

Uputa za izradu projekta

U skladu s najavama na početku semestra, kao jedan od preduvjeta za prolaz vještine potrebno je napraviti zadovoljavajuću verziju projekta.

1. Uvod

Projekt koji ćete rješavati nije programski previše zahtjevan, iako ima svojih momenata. Osnovni izvor složenosti projekta je potreba za savladavanjem i integracijom nekoliko tehnologija s kojima ste se do sada susreli (ili još niste). Konkretno, da biste riješili ovaj projektni zadatak, trebat ćete:

- koristiti Apache Maven kao sustav za izgradnju projekta koji sve JAR-arhive o kojima ovisi Vaš projekt dohvaća automatski iz repozitorija koji su dostupni na Internetu;
- koristiti alat GIT za upravljanje verzijama izvornih kodova – izvorne kodove Vašeg projekta pohranit ćete u repozitorij koji sam Vam otvorio na poslužitelju i potom ćete najmanje **svaki dan** nakon što završite s pisanjem koda odnosno drugih resursa u repozitorij pohraniti trenutnu verziju projekta (možete i češće, naravno; u dane kada ništa ne radite ništa ne trebate niti ažurirati u repozitoriju);
- stvoriti javni i privatni ključ kako biste kriptografski potpisali JAR-arhivu koja sadrži Vaš programski kod;
- podesiti Javu na Vašem računalu tako da vjeruje programskom kodu koji je potpisan Vašim tajnim ključem;
- podesiti Javu na Vašem računalu tako da vjeruje programskom kodu koji je potpisan tajnim ključem koji je korišten za potpisivanje osnovnih biblioteka koje sam Vam pripremio a koje će koristiti Vaš kod;
- u razvojnom okruženju Eclipse naučiti kako raditi s Maven projektima i sustavom za vođenje revizija koda;
- implementirati sve što se traži u okviru projekta:
 - sav izvorni kod mora biti prikladno dokumentiran kako bi se za njega mogao dobiti javadoc,
 - dio koji nije vezan uz grafičko korisničko sučelje mora biti pokriven *junit* testovima,
 - potrebno je napisati korisničku dokumentaciju,
- predati i prezentirati projekt te dobiti potvrdu da se projekt prihvaća (odnosno da je završen).

Poveznice koje biste trebali pogledati su:

<http://git-scm.com/book/en/>

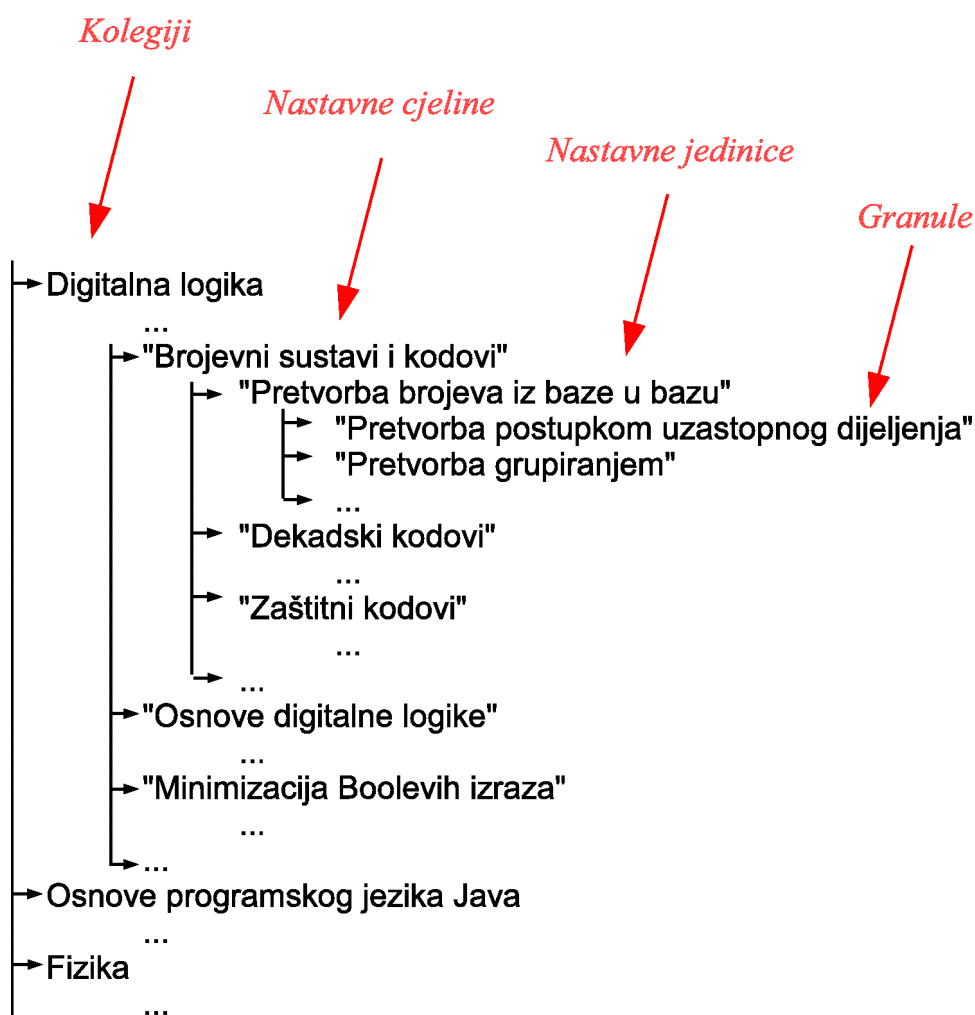
<http://maven.apache.org/users/index.html>

(brzinski prolaz: <http://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>)

Radite li na Windowsima, predlažem da instalirate <https://code.google.com/p/gitextensions/>.

2. Problem koji rješavate

Kao pripremu za ovaj projekt razradio sam jednostavan sustav e-učenja koji podržava prikaz napisanih interaktivnih materijala svojim korisnicima. Konceptualno, u sustavu postoje kolegiji, svaki kolegij ima nastavne cjeline (primjerice: "Brojevi sustavi i kodovi", "osnove digitalne logike", "Minimizacija Boolevih izraza", ...) a svaka cjelina ima više nastavnih jedinica (u projektu ćemo koristiti engleski termin *Knowledge Unit*, KU). Svaka nastavna jedinica može imati više granula (u projektu ćemo koristiti engleski termin *Knowledge Item*, KI). Tako primjerice, nastavna cjelina "Brojevi sustavi i kodovi" mogla bi imati nastavne jedinice "Pretvorba brojeva iz baze u bazu", "Dekadski kodovi", "Zaštitni kodovi", ... dok bi nastavna jedinica "Pretvorba brojeva iz baze u bazu" mogla imati granule "Pretvorba postupkom uzastopnog dijeljenja", "Pretvorba grupiranjem", ...



