu potrebno je postaviti veze između Linux sistema i instalacionog foldera Jave. To se postiže sledećim naredbama:

```
sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.8.0/bin/java" 1
```

```
sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.8.0/bin/javac" 1
```

```
sudo update-alternatives --install "/usr/bin/javaws" "javaws" "/usr/lib/jvm/jdk1.8.0/bin/javaws" 1
```

Pozivom komande java -version proverićemo da li je Java instalirana. Često se dešava da Linux instalacija dolazi sa već preinstaliranom Javom (open Jdk). U prethodnom slučaju potrebno je naznačiti sistemu koju instalaciji Jave da koristi kao primarnu. Upravljanje sa više instalacija Jave postiže se komandama

```
sudo update-alternatives --config java
```

Prethodna komanda će vratiti listu instalacija, gde se zadavanjem broja (koji se nalazi ispred željene instalacije) definiše podrazumevana instalacija koja će se koristiti (u našem slučaju instalacija "/usr/lib/jvm/jdk1.8.0/bin/java").

Ponoviti prethodno pomenuti postupak i za javac i javaws

```
sudo update-alternatives --config javac
```

```
sudo update-alternatives --config javaws
```

Vodite računa da preuzmete najnoviju verziju Oraklove Jave 8.

<http://askubuntu.com/questions/55848/how-do-i-install-oracle-java-jdk-8>

Kompajliranje i pokretanje Java koda

Kompajliranje java klasa iz konzole postiže se komandom `javac`. Kao parametri poziva pomenutoj komandi se prosleđuje ime java klase, pri čemu će rezultat biti odgovarajući `.class` fajl (mogu se prosleđivati i drugi parametri). Npr. izvršavanjem komande `javac Test.java` nastaje fajl `Test.class`. Pokretanje java programa unutar Java virtuelne mašina postiže se upotrebom komande `java Test.class`. Ukoliko je za kompajliranje ili pokretanje potrebno pristupiti i drugim klasama `javac` i `java` će te klase tražiti upotrebom **CLASSPATH** varijable. Ukoliko prilikom pokušaja pokretanja programa upotrebom komande `java Test.class` doviđete poruku „could not find or load main class“, neophodno je da podesite CLASSPATH sistemsku promenljivu. Procedura je ista kao i za podešavanje JAVA_HOME promenljive, koja je opisana iznad, do tačke 5. U tački 6. umesto naziva promenljive JAVA_HOME, uneti naziv CLASSPATH, u tački 8. uneti vrednost promenljive kao putanje do **lib** loldera instaliranog JDK-a, na primer: `%JAVA_HOME%\jre\lib;`, (uključujući i . iza ;).

Za kompajliranje java fajlova i njihovo pokretanje na ovom kursu, korišćemo alat Eclipse, koji ove radnje automatizuje i olakšava.

2. Eclipse

Eclipse predstavlja open-source razvojnu platformu za više programskih jezika, odnosno, integrisano okruženje za razvoj softvera (*IDE – integrated development environment*). U ovom momentu je jedno od najčešće korištenih razvojnih okruženja za kreiranje Java aplikacija (pored IntelliJ i NetBeans).

Instalacija

Instalacija Eclipse alata je moguća na dva načina, preuzimanjem zapakovanog programa (zip) i raspakivanjem na željenoj lokaciji ili instaliranjem programa korištenjem instalera. Oba postupka su opisana ispod:

1. Preuzimanje zapakovanog programa (verzija *Eclipse IDE for Enterprise Java Developers*):

1.1. Ukoliko imate 64-bitni Windows, preuzmite zapakovan program Eclipse sa linka ispod
https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2022-03/R/eclipse-jee-2022-03-R-win32-x86_64.zip

1.2. Ako imate 32-bitni Windows, preuzmite zapakovan program Eclipse sa sledećeg linka:
<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2018-09/R/eclipse-jee-2018-09-win32.zip>

Odnosno, preuzmite instalaciju koja **odgovara vašem operativnom sistemu** (Windows, Linux, Mac OS...) i **njegovoj bitnoj verziji** (x86 ili x64).