Sustav e-učenja napisan je modularno tako da autorima omogućava pisanje granula kao samostalnih paketa koji se potom samo učitaju u sustav i dalje automatski rade. Predviđeno je da se svaka granula distribuira u obliku distribucijskog paketa koji čini jedna ZIP-arhiva. U toj arhivi nalazi se propisana struktura direktorija koja autorima omogućava pohranu napisanih stranica koje sadrže pisani dio materijala za učenje (direktorij *parts*), prikaz statičkih resursa koje te stranice trebaju (direktorij *resources*), JAR-arhiva koja je digitalno potpisana i koja sadrži implementacije svih interaktivnih demonstracija koje se koriste u matrijalu za učenje (*demonstration.jar*), računalu čitljiv opis

granule znanja koji definira od kojih se stranica i demonstracija sastoji granula te koji su preduvjeti za pristupanje pojedinoj stranici (`knowledge-item.xml`) te korisničke dokumentacije (direktorij `docs`, početna datoteka `docs/index.html`) koja osobi koja želi modificirati napisani materijal mora dati sva potrebna pojašnjenja o načinu i uporabi napisanih demonstracija (koje demonstracije postoje, kako se zovu pripadni razredi, koje parametre demonstracija podržava i slično). Sve tekstovne datoteke u ovoj arhivi (uključivo html dokumente) moraju biti pohranjene uporabom kodne stranice UTF-8.

Konceptualni model granule je jednostavan: granula se sastoji od slijeda od nekoliko stranica. Na većini stranica, uz napisani tekst nalazit će se interaktivna demonstracija koju će korisnik morati isprobati (ako demonstracija sadrži animaciju, korisnik će je morati odgledati; ako traži od korisnika da klikanjem prođe kroz niz koraka, korisnik će to stvarno morati naklikati, itd.). Jednom kada je demonstracija odgledana, demonstracija će to dojaviti sustavu e-učenja signaliziranjem određene zastavice. Prilikom pisanja opisa granule, autor granule za svaku će demonstraciju definirati naziv zastavice koju demonstracija signalizira, a za svaku stranicu popis zastavica koje moraju biti signalizirane kako bi korisnik mogao pristupiti toj stranici. Jednom kada je opis granule korektan, sustav će novom korisniku od svih napisanih stranica ponuditi samo onu koja nema niti jedan preduvjet. Na toj stranici vjerojatno će postojati neka demonstracija koja će, kada je korisnik odgleda, signalizirati prvu zastavicu. Sustav će potom korisniku ponuditi da pređe na sljedeću stranicu (za koju je sada ispunjen preduvjet pristupanja); ta će stranica vjerojatno imati neku drugu demonstraciju koja će opet signalizirati neku novu zastavicu, i tako sve do pregledavanja posljednje stranice.

U okviru ovog projekta zadana Vam je tema za koju morate napraviti maven-projekt koji će sadržavati programski kod i sve potrebne resurse na temelju kojih će se automatski generirati ZIP arhiva koja predstavlja distribucijski paket za zadanu granulu znanja. Same demonstracije pišu se kao klasični Java razredi koji se izvode iz sučelja `hr.fer.zemris.sicoma.demo.JinteractiveDemo`. U okviru tog sučelja postoji metoda čijim će se pozivanjem očekivati dostava Swing-komponente koja čini korisničko sučelje demonstracije. Uporabom tehnologije Java Applets ove se komponente automatski učitavaju u web-stranice nastavnih materijala.

3. Prvi korak: dohvat demonstracijskog paketa

S adrese <https://ferko.fer.hr/projects/SiCoMa/SiCoMa-TestWebGUI-Prepacked.zip> skinite Web-temeljeni sustav za probni prikaz napisanih granula znanja. ZIP arhivu koju ste dobili negdje raspakirajte (ovo nije dio Eclipseovog projekta – služi samo za početno upoznavanje sa sustavom). Detaljan opis daljnji koraka dan je u potpoglavljima koja slijede. Ovdje je dan sažetak.

U direktoriju `demonstrations` pronaći ćete već napisanu granulu *Uvod u Bezierove krivulje*. Javni ključ koji se nalazi u datoteci `sicoma-certificate.cer` učitajte u Javin popis ključeva kojima se vjeruje. Ako ovaj dio ne napravite, Java plugin u pregledniku neće dopustiti izvođenje naših Java apleta. Ako ovo napravite, plugin će Vas pitati želite li dopustiti pokretanje i ponudit će mogućnost da odluku zapamtite.

Pokrenite poslužitelj. Poslužitelj će automatski slušati na portu 8088. Pokrenite bilo koji preglednik i spojite se na <http://localhost:8088/>. Prođite kroz napisanu granulu: sve demonstracije moraju raditi dobro. Demonstracija će biti označena kao pogledana tek kada mišem uhvatite jednu od točaka Bezierove krivulje i kada je pomaknete da vidite kako to mijenja krivulju. Kod demonstracija koje nude više opcija, morate isprobati sve opcije. Kod demonstracija koje sadrže animaciju morate odgledati jedan ciklus animacije – demonstracija će se tada sama označiti kao odgledana. Proučite malo ZIP arhivu (strukturu, datoteke).

Napomena: iako je moguće promijeniti naziv poslužitelja i port na kojem poslužitelj sluša, nemojte to raditi. Zbog pojačanog sigurnosnog modela koji se od nedavno primjenjuje u tehnologiji Java Applets, da bi sve radilo korektno, nužno je osigurati da je kôd koji se pokreće digitalno potpisan i da je specificirano za koji je poslužitelj (ime+port) namijenjen. Stoga su sve biblioteke (za koje je to važno) u okviru ovog projekta potpisane uz informaciju da će se koristiti na poslužitelju `localhost:8088`. Ako to želite promijeniti, nužno je ponovno prekompajlirati i nanovo digitalno potpisati dio biblioteka SiCoMa projekta (uputa za to dana je nešto kasnije u ovom dokumentu).

3.1. Učitavanje javnog ključa u popis ključeva kojima se vjeruje

Više o radu s certifikatima može se pročitati na adresi:

https://blogs.oracle.com/java-platform-group/entry/self_signed_certificates_for_a

Za potrebe pokretanja demonstracijskog paketa programski kod je potpisan certifikatom projekta SiCoMa koji je dan u datoteci `sicoma-certificate.cer`. Osigurajte da ste u konzoli pozicionirani u direktoriju s ovom datotekom. Potom zadajte naredbu:

```
keytool -importcert
        -keystore "C:\Program Files (x86)\Java\jre7\lib\security\cacerts"
        -storepass changeit -alias sicoma -file sicoma-certificate.cer
```

(prethodnu naredbu napišite sve u jednom retku – ovdje je prijelom zbog širine stranice)

Naravno, prilagodite lokaciju gdje imate instaliranu javnu verziju JRE-a koju koristi Java-plugin. Sažetak ovog ključa je:

```
Certificate fingerprint (SHA1):  
AA:50:DB:44:B7:3C:85:A6:2D:01:B9:7B:73:E8:F3:10:13:B5:F5:42
```

pa provjerite da je to doista tako kada radite import (sustav će Vam ovo ispisati). Ako je sažetak neki drugi, prekinite učitavanje (CTRL+C) – netko Vam je podvalio krivi ključ i tom ključu nikako ne želite vjerovati. Defaultna šifra koja se koristi za zaštitu Javinog repozitorija ključeva je doista `changeit` (doslovno), tako da nemojte u gornjoj naredbi to pokušavati mijenjati (ako niste sami promijenili ovu šifru – no tada i sami znate što radite pa Vam ova uputa ne treba). Dane staze vrijede za operacijski sustav Windows.

Da biste provjerili je li sve prošlo ispravno, pokušajte provjeriti integritet jednog od potpisanih JAR-ova iz direktorija `dep` (pretpostavka je da ste i dalje u vršnom direktorij). Zadajte naredbu:

```
jarsigner -verify dep\SiCoMa-ClientGUI-1.0-SNAPSHOT.jar
```

i kao odgovor biste trebali dobiti:

```
jar verified.
```

```
Warning:  
This jar contains entries whose certificate chain is not validated.  
This jar contains signatures that does not include a timestamp. Without a  
timestamp, users may not be able to validate this jar after the signer  
certificate's expiration date (2024-05-12) or after any future revocation date.
```

```
Re-run with the -verbose and -certs options for more details.
```

Prvi redak sam označio crvenom bojom – mora pisati "jar verified." što znači da je integritet koda u arhivi očuvan. Kako koristimo samo-potpisani certifikat, alat će izbaciti i prikazana upozorenja, ali njih u ovom slučaju možemo zanemariti.

3.2. Pokretanje poslužitelja

Za potrebe pokretanja poslužitelja pripremljene su dvije skripte. Skriptu `pokreni.bat` koristite ako ste na operacijskom sustavu Windows a skriptu `pokreni.sh` ako ste na operacijskom sustavu Linux. U obje skripte nalaze se opcije uz koje se poslužitelj pokreće. U tim opcijama pronaći ćete način kako promijeniti port na kojem testni poslužitelj sluša, kao i druge parametre. Savjet: dok se dobro ne upoznate s ovim projektom (i izgradnjom čitave SiCoMa infrastrukture – SiCoMa je krovni naziv projekta u okviru kojeg je napravljen ovaj sustav za prikaz granula), nemojte ovo mijenjati. Odlučite li se na promjenu porta ili naziva poslužitelja (dakle, ako nije `localhost:8088`) morat ćete nanovo izgraditi čitav SiCoMa projekt kao i granule koje želite gledati jer ih morate potpisati uz informaciju o poslužitelju i portu na kojem će biti korištene.

Prilikom pokretanja poslužitelja, poslužitelj na određenim mjestima traži različite datoteke (granule, implementaciju omotača granule kao Java Apleta i slično). Ako mijenjate bilo što, pripazite da sustav može doći do tih potrebnih datoteka. Po pokretanju, poslužitelj će u konzolu ispisati staze do datoteka koje koristi.

Prekidanje poslužitelja radi se na uobičajen način: CTRL+C.

4. Drugi korak: priprema Eclipseovog workspacea

Za potrebe ovog projekta napravite u Eclipse-u čitav novi workspace – nazovite ga *SiCoMa*. U njemu ćete dodati nekoliko projekata.

4.1. Generiranje Vašeg para javni+privatni ključ

U direktoriju prethodno-stvorenog Eclipseovog workspacea ručno napravite direktorij: `SiCoMa-MainConfig`, pa u njemu poddirektorij `codesignstore`. Pozicionirajte se u ovaj posljednji direktorij u konzoli. Tu ćete napraviti svoj vlastiti javni i privatni ključ kojim ćete potpisivati kod koji Vi napravite. Ovaj će ključ biti Vaše vlasništvo. Generirani par ključeva negdje ćemo trebati pohraniti; Java za to koristi repozitorije ključeva (datoteke koje uobičajeno imaju ekstenziju `.keystore`, no nije nužno, kao što smo vidjeli kod datoteke `cacerts`). Jedan repozitorij može sadržavati jedan ili više parova ključeva. Cjelokupni repozitorij štiti se šifrom repozitorija (navodi se argumentom `-storepass`). Svaki privatni ključ koji pohranimo u repozitorij dodatno se štiti vlastitom šifrom (navodi se argumentom `-keypass`).

Zadajte sada naredbu kojom ćete zatražiti generiranje para javnog+privatnog ključa. Tom paru u repozitoriju je potrebno dodijeliti i alias (kratko simboličko ime pod kojim ćemo ga kasnije referencirati). Želimo sljedeće.

- Repozitorij ključeva imena "kljucevi.keystore" koji se štiti šifrom "supertajna".
- Želimo da nam sustav generira par ključeva (javni+tajni) za uporabu krypto-algoritmom RSA.
- Duljina ključeva bi trebala biti 2048 bita.
- Valjanost ključeva bi trebala biti 10 godina od trenutka stvaranja (365 dana x 10 godina = 3650).
- Stvoreni par ključeva želimo zaštititi šifrom "tajna7".
- Stvorenom paru ključeva želimo dodijeliti alias "marko-java".

Potrebna naredba je:

```
keytool -genkeypair -alias marko-java -keyalg rsa -keysize 2048  
-validity 3650 -keystore kljucevi.keystore -keypass tajna7  
-storepass supertajna
```

Želite li bilo što od parametara promijeniti, zapišite si ih negdje (ne smijete ih zaboraviti) i prikladno ažurirajte naredbu. Pokretanjem naredbe uslijedit će serija pitanja kojima se morate identificirati: to su podatci koji će uz Vaš javni ključ biti upisani u certifikat. Primjer ove konverzacije prikazan je u nastavku. Moji unosi prikazani su plavom bojom.

What is your first and last name?
 [Unknown]: Marko Čupić
 What is the name of your organizational unit?
 [Unknown]: ZEMIS
 What is the name of your organization?
 [Unknown]: Fakultet elektrotehnike i računarstva
 What is the name of your City or Locality?
 [Unknown]: Zagreb
 What is the name of your State or Province?
 [Unknown]: Zagrebačka županija
 What is the two-letter country code for this unit?
 [Unknown]: HR
 Is CN=Marko Čupić, OU=ZEMIS, O=Fakultet elektrotehnike i računarstva, L=Zagreb, ST=Zagrebačka županija, C=HR correct?
 [no]: yes

Po završetku imat ćete stvorenu datoteku kljucevi.keystore i u njoj pohranjen ključ. Sada možete tražiti ispis informacija o ovom paru: listamo informacije o ključu čiji je alias *marko-java* (ako ste mijenjali alias, i ovdje ga promijenite):

```
keytool -list -keystore kljucevi.keystore -alias marko-java
-storepass supertajna
```

Dobit ćete ispis oblika:

```
marko-java, Jun 8, 2014, PrivateKeyEntry,
Certificate fingerprint (SHA1):
D6:DA:F3:65:83:45:8B:FD:1B:C0:99:F4:EE:56:95:12:A0:74:F2:07
```

Kako biste javni ključ kojim ćete potpisivati Vaš kod dodali u Javin repozitorij ključeva kojima se vjeruje, najprije iz Vašeg repozitorija morate exportati javni ključ u obliku certifikata, a potom taj certifikat trebate dodati u Javin repozitorij ključeva kojima se vjeruje (naredbu smo već vidjeli). Stoga najprije zatražite export certifikata koji sadrži javni ključ korisnika čiji je alias *marko-java* (želimo da datoteka u koju će biti pohranjen ovaj certifikat dobije ime *marko-java.cer*):

```
keytool -exportcert -keystore kljucevi.keystore -alias marko-java
-file marko-java.cer -keypass tajna7 -storepass supertajna
```

Uočite, da biste napravili eksport, morate navesti i šifru kojom se štiti sam ključ. Na disku će nastati tražena datoteka *marko-java.cer* koja predstavlja certifikat (dokument u kojem su uz javni ključ dane i informacije o identitetu korisnika čiji je to ključ).