U nastavku slede primeri određenih instalacija:

Windows x86 naziv fajla `eclipse-jee-mars-1-win32.zip`

Windows x64 naziv fajla `eclipse-jee-mars-1-win32-x86_64.zip`

Linux x86 naziv fajla `eclipse-jee-mars-1-linux-gtk.tar.gz`

Linux x64 naziv fajla `eclipse-jee-mars-1-linux-gtk-x86_64.tar.gz`

Mac OS X naziv fajla `eclipse-jee-mars-1-macosx-cocoa-x86_64.tar.gz`

< -----Windows ----->

U folderu Downloads otpakovati sadržaj preuzete zip arhive (npr. `eclipse-jee-2020-12-R-win32-x86_64.zip`), a zatim raspakovani folder pod nazivom `eclipse` preimenovati u `Eclipse` i premestiti (cut - paste) u folder C:\Program Files.

Na disku D: u folderu D:\ kreirati direktorijum Workspace (ako nemate disk D: folder workspace kreirajte na disku C:).

Pokrenuti aplikaciju Eclipse preko C:\Eclipse\eclipse.exe i postaviti putanju radnog direktorijuma Eclipse na D:\Workspace

< -----Linux ----->

Otpakovati sadržaj preuzete zip arhive (npr. `eclipse-standard-kepler-SR1-win32.zip`) na u okviru Vašeg home direktorijuma ~/ i promeniti naziv otpakovanog fajla sa `eclipse` na `eclipseLinux`.

Na disku okviru home direktorijuma ~/ kreirati direktorijum JTemp.

Pokrenuti aplikaciju eclipse preko komande `sh ~/eclipseLinux/eclipse` i postaviti putanju radnog direktorijuma Eclipse na ~/JTemp

2. Instaliranje programa korištenjem instalera (verzija *Eclipse IDE for Java EE Developers*):

Ukoliko imate 64-bitni Windows, preuzmite Eclipse Installer sa sledećeg linka:

<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2022-03/R/eclipse-inst-jre-win64.exe>

Odnosno, preuzmite instalaciju koja **odgovara vašem operativnom sistemu** (Windows, Linux, Mac OS...) i **njegovoj bitnoj verziji** (x86 ili x64).

U slučaju da pri pokretanju preuzetog fajla web browser prikaže Security Warning dijalog, odaberi opciju Run. Odaberi opciju *Eclipse IDE for Java EE Developers*. Nakon toga je moguće odabrati instalacionu putanju (može se ostaviti ponuđena) i pritiskom na dugme *Install* se započinje instalacija. Pri uspešnom završetku instalacije žuto dugme *Install* će postati zeleno dugme *Launch* za pokretanje Eclipse alata. Dodatni detalji o instalaciji se nalaze na gore naznačenoj adresi.

< ----- Podašavanje i provera Jave za program Eclipse----->

Predefinisano ponašanje Eclipse je da preuzima poslednju instaliranu verziju Jave, kao aktuelnu za pokretanje i kompajliranje Java programa.

Podešavanje instalacija JRE može se postići u Eclipse programu odlaskom na Window->Preferences->Java->Installed JREs