Konačno, sada ovaj certifikat možemo učitati u Javin popis certifikata kojima se vjeruje. Naredba je:

```
keytool -importcert
-keystore "C:\Program Files (x86)\Java\jre7\lib\security\cacerts"
-storepass changeit -alias marko-java -file marko-java.cer
```

Sada datoteku s certifikatom možete i obrisati – uvijek je možete ponovno eksportati. Također, ako Vaš kod želite pokrenuti na nekom drugom računalu (ili dati nekome drugome da ga pokrene kod sebe), morat ćete mu dostaviti Vaš certifikat (datoteku *marko-java.cer*, nikako ne *kljucevi.keystore* koja sadrži i Vaš privatni ključ) kako bi prije pokretanja programa taj korisnik mogao na svojem računalu certifikat

importati u popis certifikata kojima se vjeruje. U suprotnom, na tom računalu neće biti moguće provjeriti integritet Vašeg koda pa okoline koje kod izvode pod posebnim sigurnosnim uvjetima (poput Java plugin-a) takav kod neće uopće htjeti pokrenuti. Upravo iz ovog razloga, u poglavlju 3.1 prošli ste kroz korak u kojem ste certifikat kojim su potpisane biblioteke projekta SiCoMa dodali u repozitorij certifikata kojima se vjeruje.

4.2. Konfiguracijska datoteka s postavkama

Prilikom generiranja produkcijskog paketa granule maven mora znati nekoliko informacija temeljem kojih će sve pripremiti. Primjerice, prilikom pakiranja Vaših napisanih demonstracija u JAR arhivu, maven će istu odmah i digitalno potpisati (tako projekt mora biti iskonfiguriran). Međutim, da bi to bilo moguće, maven mora znati kako doći do Vašeg tajnog ključa koji se koristi za potpisivanje (što uključuje sve potrebne informacije: gdje je Vaš repozitorij ključeva, kojom se šifrom štiti, alias para ključeva koji će se koristiti za potpisivanje, šifru za pristup tajnom ključu i slično). Napravite stoga u Vašem *home* direktoriju (na Windowsima, to je C:\Users\VašLogin; na Linuxu to je ~, ako ste logirani kao Vi sami – obično /home/VašLogin) datoteku `sicoma-cm-devel.properties`, sljedećeg sadržaja:

```
#properties
sicoma.configdir = ../SiCoMa-MainConfig
sicoma.security.alias = marko-java
sicoma.security.storepass = supertajna
sicoma.security.keypass = tajna7
sicoma.security.keystore.dir = codesignstore/kljucevi.keystore
sicoma.web.deploy.codebase = localhost:8088
```

4.3. Dohvat gotove granule

Pokrenite u Eclipse-u čarobnjaka za Import. Dodajte Maven projekt direktno iz git-repozitorija (detaljnija uputa kako slijedi u 4.3.2). Adresa do GIT repozitorija je:

```
https://ferko.fer.hr/scm/git/IRG-BezierovaKrivuljaOsnovno
```

Pogledajte kako je projekt organiziran. U projektu ćete pronaći datoteku `README.txt`. Upoznajte se s njenim sadržajem. Pokušajte napraviti neku promjenu u projektu. Primjerice, pokušajte na nekoj stranici promijeniti tekst, ubaciti sliku i slično. Dodajte neku formulu koristeći LaTeX sintaksu kako je opisano u `README.txt`. Kroz Eclipse ponovno izgradite projekt, ili u komandnom retku dok ste pozicionirani u direktoriju projekta zadajte:

```
mvn clean install
```

čime će Maven nanovo izgraditi distribucijski paket (ZIP arhivu) u poddirektoriju `target`. Uzmite tu arhivu i ubacite je u testni web-poslužitelj (pregazite staru arhivu koja je tamo bila u direktoriju `demonstrations`). Restartajte poslužitelj i provjerite je li sve OK. Ovo je izuzetno važan korak jer će sada izvršni kod distribucijskog paketa biti potpisan Vašim privatnim ključem. Ako niste dobro napravili prethodne korake, onda sada Java plugin neće htjeti učitati programski kod demonstracije.

4.3.1. Problemi s certifikatom

Ako iz komandne linije pokušate klonirati prethodni repozitorij:

```
git clone https://ferko.fer.hr/scm/git/IRG-BezierovaKrivuljaOsnovno
```

git će se pobuniti da ne može provjeriti priča li doista s poslužiteljem ferko.fer.hr ili se je u komunikaciju ubacio netko treći tko želi ukrasti Vaše korisničke podatke. Jedno rješenje (koje NE želimo) jest umjesto protokola **https** koristiti protokol **http** koji je sa sigurnosnog stajališta apsolutno nesiguran. Bolje rješenje je podesiti Vaše računalo tako da vjeruje Ferkovom certifikatu. Ferkov certifikat dostupan je na adresi:

<https://ferko.fer.hr/projects/SiCoMa/ferko-certificate.pem>

Skinite tu datoteku. Otvorite je u bilo kojem uređivaču teksta (to je tekstovna datoteka). Njezin sadržaj bi morao biti (ovdje dajem samo početak i kraj datoteke):

```
-----BEGIN CERTIFICATE-----
MIIGmzCCBIOgAwIBAgIJAIvYuTDO3ZXtMA0GCSqGSIb3DQEBBQUAMIGPMQswCQYD
VQQGEwJlUjEucmBoGA1UECBMTWmFncmViYWNRySB6dXBhbmlqYTEPMA0GA1UEBxMG
WmFncmViMQwwCgYDVQQKEwNCRVlxdzANBgNVBAsTBlpFTVJJUzEVMBMGAlUEAxMM
ZmVya28uZmVyLmhyMRswGQYJKoZIhvcNAQkBFgxmZXJrb0BmZXIuaHIwHhcNMTEx
...
m/ijlUtszfzlyvsebayTCgP20MbzxIiLYLTks2VGfsD9nx2PrCn7YaXJ06uV6Q1Q
FB6to9c2E1GzOYC7wnQdMXCdRfn8r+CgEjoeEKGPEnQm7fAJvpsPR11RcIQZiobLB
Jr6U05DBoA+QJoBIYqPcIclMaOibA6UBou6ZVAuSzpVIA2m1DQLMDv0Enl4HHSIH
wu3j74+8wv+269XKrNEY
-----END CERTIFICATE-----
```

Provjerite je li to doista u datoteci koju ste skinuli. Ako nije – ne nastavljajte dalje jer certifikat očito niste skinuli s pravog Ferka. Ako je sve OK, u Vašem korisničkom direktoriju napravite poddirektorij git-certs, i u njega prekopirajte ovu datoteku. Radite li na Linuxu kao korisnik perica, imat ćete:

```
/home/perica/git-certs/ferko-certificate.pem
```

dok ćete na Windowsima imati nešto tipa:

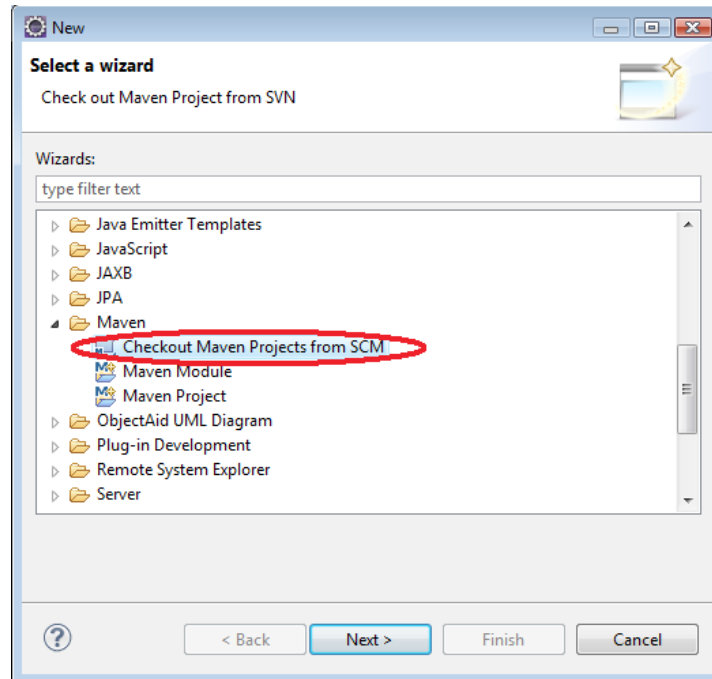
```
C:\Users\perica\git-certs\ferko-certificate.pem
```

Sada je dovoljno u git ljsuci podesiti varijablu koja git-u govori gdje su certifikati kojima vjeruje. Naredba će biti oblika (na Linuxu):

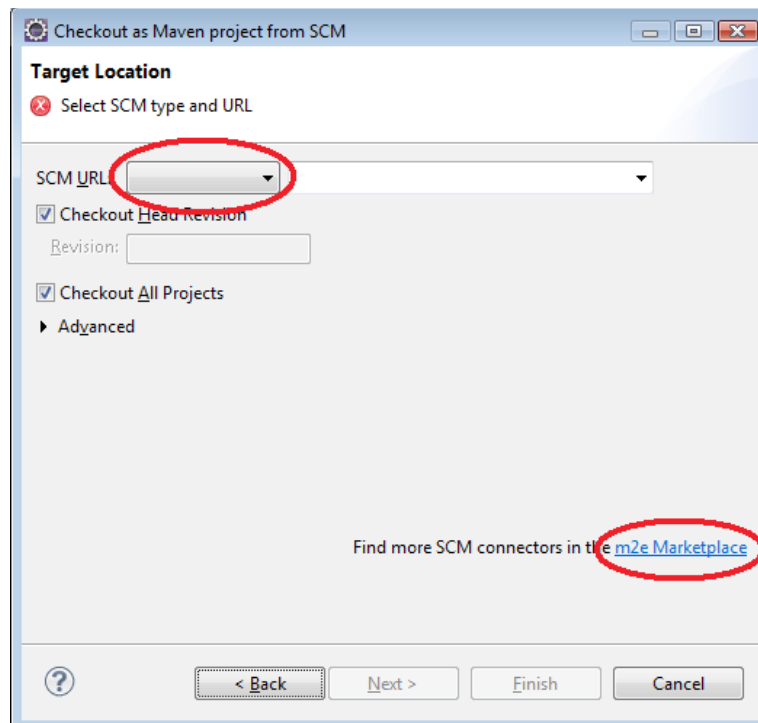
```
git config --global http.sslCAInfo ~/git-certs/ferko-certificate.pem
```

4.3.2. Uključivanje projekta iz SCM-a temeljenog na Mavenu u Eclipse

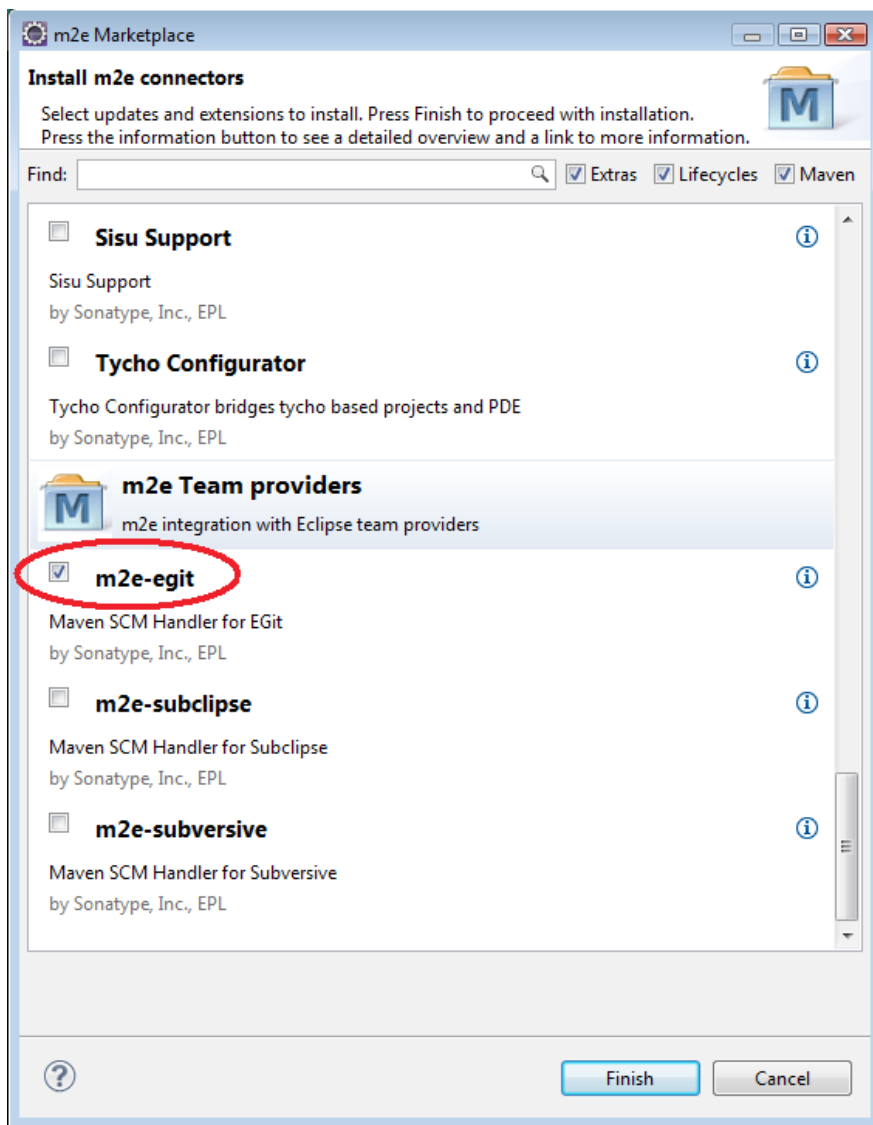
Pokrenite čarobnjaka (engl. *Wizard*) za stvaranje novog projekta (Izbornik *File*→*New*→*Other...*). U dijalogu koji se otvori odaberite: *Maven*→*Checkout Maven Projects from SCM*, kako je prikazano na slici u nastavku.



Otvorit će se novi dijalog koji omogućava izbor adrese preko koje se može pristupiti projektu koji je pod kontrolom sustava za upravljanje verzijama kôda – slika u nastavku.

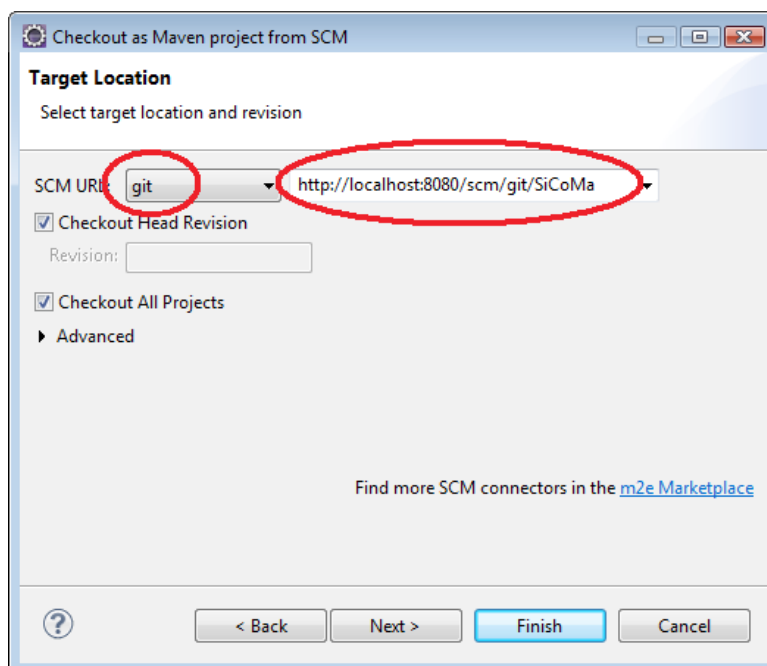


Ako pod *SCM URL* nema prikladnih providera (na prethodnoj slici crveno zaokruženo područje u gornjem lijevom dijelu slike), odaberite poveznicu "*m2e Marketplace*". Time će se otvoriti čarobnjak za instalaciju m2e konektora (slika u nastavku). Na dnu odaberite eGit konektor.



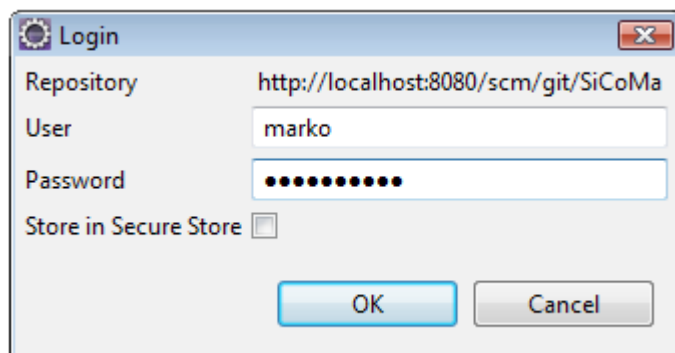
Potom kliknite na gumb *Finish*. Pokrenut će se instalacija dodatka *m2e-egit*. Po završetku instalacije Eclipse će Vas zatražiti da se restarta, pa mu to dopustite.

Sada ponovno možemo ponoviti početne korake: pokrećemo čarobnjaka za stvaranje novog projekta, biramo Maven projekt iz SCM-sustava; u dijalogu koji se otvori sada možemo odabrati *git* te upisati adresu repozitorija (na slici u nastavku prikazan je jedan primjer; Vi prilagodite adresu tako da odgovara adresi projekta koji želite unijeti u Eclipse).

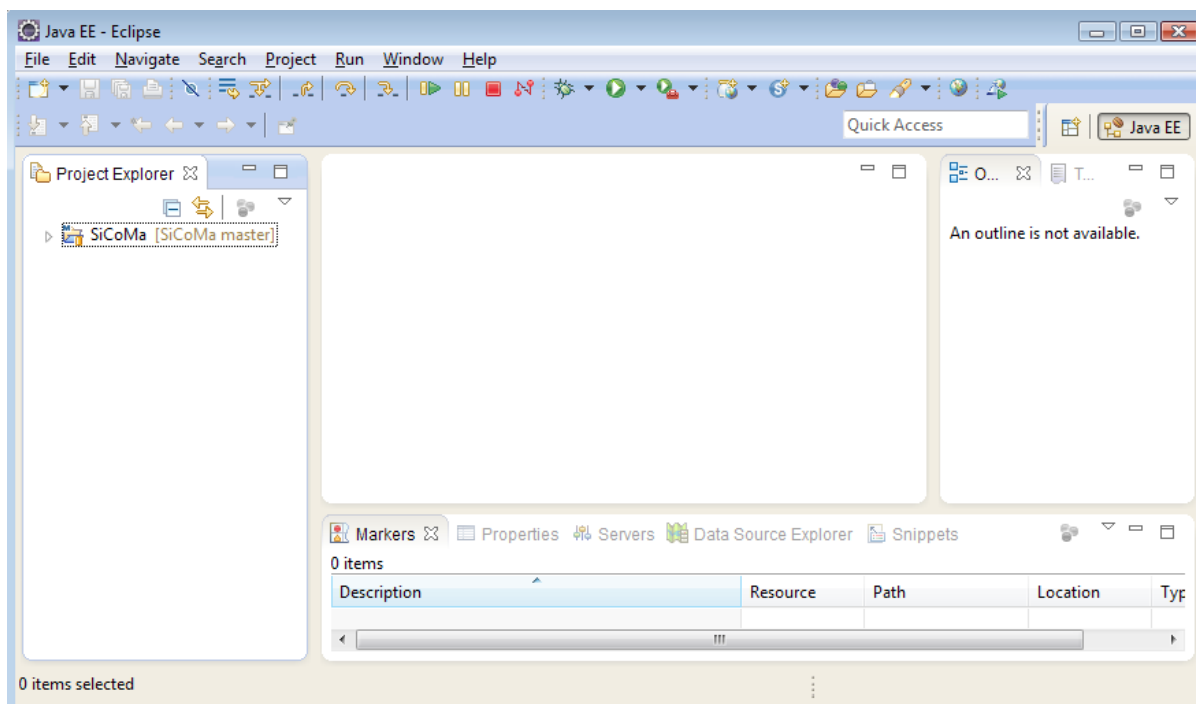


Nakon što ste unijeli adresu, kliknite na gumb *Next* pa *Finish*.