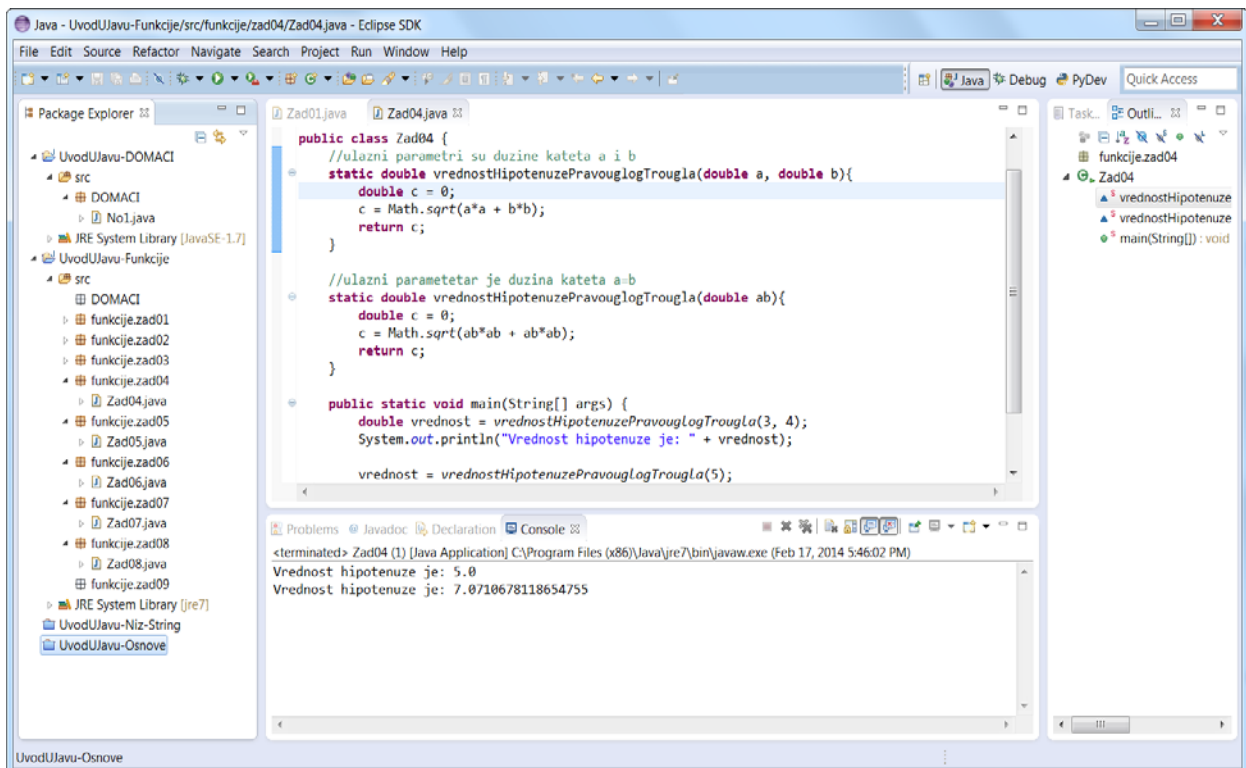
Podešavanje JDK i kompatibilnosti kompajlera u Eclipse programu odlaskom na Window->Preferences->Java->Compiler

Kompatibilnost pokretanja i kompajliranja Java programa je bitna, ukoliko je kod inicijalno pisan za jednu (npr. java 5) a pokušava se kompajlirati za drugu (npr. 8) verziju jave (različite verzije Jave).

Pri kreiranju novog Java projekta, podešavanje Jave preuzeće se iz podešavanja Eclipse. Podešavanja Jave moguće je naknadno izvršiti, u okviru svakog Java projekta zasebno.

Okruženje:

Pri pokretanju okruženja potrebno je definisati koji folder na hard disku će biti korišten kao **radni prostor** (**workspace**). Za isti Eclipse može biti definisano više workspace-ova, na koje korisnik može primeniti različita podešavanja. Radni prostor je moguće organizovati na više načina, odnosno, **perspektiva** (**perspective**). Različite perspektive uključuju određeni skup alata i specijalizovane su za odgovarajuće namene (desni gornji ugao Eclipse alata; perspektive: Java - razvoj standardnih java aplikacija, Jave EE - prošireno standardno okruženje koje se npr. može koristiti za razvoj web aplikacija, Debug - okruženje za debugovanje/trazjenje grešaka).



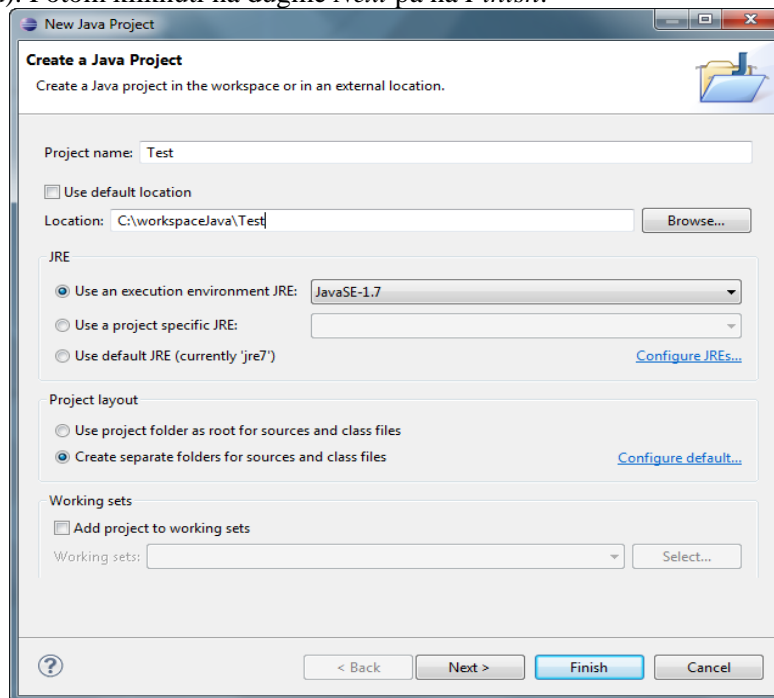
Java perspektiva se sastoji od sledećih **pogleda**: **PackageExplorer** (prikazuje listu projekata koji se nalaze u radnom prostoru i hijerarhiju Java resursa u okviru projekata), **Outline** (prikazuje listu atributa i metoda klase koja se edituje), **Problems** (prikazuje tabelu grešaka i upozorenja vezanih za izvorni kod), **Console** (prikaz poruka na ekran za pokrenuti java program) i **Editor** (zausima centralni deo površine radnog prostora). Pogledi koji se nalaze

u određenoj perspektivi se mogu prilagoditi potrebama i navikama programera. Mogu se dodavati novi ili brisati postojeći pogledi iz perspektiva, a mogu se i menjati veličine i položaj pogleda unutar radnog prostora. Novi pogledi se dodaju iz opcije glavnog menija *Window->Show view*.

Rad sa projektima

Programski kod se organizuje unutar projekata. Projekat se sastoji od hijerarhije klasa (u skladu sa pravilima definisanja paketa u Java programskom jeziku) i drugih datoteka (npr. slika, txt fajlova, konfiguracionih datoteke itd.). U okviru radnog prostora može da bude definisano više projekata. Projekat može da bude otvoren (ikonica otvorena fascikla) ili zatvoren (ikonica plava zatvorena fascikla). Samo otvoreni projekti su aktivni, tj. nad njima je jedino dozvoljen rad u eklipsi. Zatvoreni projekti ne mogu da se menjaju jer nisu aktivni (ali se i dalje nalaze u lokalnom fajl sistemu). Poželjno je **zatvarati projekte** na kojima trenutno ne radimo, jer se na taj način oslobađaju zauzeti memorijski resursi i podižu performanse Eclipse okruženja.

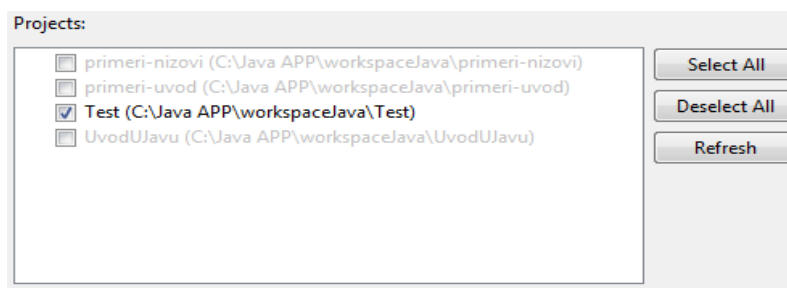
Kreirati svoj prvi java projekat pokretanjem vizarda *File->New->Java Project*. Dodeliti vrednost Test kao ime projekta (pogledati sliku). Potom kliknuti na dugme *Next* pa na *Finish*.



Zatvaranje otvorenog projekta - Desni klik na projekat pa stavka **Close Project**. Projekat je zatvoren ali ne i uklonjen iz okruženja.

Uklanjanje projekta iz okruženja - Projekte je moguće ukloniti logički i fizički iz radnog prostora. Uklanjanje se postiže desnim klikom na selektovani projekat, klikom na stavku **Delete** i na dugme OK (za fizičko brisanje neophodno je čekirati polje ispred „delete project contents od disk“).

Uvoz (import) postojećeg projekta - Prebaciti folder postojećeg projekta u radni direktorijum (workspace npr. *Jtemp*). Pozvati opciju *File->Import->General-> Existing project into Workspace* i kliknuti na dugme **Next**. Kliknuti na dugme **Browse**, postaviti putanju do radnog direktorijuma *JTemp*. Iz liste ponuđenih projekata odabrati onaj koji se importuje i zatim kliknuti na dugme **Finish**.

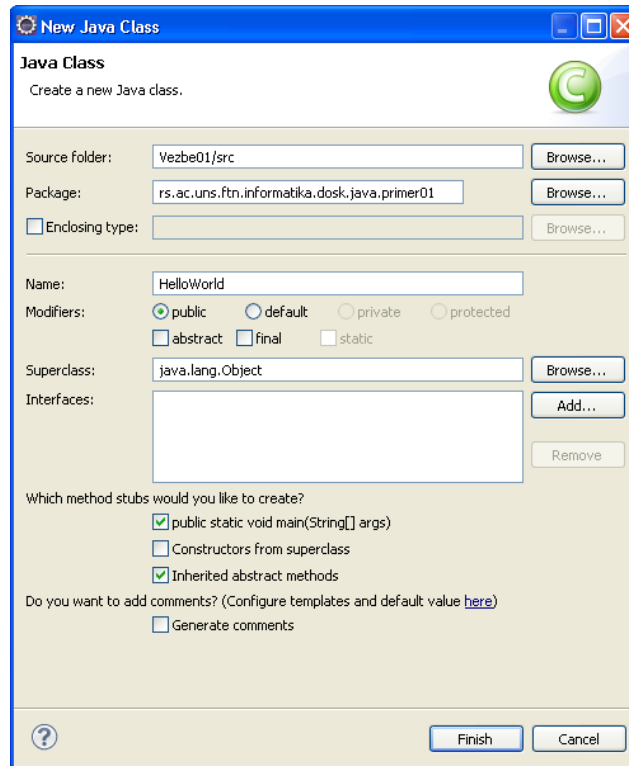


Rad sa paketima

Paketi predstavljaju hijerarhijsku organizaciju fajlova po folderima i podfolderima. Paketi se kreiraju pokretanjem vizarda, desnim klikom na *src folder* → *New* → *Package* i zadavanjem imena novog paketa. Kreirati nekoliko novih paketa pod nazivom *vezbe01*, *vezbe02.podpaket*, *vezbe03.podpaket.podpaketpodpaketa*. Kroz operativni sistem otići na lokaciju projekta na disku i istražiti sadržaj foldera *src*. U pakete se dodaju Java klase. Moguće je promeniti naziv paketa, kopirati klase iz jednog paketa u drugi ili obrisati klase. Imena paketa treba da budu jedinstvena, ne na nivou projekta nego na globalnom nivou. Pored paketa u projekte se mogu dodavati i folderi, tekstualni fajlovi, klase, itd.

Rad sa Java klasama

Kreirati svoju prvu Java klasu pokretanjem vizarda, desni klik na *paket* → *New* → *Class*. Dodeliti naziv *HelloWorld* kao ime klase (pogledati sliku) i selektovati opciju *public static void main*. Potom kliknuti na dugme **Finish**.



Nakon toga u centralnom editoru otvara se *HelloWorld* klasa u kojoj se nalazi i *public static void main* metoda. Metoda *main* je mesto odakle počinje izvršenje programa.

Eclipse nudi i **autocomplete**, odnosno omogućava Vam da započnete naredbu unošenjem prvih nekoliko karaktera i pritisnete **Ctrl+Space**, nakon čega će ta naredba biti dovršena. Na primer: unutar *main* metode napišite *syso*, pritisnite **Ctrl+Space** i Eclipse će napisati *System.out.println();* Unutar zagrada napišite "Hello world!".

Kompajliranje i pokretanje

Kompajliranje se obavlja automatski, ali se ovakvo ponašanje Eclipse-a može i promeniti putem opcije iz glavnog menija *Project->Build automatically*. Ako se ova opcija isključi postaju aktivne opcije *Build All* i *Build Project*. Podrazumevano ponašanje Eclipse-a je da kreira *bin* folder u projektu, u koji smešta class fajlove, dobijene kompajliranjem. Ovaj *bin* folder se ne vidi iz *Package Explorer* pogleda, ali se vidi iz *file browser*-a, a takođe se može videti i iz Eclipse okruženja putem *Navigator* pogleda (*Window->Show view*).

Pokretanje programa se bira iz padajuće liste koja se dobija desnim klikom na klasu u *Package Explorer* pogleda. U padajućoj listi postoji opcija *Run As -> Java Application* čijom selekcijom se vrši pokretanje Java virtuelne mašine i u njoj izvršenje *main* metode selektovane klase. U projektu može postojati proizvoljno mnogo klasa koje imaju *main* metodu, ali se pokreće *main* metoda selektovane klase. Prečica za ovu akciju je **Alt+Shift+X**, nakon

čega je potrebno pritisnuti J. Nakon ovoga se izvršava *main* metoda selektovane klase. Eclipse ima mnogo prečica, a njihova lista se može videti (a i menjati) putem opcije *Window -> Preferences -> General -> Keys*.

Metoda *main* može imati i listu ulaznih parametara koji se mogu proslediti prilikom pokretanja programa putem opcije iz padajuće liste koja se dobija desnim klikom na klasu u *Package Explorer* pogleda: *Run As -> Run Configurations...*

Debug

Eclipse okruženje poseduje posebnu perspektivu za debug-ovanje aplikacije (otkrivanje grešaka u aplikaciji) koja pruža mnoštvo mogućnosti. Postoji mnogo dostupnih tekstova o debug-ovanju Java koda pomoću Eclipse-a kao što je na primer: <http://www.vogella.com/tutorials/EclipseDebugging/article.html>.

Plug-ins

Eclipse okruženje je otvorenog koda ali i otvorene arhitekture i vrlo ga je lako proširiti. Postoji mnoštvo proširenja za Eclipse. Instaliranje novog dodatka se može postići opcijom glavnog menija *Help->Install New Software...* (to nije jedini način za instaliranje, ali je najčešći). Potreban Vam je *update site* nekog dodatka (Na primer: http://subclipse.tigris.org/update_1.10.x). Postoje i sajtovi koji sadrže listu dodataka za Eclipse kao što je: <http://marketplace.eclipse.org/>.

2. Java – programski jezik

Sintaksa je slična sintaksi C++.

Sve ključne reči programskog jezika Java se pišu malim slovima. Svaki Java program se sastoji iz deklaracije paketa, import sekcije, deklaracije atributa i definicije metoda. Sva četiri dela su opciona.

Izrazi u programskom jeziku Java se obično završavaju simbolom tačka-zarez ;.

Programski blok čine dve ili više naredbi ili izraza i on je ograđen vitičastim zagradama {}. Ako imamo samo jednu naredbu ili izraz, ne moramo da je pišemo unutar vitičastih zagrada.

Izvršenje programa počinje od *main* metode koja može primati i parametre u vidu liste stringova.

Komentari

Postoje tri vrste komentara: jednolinijski, višelinijski i dokumentacioni.

1. Jednolinijski komentari staju u jedan red i navode se nakon dvostruke kose crte:

// ovo je komentar

Prečica za jednolinijski komentar je Ctrl+/

2. Višelinijski komentari mogu da se pružaju u više redova i nalaze se između /* i */ simbola.

/ ovo*

*je komentar */*

Prečica za višelinijski komentar je Ctrl+Shift+/

3. Postoje i dokumentacioni (JavaDoc) komentari, koji služe za generisanje API dokumentacije, a koji se navode između /** i */ simbola.

*/** ovo je dokumentacioni*

*komentar */*

Tipovi podataka

1. Primitivni tipovi

- kao lokalne promenljive i parametri metoda, čuvaju se na stek-u (deo radne memorije računara)

2. Klase - odnosno objekti kao instance klasa

- instance klasa se zovu objekti i čuvaju se na heap-u (deo radne memorije računara)

Tipovi promenljivih u Javi:

1. Celobrojni tipovi:
 - a. **int**: većina celih brojeva se predstavlja pomoću ovog tipa
 - b. **long**: veliki brojevi, u oblasti istraživanja, finansija, računara
 - c. **short**: mali brojevi, ne mnogo korisno
 - d. **byte**: veoma mali brojevi
2. Decimalni (Floating Point) tipovi:
 - a. **float**: Single-precision decimalni brojevi
 - b. **double**: Double-precision decimalni brojevi
3. Ostali tipovi:
 - a. **String**: tekstualni tip, niz karaktera
 - b. **boolean**: tačno ili netačno (true ili false)
 - c. **char**: Alfanički karakteri

Primitivni tipovi

Lista primitivnih tipova:

<u>Primitivni tip</u>	<u>Veličina</u>	<u>Minimum</u>	<u>Maksimum</u>
<u>boolean</u>	<u>1-bit</u>	-	-
<u>char</u>	<u>16-bit</u>	Unicode 0	Unicode $2^{16} - 1$
<u>byte</u>	<u>8-bit</u>	-128	+127
<u>short</u>	<u>16-bit</u>	-2^{15}	$+2^{15} - 1$
<u>int</u>	<u>32-bit</u>	-2^{31}	$+2^{31} - 1$
<u>long</u>	<u>64-bit</u>	-2^{63}	$+2^{63} - 1$
<u>float</u>	<u>32-bit</u>	<u>IEEE754</u>	<u>IEEE754</u>
<u>double</u>	<u>64-bit</u>	<u>IEEE754</u>	<u>IEEE754</u>
<u>void</u>	-	-	-

Literali za primitivne tipove:

- Celobrojni: **2, 2000000L**
- Razlomljeni: **3.14f, 3.14**
- Heksadecimalni: **0xF, 0xFF**
- Oktalni: **0123**
- Binarni: **0b1001101**
- Boolean: **true, false**
- Znakovni:
 - **'a'**
 - **'\n'**
 - **'\xxx'**, gde je **xxx** oktalni ASCII kod karaktera

Deklaracija i inicijalizacija primitivnih tipova:

- `int a`; ovo je samo deklaracija celobrojne promenljive `a`
- `int a = 0`; ovo je istovremena deklaracija i inicijalizacija celobrojne promenljive `a`
- `int a, b`; ovo je istovremena deklaracija dve celobrojne promenljive `a` i `b` - može samo kad su istog tipa
- `int a = 0, b = 3`; ovo je istovremena deklaracija i inicijalizacija dve celobrojne promenljive `a` i `b` - i ovo može samo za promenljive istog tipa

Operatori

Postoje sledeće vrste operatora:

1. Aritmetički: +, -, *, /, % (umesto $x = x + 1$ može $x += 1$)
2. Relacioni: <, >, <=, >=, ==, !=
3. Logički: && (I), || (ILI), ! (NE)
4. Bit operatori
5. Operator dodele vrednosti: =

Konvencije u kodiranju:

1. Konvencija identacije - je konvencija o nazublivanju, odnosno uvlačenju blokova koda, radi bolje preglednosti i predstavljanja strukture koda.
2. Konvencije imenovanja - predstavljaju skup pravila kojih se treba pridržavati, prilikom definisanja identifikatora, odnosno, naziva paketa, klasa, lokalnih promenljivih...

Koristeći standardne konvencije o imenovanjima u Javi, kod postaje lakši za čitanje, kako za onoga ko ga je pisao tako i za druge programere koji treba da održavaju ili debuguju predhodno kreiran kod. Čitljivost Java programa je veoma važna. Poštovanje konvencija **skraćuje vreme** potrebno da se shvati šta kod radi.

Ime	Konvencija
Ime klase	Treba početi velikim slovom i biti imenica npr. String, Color, Button, System, Thread itd.
Ime interfejsa	Treba početi velikim slovom i biti pridev npr. Runnable, Remote, ActionListener itd.
Ime metode	Treba početi malim slovom i biti glagol npr. actionPerformed(), main(), print(), println() itd.
Ime promenljive	Treba početi malim slovom npr. firstName, orderNumber itd.
Ime paketa	Treba biti ispisano malim slovima npr. java, lang, sql, util itd.
Ime konstante	Treba biti ispisano velikim slovima npr. RED, YELLOW, MAX_PRIORITY itd.

Kamilja notacija (camelcase) u konvencijama imenovanja u Javi:

Java sledi tzv. kamilju notaciju za imenovanje klase, interfejsa, metode i varijable. Ako je ime kombinovano od dve reči, druga reč će uvek početi sa velikim slovom napr. actionPerformed(), firstName, ActionEvent, ActionListener itd.

3. Zadaci za vežbanje

1. Ispisati na konzoli sledeći tekst (primena konstrukcije za uvlačenje teksta)

A.

B.

B.1.

B.1.1.

B.2.

C.

2. Napisati program koji za date stranice $a=3$, $b=4$ i $c=5$ računa površinu i zapreminu kvadra. Ispisati na ekran izračunate vrednosti. $P = 2(ab+ac+bc)$; $V = a * b * c$;

3. Napisati program koji izračunava površinu kupe prečnika 6 i visine 4? Ako nam je poznata formula:

$$S = S_o + S_b = r s \pi + r^2 \pi = r \pi (s + r), \text{ gde je } s \text{ dužina izvodnice.}$$

Ispisati na ekran izračunatu vrednost.