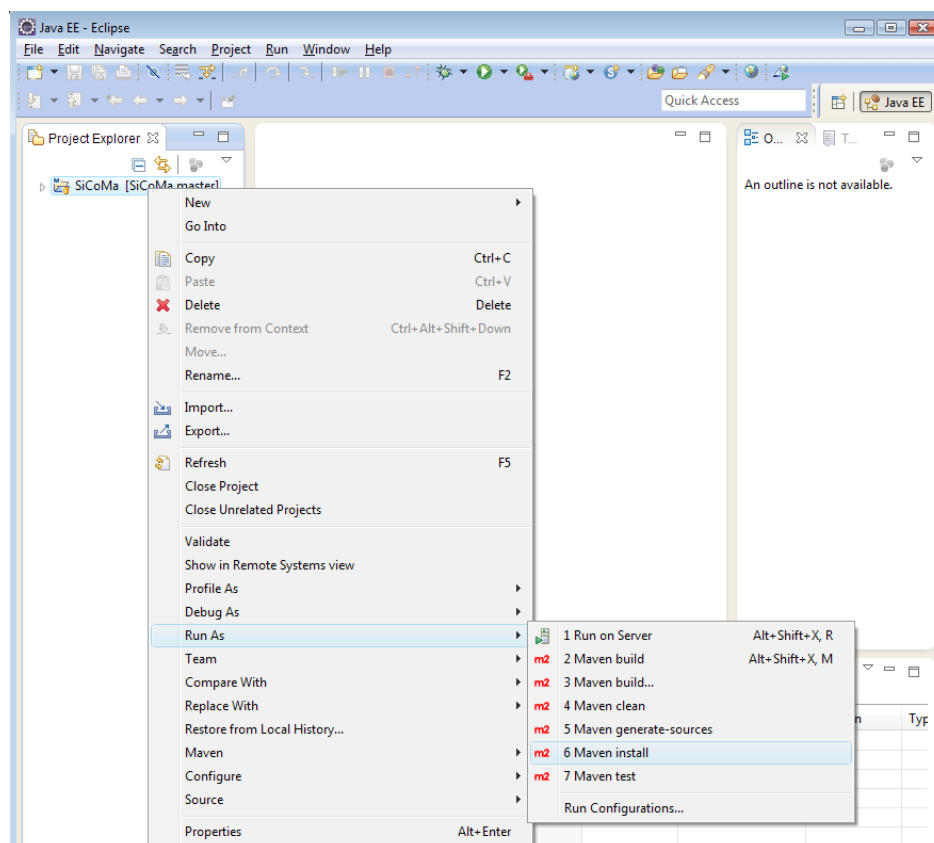
Sustav će Vas pitati za korisničko ime i zaporku (sljedeća slika). Unesite tražene podatke (podatci su isti kao i za prijavu na Ferka).



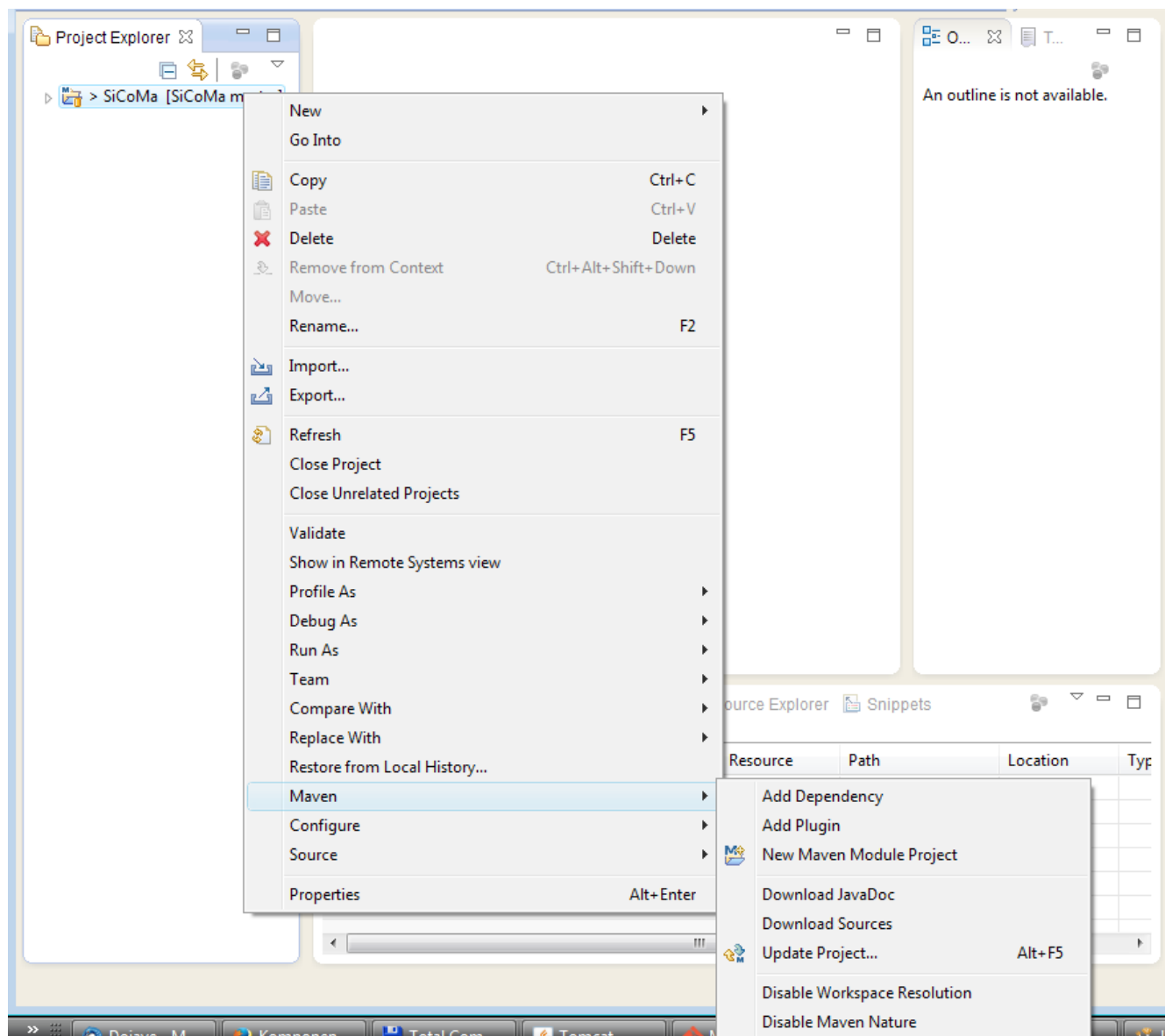
Ako je sve prošlo OK, dobit ćete novi projekt u Eclipse-u (slika u nastavku).



Nakon što modificirate projekt (izvorni kod ili druge resurse), potrebno je pokrenuti izgradnju projekta. To možete napraviti desnim klikom na projekt, pa iz iskočnog izbornika birate *Run As* pa *Maven install*.



Želite li za biblioteke o kojima projekt ovisi dohvatiti izvorne kodove (kako biste u Eclipse-u mogli pregledavati izvorni kod), napravite opet desni klik na projekt, ali ovaj puta u iskočnom izborniku potražite podizbornik *Maven*, pa u njemu odaberite stavku *Download Sources*.



Velika prednost uporabe alata *Apache Maven* nad alatom *Apache Ant* je integriran podsustav za automatsko dohvaćanje svih biblioteka o kojima projekt ovisi – sada više nema potrebe da ručno tražite po Internetu gdje se nalazi koji *jar* i da ih sve sami pohranjujete u direktorij `lib` (pa još jednom ponovite proceduru za izvorne kodove). Danas je navjeći broj popularnijih biblioteka raspoloživ u nekom od javnih repozitorija sustava Maven.

Uporaba sustava Maven donosi još jednu prednost ako na računalu razvijate više projekata: umjesto da svaki projekt dohvaća svoju vlastitu kopiju biblioteka (čime se bespotrebno troši diskovni prostor u slučaju kada više projekata treba iste biblioteke), Maven sve biblioteke drži na jednom centralnom mjestu na Vašem računalu (takozvani lokalni repozitorij). Prvi puta kada se pokrene izgradnja nekog projekta, Maven provjerava ima li u lokalnom repozitoriju sve potrebne *jar*-ove, i ako nema, dohvaća ih s Interneta iz javnih repozitorija te sprema u lokalni repozitorij. Vaš projekt te biblioteke potom

koristi direktno iz lokalnog Mavenovog repozitorija. U tom smislu, prodiskutirajmo još i razliku između nekoliko često korištenih zadataka koje radimo s Mavenom.

Naredba:

```
mvn compile
```

pokreće postupak prevođenja izvornih kodova (i sve što tome prethodi). Prevedeni kod uobičajeno će završiti u poddirektoriju `target/classes`.

Naredba:

```
mvn package
```

pokreće postupak generiranja `jar`-arhive za Vaš projekt (i sve što tome prethodi, uključivo i samo prevođenje izvornih kodova koje smo maloprije spomenuli, slično kao što to radimo u `ant`-u gdje ručno definiramo ove ovisnosti). Generirana arhiva uobičajeno će završiti direktno u poddirektoriju `target`.

Naredba:

```
mvn install
```

konačni proizvod Vašeg projekta (primjerice, `jar`-arhivu) kopira u Mavenov lokalni repozitorij kod Vas na disku: ovo je nadopuna koraka u kojem se radi pakiranje. Naime, važno je uočiti da, tako dugo dok je novi `jar` Vašeg projekta u Vašem `target` direktoriju, drugi projekti koji možda ovise o ovom projektu a koje također gradite uporabom sustava Maven taj `jar` ne vide (sjetite se, rekli smo da Maven ovisnosti najprije traži u lokalnom repozitoriju na računalu a potom na Internetu; stoga ako na računalu razvijate dva projekta od kojih drugi ovisi o prvom, svaki puta kada modificirate prvi projekt, morate Mavenu reći da novu verziju `jar`-arhive prekopira u lokalni repozitorij koji dijele svi projekti).

Konačno, naredba:

```
mvn deploy
```

predstavlja nadopunu prethodnog koraka u kojem se konačni produkt projekta dodatno objavljuje i u javnom repozitoriju na Internetu kako bi bio dostupan drugim korisnicima. Da bi ovaj korak funkcionirao, potrebno je podesiti točnu adresu javnog repozitorija u koji će produkt biti pohranjen te definirati podatke kojima će se Maven autorizirati na tom javnom poslužitelju kako bi mogao napraviti pohranu. Vi ovaj korak nećete raditi.

5. Izrada Vama dodijeljene granule

Što točno trebate napraviti, dostavio sam Vam elektroničkim putem. Na raspolaganju Vam stoji *git* repozitorij:

adresa – još nije definirana --

kojem pristupate korisničkim podacima kojima se prijavljujete na Ferka. Vaš je zadatak u tom repozitoriju oformiti istu strukturu projekta kakvu ima i projekt granule `IRG-BezierovaKrivuljaOsnovno`. U datoteci `pom.xml` prilagodite sve potrebne meta-podatke.

Za zadanu temu trebate napisati stranice (po mogućnosti više od jedne) na kojima se objašnjava gradivo granule te na kojima se izvode demonstracije koje ćete isprogramirati. Osmislite koncept (što želite reći, što želite prikazati demonstracijom, kojim redoslijedom i slično). U slučaju bilo kakvih nejasnoća, slobodno me kontaktirajte (ili još bolje - dodite uživo).

Prilikom vrednovanja projekta u obzir će ulaziti svi aspekti ostvarenog rješenja: od kvalitete programskog koda do kvalitete dokumentacije, intuitivnosti demonstracija te gramatičke korektnosti napisanog teksta. Rješenja koja nisu zadovoljavajuće kvalitetna po svim aspektima neće biti prihvaćena.

Prilikom rada s *git* repozitorijem posebno obratite pažnju što se pohranjuje u repozitorij a što ne: bilo kakav proizvod koji sam projekt generira (tipa: direktorij `target` i njegov sadržaj) **ne smije ići** u repozitorij. Također, nemojte niti slučajno datoteku `kljucevi.keystore` pohranjivati u repozitorij – ona sadrži Vaš privatni ključ i isključivo je Vaše vlasništvo. Da se to slučajno ne bi dogodilo, tražio sam Vas da tu datoteku stvorite u zasebnom direktoriju koji je izvan ovog projekta. Ali zlu ne trebalo – bolje da to još jednom naglasim.

Svaki dan kada završite s modifikacijama projekta, u repozitorij pošaljite trenutnu verziju projekta. Kada prođe rok za predaju projekta pogledat ću logove repozitorija. Ako ustanovim da do datuma `X` u repozitoriju nije bilo ničega a na dan `X+1` je nastalo 100k Java koda, neće biti dobro.

Najkasnije u trenutku predaje ovog projekta morate mi također dostaviti i potpisanu izjavu koja se nalazi u nastavku ovog dokumenta.

5.1. Isprobavanje jedne demonstracije

Prilikom razvoja demonstracije zgodno je imati mogućnost kod pokrenuti direktno u Eclipse-u bez potrebe za izradom produkcijskog paketa i deployanja na testni poslužitelj. U tu svrhu razvijena je biblioteka *SiCoMa-TestWebGUI*:

```
<groupId>hr.fer.zemris.sicoma.demo</groupId>
<artifactId>SiCoMa-TestWebGUI</artifactId>
<version>1.0-SNAPSHOT</version>
```

koja nudi implementaciju razreda `hr.fer.zemris.sicoma.demo.gui.test.JInteractiveDemoFrame`

i omogućava pokretanje jedne konkretne demonstracije. Dodajte ovu biblioteku kao *dependency* koji se koristi samo prilikom testiranja koda. Potom u `src/test/java` dodajte razred `Runner` koji u metodi `main` samo pozove `JinteractiveDemoFrame.main(...)`. Kao argument iz naredbenog retka morate predati stazu do *xml* datoteke koja sadrži opis granule te identifikator demonstracije koju želite pokrenuti. Program će potom demonstraciju prikazati u `JFrame`-u direktno u Eclipse-u pa ćete ga moći pokretati i debugirati iz Eclipse-a.

Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave
Fakultet elektotehnike i računarstva
Sveučilište u Zagrebu
Hrvatska

Izjava

JMBAG: _____

Ime: _____

Prezime: _____

Ovom izjavom potvrđujem da dopuštam da programski kôd te ostali materijali koje sam razvio/la u okviru *završnog projekta* na vještini *Osnove programskog jezika Java* u akademskoj godini *2013./2014.* budu javno dostupni (u nekomercijalne svrhe) te da se smiju koristiti u edukacijske svrhe.

Datum:

Potpis:
