Virtuaalikoneiden Java-version ohjelmisto-opas

Ari Mäkeläinen, Tuisku Polvinen, Timo Ylikännö,
Jari-Matti Mäkelä ja Sampsa Rauti

Versio 1

13. joulukuuta 2019

TURUN
YLIOPISTO

Sisältö

1 Java-koneen ohjelmistot

Koneeseen on asennettu seuraavat kehitystyökalut:

Eclipse	Ant	Scene Builder
OpenJDK	Maven	VisualVM
DrJava	Ivy	SQLite
SBT	Anltr 3 & 4	Proguard
IntelliJ Idea		

Ensisijaiseksi kehitysympäristöksi on asennettu Eclipse IDE, jonka käynnistyskuvake sijaitsee työpöydällä ja Ohjelmat-valikossa. Vaihtoehtoisena kehitysympäristönä koneelle on asennettu IntelliJ Idea. Se on Eclipseä yksinkertaisempi käyttää, muttei tue yhtä hyvin monen käyttäjän kehitystyötä. Idealla pystyy tekemään oikeastaan saman kuin Eclipselläkin, mutta se ole yhtä yleinen kuin Eclipse. Jos olet aiemmin käyttänyt IntelliJ:n muita tuotteita, on tämä versio sinulle helppo omaksua.

1.1 Eclipse

Eclipse on ilmainen graafinen kehitysympäristö, jota käytetään laajasti Java-kehityksessä. Eclipse on saatavilla kaikille käyttöjärjestelmille joilla java on käytettävissä. Sen vahvuuksia ovat erittäin kattava lisäosavalikoima ja laaja tarjonta ohjeita Internetissä. Virtuaalikoneissa Eclipsen käynnistävän pikakuvakkeen löytää työpöydältä. Ajattava ohjelma on polun `/home/utu/eclipse/eclipse` päässä.

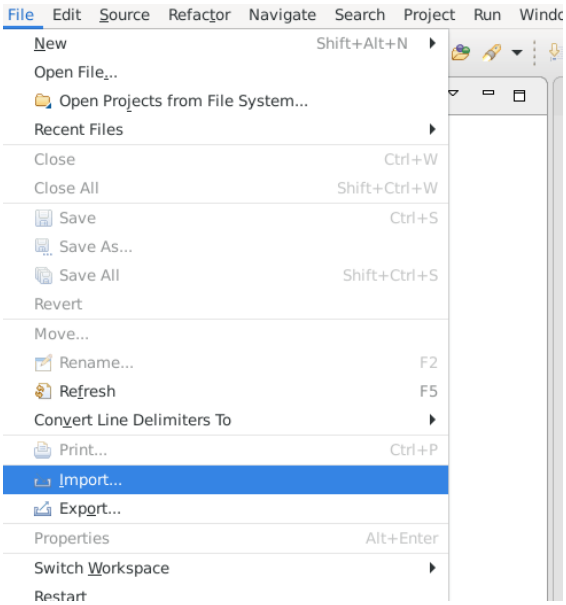
1.1.1 Maven-projektihallinta

Maven on projektin- ja riippuvuuksien hallintaan käytettävä työkalu. Maven tarjoaa ohjelmistoprojektille projektirungon (Maven:Archetype), yhtenäisen kansiorakenteen, joka ei vaihdu, vaikka vaihtaisi kehitysympäristöä sekä raportointirajapinnan. Tämä helpottaa projektien hallintaa, koska pääosin projektihenkilöstö voi käyttää omien mieltymystensä mukaisia työkaluja. Lisäksi Maven Central -portaalissa on iso joukko valmiita kirjastoja joita on helppo ottaa käyttöön Mavenin kanssa.

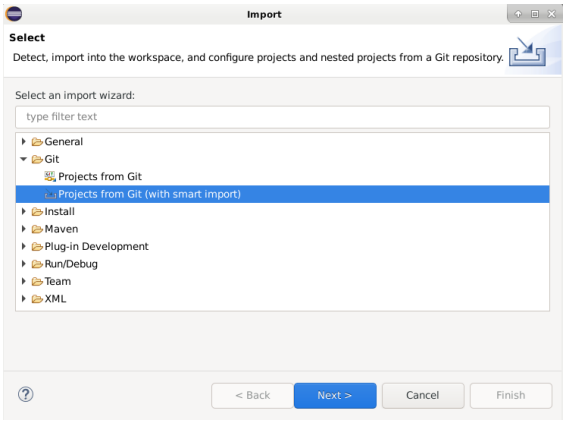
Kurssien virtuaalikoneisiin versiosta 308 eteenpäin on jo esiasennettu tarvittava plugin. Aiempia versioita ja muuta käyttöä varten Maven-liitäinnäisen asennusohjeet löytyvät dokumentin lopusta liitteestä A.

1.1.2 Maven-projektin tuominen Eclipseen

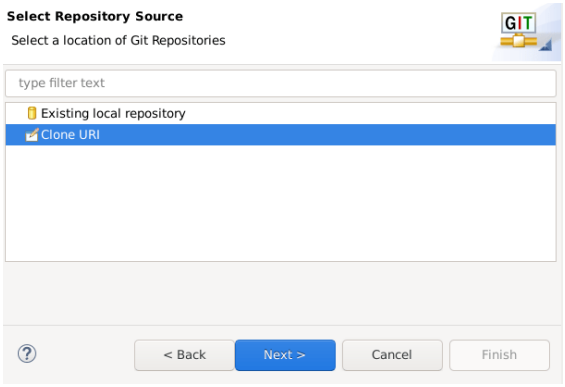
Seuraavassa kuvasarjassa käydään vaiheittain läpi Maven-projektin tuominen Eclipseen. Esimerkissä projekti tuodaan GitLabista ssh-yhteyden kautta. Jos et ole asettanut ssh-avaintasi GitLabiin, katso ohjeet git-oppaasta tai käytä tuotavan projektin http-linkkiä ssh-linkin sijaan kuvan 4 näkymässä. Jos ole jo tuonut projektin koneellesi muttet Eclipseen, esimerkiksi kloonaamalla repositorion terminaalin kautta, valitse kuvan 3 valikossa vaihtoehto *Existing local repository*.



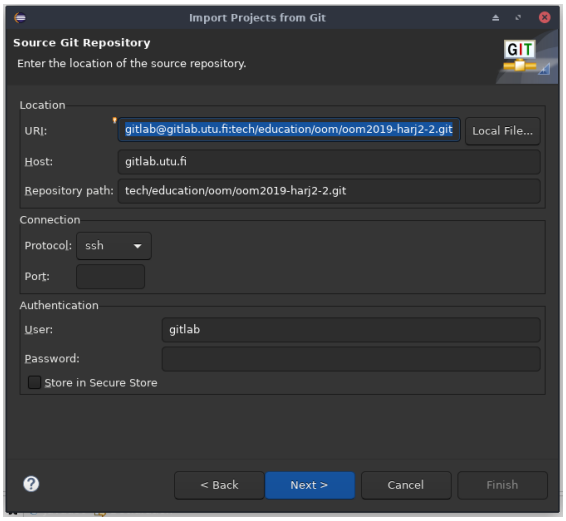
Kuva 1: Valitse *Import*



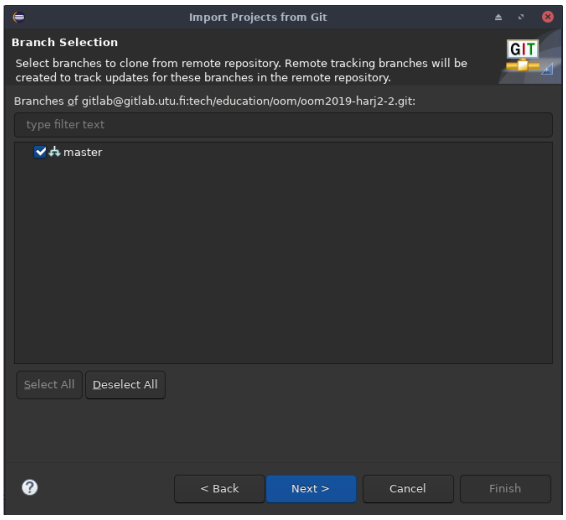
Kuva 2: Valitse *Projects from Git (with smart import)*



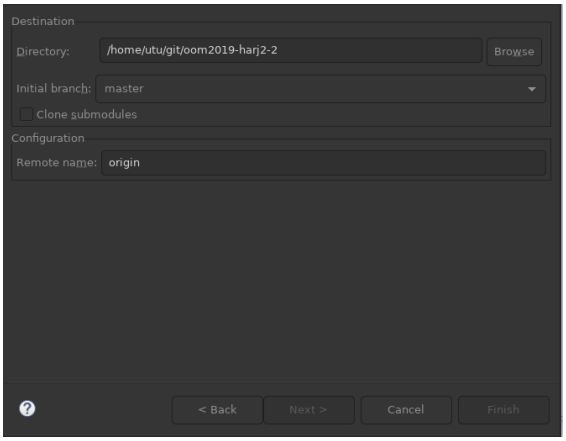
Kuva 3: Valitse *Clone URI*



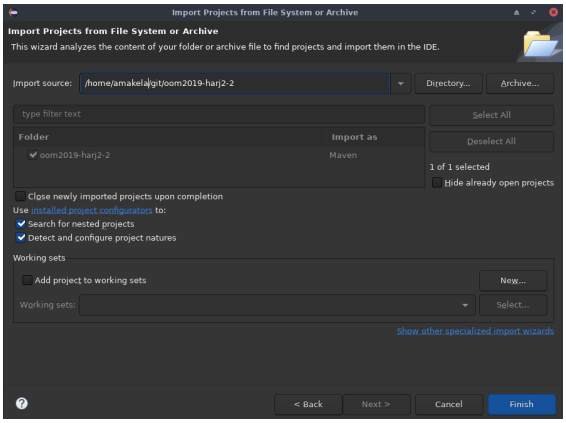
Kuva 4: Täytä git-tietovaraston tiedot



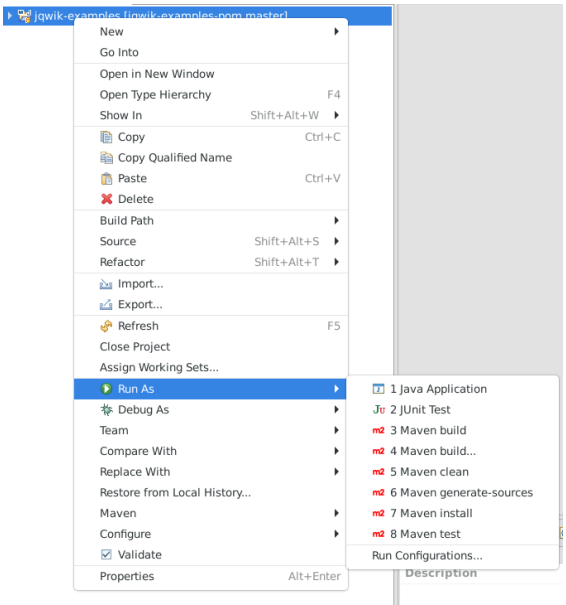
Kuva 5: Valitse käytettävä Branch, kuvassa 'master'



Kuva 6: Valitse mihin paikalliseen sijaintiin git-projekti kloonataan



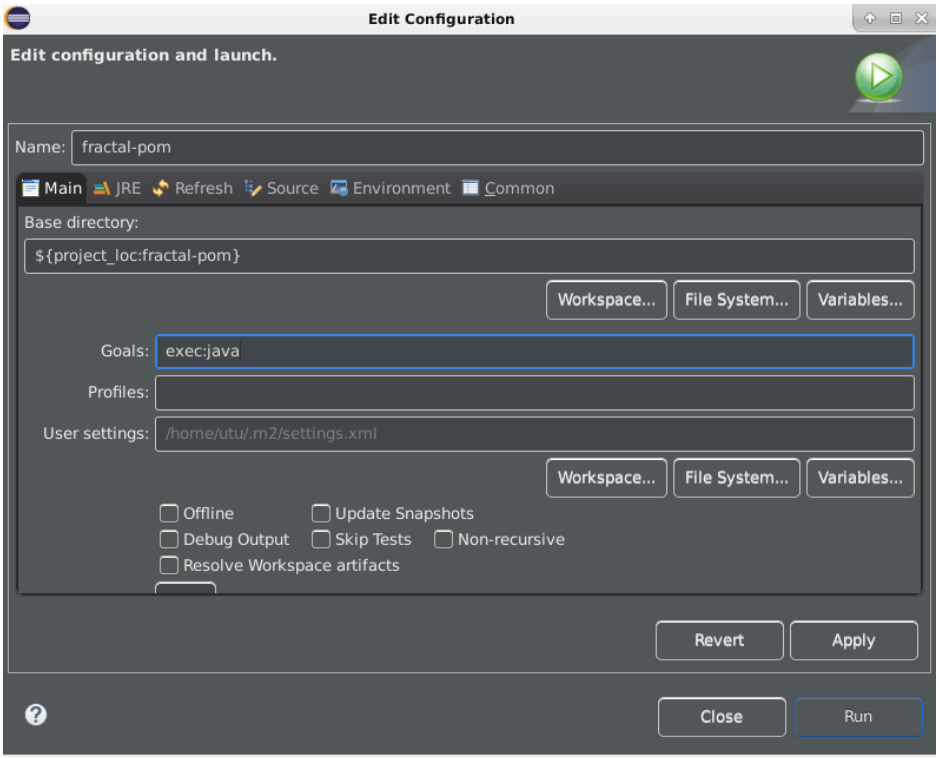
Kuva 7: Viimeiseksi voit vielä halutessasi tarkastella tuonnin tietoja. *Finish* viimeistelee tuonnin



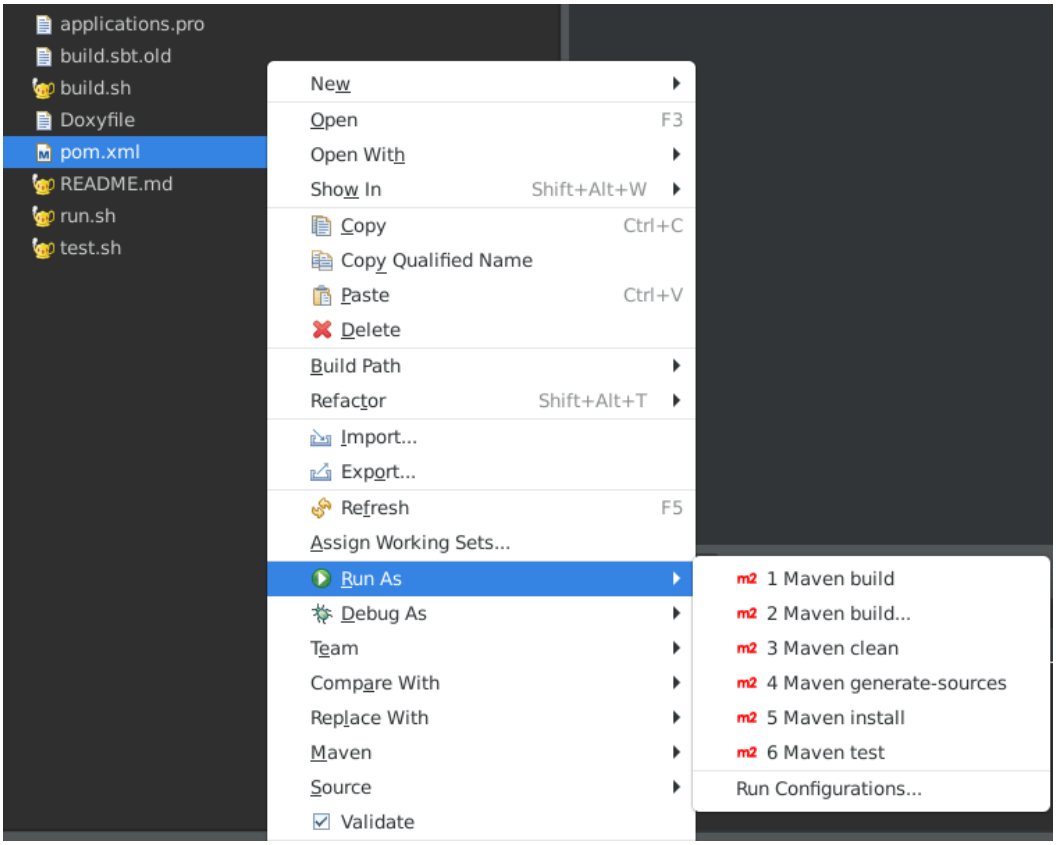
Kuva 8: Importin jälkeen projekti on täysin käytettävissä

1.1.3 Maven-projektin ajaminen Eclipsessä

Mikäli ajat Eclipse-projektisi ensimmäistä kertaa valinnalla *Maven build*, eteesi avautuu kuvan 9 mukainen valikko. Kirjoita *Goals*-kenttään `exec:java`, jolloin projekti käännetään ja ajetaan Maven-konfiguraation kautta painettaessa *Run*-painiketta. Pisteellinen valinta *Maven build...* antaa jatkossa uudelleen määrittää, mitkä tavoitteet (`goals`) suoritetaan, ja valinta *Maven build* ajaa projektin senhetkisillä tavoitteilla.



Kuva 9: Maven build



Kuva 10: pom.xml:n kontekstivalikko

Kuvan 10 Run As -valikon selitykset:

1. *Maven build* – selitetään kuvassa 9.
2. *Maven build...* – selitetään kuvassa 9.
3. *Maven clean* – tyhjennä `/target` -hakemisto ja generoitu lähdekoodi.
4. *Maven generate-sources* – määrittää, suoriteetaanko koodin automaattinen luonti mahdollisten hibermate (ottaa syötteenä tietokanta- tai XML- scheman ja luo sitä vastaavan java-ohjelman luurangon) tai muiden vastaavien sovelluskehysten (engl. framework) mukaan. (Kurssien oppimäärän kannalta tämä on irrelevanttia, mutta saattaa olla mukava tietää).
5. *Maven install* – kääntää projektin ja asentaa sen koneella olevaan Maven-pakettien tietovarastoon ("kotikansio/.m2/repository"oletusasetuksilla).
6. *Maven test* – kääntää projektin ja ajaa testit, jotka sijaitsevat projektin alla `/src/test` -hakemistossa.

2 Hajautetut ohjelmistojärjestelmät ja pilvipalvelut kurssi

2.1 Distributed-chat esimerkkiprojekti

Dockeriin tutustumista varten on rakennettu oma projektinsa. Voit asentaa projektin gitistä seuraavasti:

1. Kloonaa projekti komennolla

`git clone https://gitlab.utu.fi/tech/education/distributed/distributed-chat`

2. Vaihda kansioon komennolla: `cd distributed-chat`
3. Paketoi projekti Mavenilla: `mvn package`
4. Kun projekti on rakennettu voit käynnistää palvelimen komennolla
   `java -jar target/distributed-chat-1.0.jar [portti]`
   Portti on valinnainen, oletuksena käytetään TCP-porttia 59059.
5. Asiakas käynnistetään komennolla
   `java -jar target/distributed-chat-1.0.jar [host] [portti]`
   Host-parametri on pakollinen ja se voi olla palvelimen symbolinen nimi tai ip-osoite.

Yhdessä koneessa voi olla suorittettavana joko asiakas tai palvelin. Rajoite voidaan kiertää siirtämällä palvelin toimimaan omassa kontissaan seuraavan luvun ohjeiden avulla.

2.2 Docker-kontin rakentaminen distributed-chat ohjelmalle

Rakenna ensin kontti-image komennolla `docker build .`

Kun kontti on rakentunut onnistuneesti ('Successfully built'), listaa syntyneet imaget komennolla `docker images`

```
REPOSITORY    TAG           IMAGE ID        CREATED          SIZE
<none>        <none>        5db05b68abd9    17 seconds ago   409MB
openjdk       11.0.5-slim   c41356b393a7    2 weeks ago      401MB
```

Kopioi listasta uuden imagen id.

Anna imagelle tagi komennolla

`docker image tag 5db05b68abd9 distributed-chat:1.0`

Tarkista komennolla `docker images` että imagella on oikea nimi ja tägi. (Kohdat REPOSITORY ja TAG)

```
REPOSITORY          TAG          IMAGE ID        CREATED          SIZE
distributed-chat    1.0          5db05b68abd9    3 minutes ago    409MB
openjdk             11.0.5-slim  c41356b393a7    2 weeks ago      401MB
```

Tee seuraavaksi kontti komennolla: `docker container create distributed-chat:1.0`

Listaa kontit komennolla `docker container ls -a`

```
CONTAINER ID    IMAGE                    ...   NAMES
8b5a62568080    distributed-chat:1.0     ...   sharp_einstein
```

Käynnistä kontti komnennolla `docker contaner start <container id>`

Tarkista komennolla `docker container logs <container id>`, että kontti on toiminnassa.

```
Starting the chat server..
Listening to port 59059 at 0.0.0.0/0.0.0.0
Shut down with ctrl-c or kill the process.
```

Seuraavaksi listataan dockerin verkot komennolla `docker network ls`

```
NETWORK ID          NAME                DRIVER              SCOPE
1d61b4c9e8e9        bridge              bridge              local
3947e4c7cf11        host                host                local
f74a6e79ff90        none                null                local
```

Listaa bridged-verkosta tarkemmat tiedot komennolla `docker network inspect 1d61b4c9e8e9`


```
[
    {
        "Name": "bridge",
        "Id": "1d61b4c9e8e9b714d2d37d715d9a80337d85643880fc6af...",
        "Created": "2019-11-05T18:44:25.359992616+02:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "8b5a6256808081d657d51f2ffb207174bbccb2...": {
                "Name": "sharp_einstein",
                "EndpointID": "bb3e1299413a86f2b092dd63fdc03...",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
```

```
        "Labels": {}
    }
]
```

Listasta löydät kontin nimen perusteella (tässä tapauksessa sharp_einstein) sen käyttämän ip-osoitteen (IPv4Address). Komennolla `java -jar target/distributed-chat-1.0.jar <kontin ip osoite>` voit käynnistää chat ohjelman asiakasohjelman.

Docker muodostaa konttien ja isäntäkoneen välille virtuaalisen verkon. Aiheesta kiinnostuneille löytyy kattava selvitys dockerin käyttämistä erilaisista verkkotekniikoista osoittesta https://docs.docker.com/network/

2.3 Spawn-container

Virtuaalikoneessa on valmiina Dockerin lisäksi spawn-container. Spawn-kontit ovat Docker konetteja keveämpiä. Seuraavassa esitetään distributed-chat palvelimen imagen tekeminen ja käyttö. Aloita siirtymällä kansioon johon distributed-chat on kloonattu. Paketoi ohjelma Mavenilla antamalla komento `mvn package`.

Komennolla `spawn-container images` listataan jo asennetut imaget. Komento `spawn-container containers` listaa kontit ja `spawn-container machines` koneet. Seuraavaksi rakennetaan image komennolla `spawn-container build`. Rakentamisen jälkeen image valmistellaan komennolla `spawn-container -param CHATPORT=12345 -bridge br0 spawn distributed_chat default`. Tämän jälkeen viimeistellään kontin tekeminen komennolla `spawn-container enable distributed_chat default`.

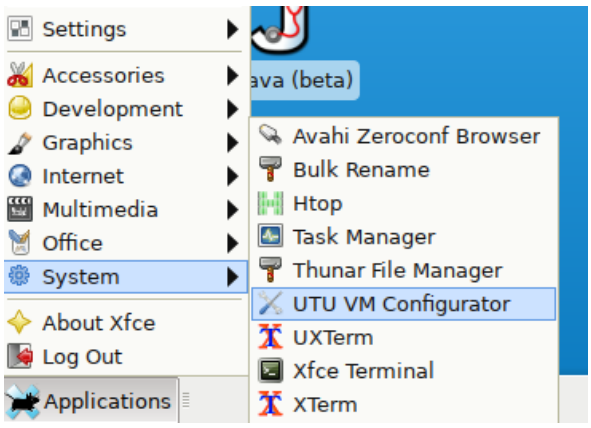
Nyt kontin voi käynnistää komennolla `sudo systemctl start spawn-container@distributed_chat-default.service` ja komennolla `sudo systemctl status spawn-container@distributed_chat-default.service` voit varmistaa, että kontti on ladattu ja aktiivinen. Viimeiseksi komennolla `spawn-container machines` näet käynnissä olevat koneet ja niihn liitetyt ip-osoitteet. Kun kontti on käynnissä, voit käynnistää distributed-chat ohjelman asiakkaan komennolla `java -jar target/distributed-chat-1.0.jar 12345 <spawn-kontin osoite>`

3 UTU VM Configurator

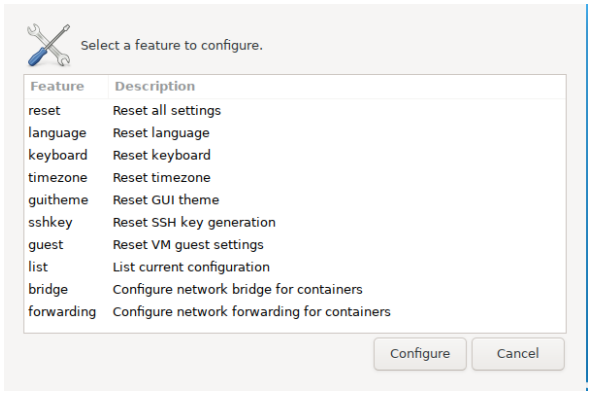
Virtuaalikoneeseen on tehty UTU VM Configurator ohjelma. Ohjelma on System-valikossa (ks. kuva 11).

Konfiguraattorin GUI:n avattuasi edessäsi on kuvan 12 mukainen ikkuna joka mahdollistaa seuraavat toiminnot:

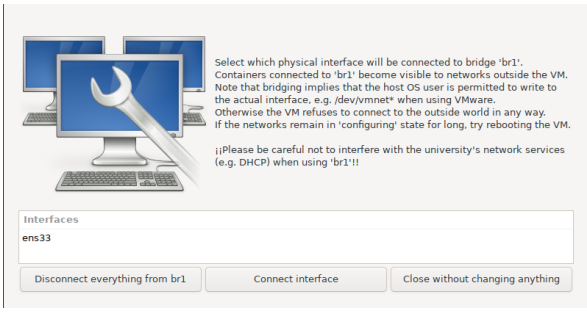
1. *reset* – Resetoi kaikki virtuaalikoneen asetukset
2. *language* – Vaihda virtuaalikoneen kieli
3. *keyboard* – Vaihda näppäimistöasetukset
4. *timezone* – Vaihda aikävyöhyke
5. *guitheme* – Vaihda työpöydän teemaa.
6. *sshkey* – Tee uusi ssh-avain ja poista vanha.
7. *guest* – Resetoidaan vierasasetukset.
8. *list* – Tulostaa käytössä olevat asetukset.
9. *bridge* – Katso kappale 3.1



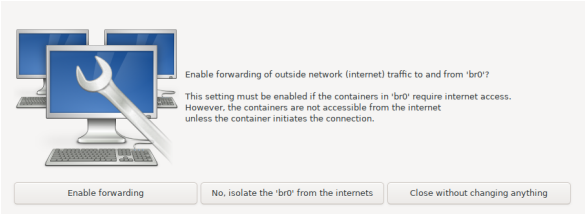
Kuva 11: Konfiguraattorin sijainti virtuaalikoneen valikossa



Kuva 12: Konfiguraattorin pääikkuna



Kuva 13: br1 konfigurointi



Kuva 14: br0 konfigurointi

10. *forwarding* – Katso kappale 3.1

Ensimmäiset kuusi toimintoa ovat samat kuin virtuaalikoneen ensimmäisellä käynnistyskerralla kysyttävät kysymykset.

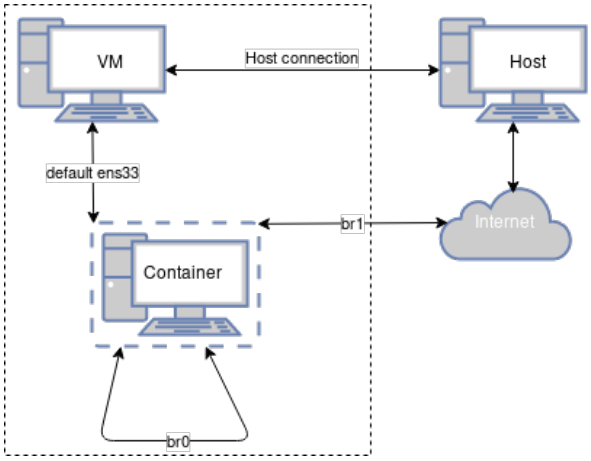
Confifuratoria voi käyttää myös terminaalissa komennolla configurator.sh. Kirjoittamalla `configurator.sh help` saat listan käytettävissä olevista parametreista. Käytännössä komennot ovat samat kuin graafisessa käyttöliittymässä. Esimerkiksi `configurator.sh keyboard`.

3.1 Virtuaalikoneen verkkoasetukset

Utu VM Configuratorin valinnoilla bridge ja forwarding voit määrittää miten koneessa olevien konttien verkkoliikennettä käsitellään. Koneessa on kaksi verkkointerfacea br0 ja br1. UTU VM Configuratorin Bridge

”Bridge” -valinnalla konttien verkkoliikenne reititetään suoraan ulos virtuaalikoneesta. Valittuasi ”bridge”, voit valita mihin verkkointerfaceen liikenne kytketään tai poistaa kytkennät br1 interfacesta. Kontit saavat verkon dhcp palvelimelta oman ip-osoitteensa, mutta ole kuitenkin varovainen, ettei virtuaalikoneen dhcp-palvelin ala jakaa osoitteita muualle verkkoon.

Valitsemalla ”forwarding”konteille tehdään oma verkko ip-osoiteavaruuteen 10.10.10.0/24. Voit valita reititetäänkö liikenne verkkoon (Enable forwarding) vai eristetäänkö verkko internetistä (NO, isolate the ’br0’ from the internets).



Kuva 15: Verkkoadapterien toimintakaavio

4 Ongelmatilanteet Java-ohjelmistossa (Eclipse jne.)

4.1 Yleisratkaisu ongelmiin

Useimmissa, ja varsinkin monimutkaisemmissa, virtuaalikoneen sisäisen ohjelmiston ongelmatilanteissa yksinkertaisin ratkaisu on tuhota virtuaalikone käyttöoppaan ohjeiden avulla ja asentaa uusi virtuaalikone.

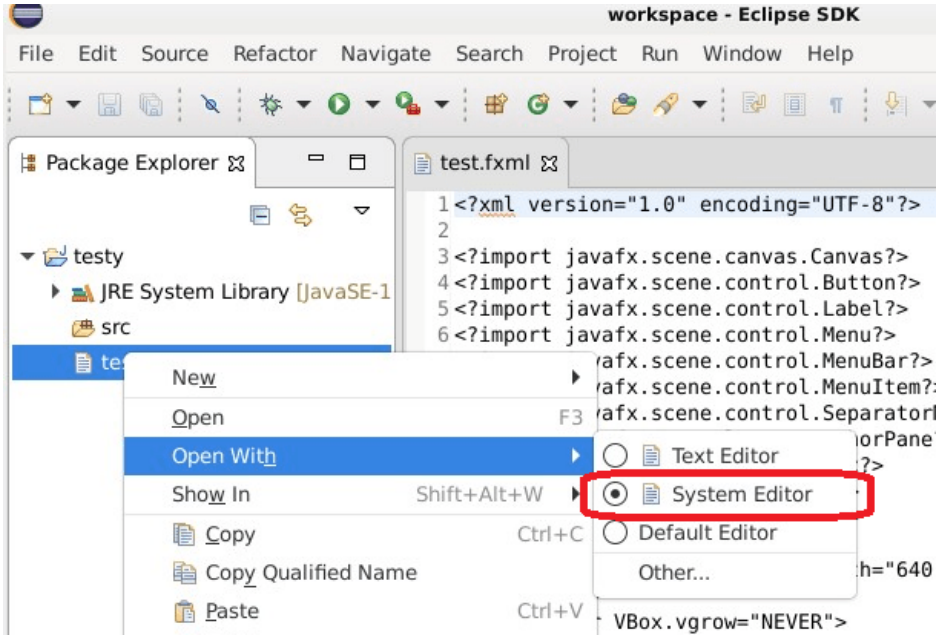
Ennen virtuaalikoneen tuhoamista varmista kuitenkin, ettet ole epähuomiossa tallentanut työtiedostojasi virtuaalikoneeseen. Käytössäsi tiedostojen tallentamiseen on Turun yliopiston pilvessä olevat Seadrive-tila ja GitLab. Halutessasi voit myös siirtää tiedostoja isäntäkoneelle koneille yhteisen jaetun kansion avulla (lue lisäohjeet wikistä VirtualBoxille). Suositeltavampaa on silti käyttää pilvitallennusta.

4.2 GitLab-palvelun yhteysongelma

Yliopiston GitLab vaatii SSH-pohjaisiin git-yhteyksiin RSA-avaimen käyttöä. Huolehdi git/ssh-ohjeita konsultoiden, että sinulla on ssh-hakemistossa `id_rsa.pub`-avain. **Älä käytä ED25519-tyyppistä avainta** (se ei ole tuettu Turun yliopiston GitLab-järjestelmässä). Avainongelma voi ilmetä myös HTTPS-yhteyksissä GitLabin kanssa.

4.3 Eclipse ei avaa Scenebuilder-tiedostoja

Jos Eclipse ei avaa Scenebuilderin tiedostoja automaattisesti, klikkaa tiedostoa, valitse *Open With → System Editor* (ks. kuva 16).

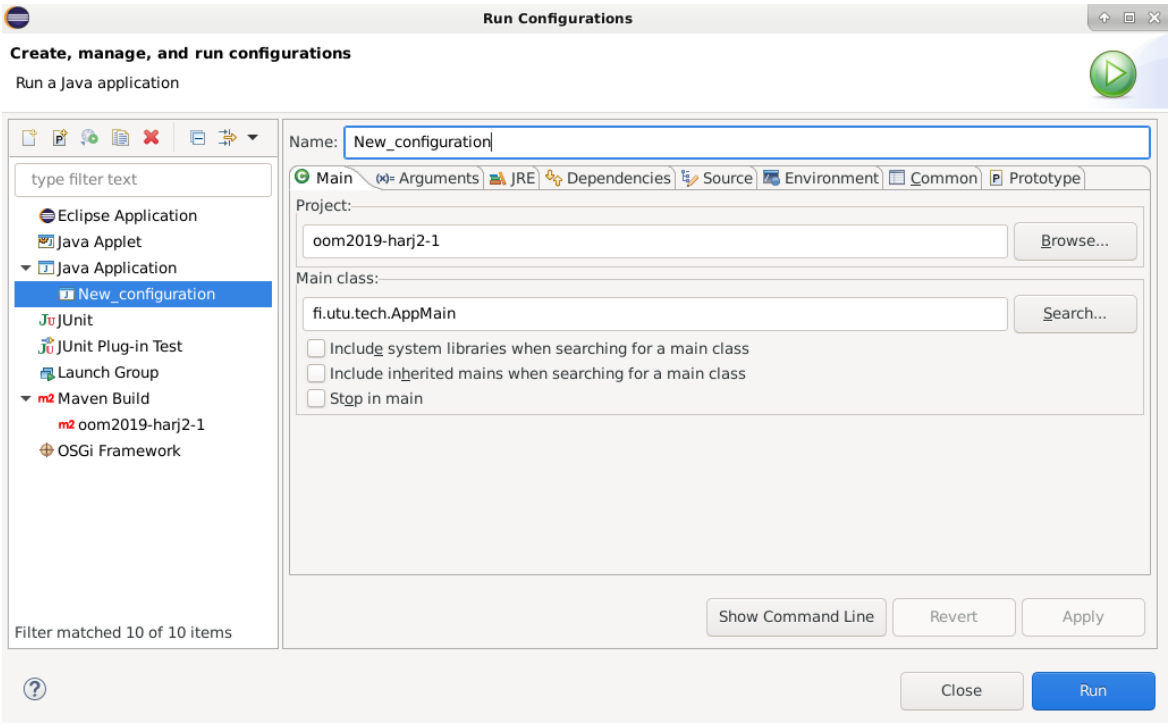


Kuva 16: Eclipsen asetusten säätäminen

4.4 Launch error: Selection does not contain a main type

Mikäli törmäät kyseiseen virheilmoitukseen, on Eclipsellä hankaluuksia löytää main-metodi. Tähän on 2 ratkaisua:

- Yksinkertaisin tapa on, että projekti suoritetaan sen juuressa sijaitsevasta *pom.xml*-tiedostosta. Tiedosto avataan valinnalla *Run As → Maven build*.
- Toinen tapa on osoittaa Eclipselle mitä luokkaa kutsutaan kun sovellus halutaan käynnistää. Ensin valitaan ylävalikosta *Run → Run Configurations*. Avautuvan konfiguraatioikkunan kohtaan *Main class* kirjoitetaan täydellinen polku luokkaan joka sisältää main-metodin. Kuvassa 17 on käytetty esimerkkinä OOM-kurssin vuoden 2019 harjoitustehtävää 2-1, joka laukaisi ajovirheen. Esimerkin polku on `fi.utu.tech.AppMain`. Asetuksen voi nimetä mielensä mukaan, esimerkissä on jätetty oletusnimi *New_configuration*.



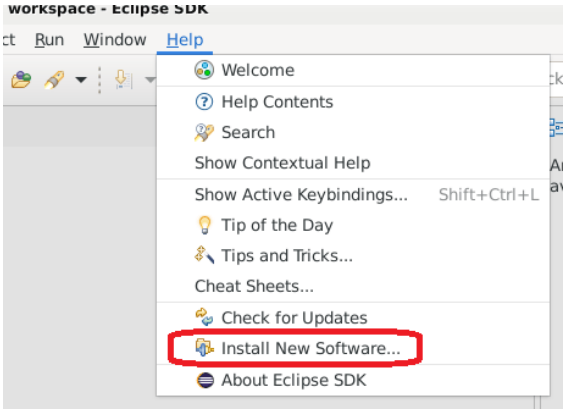
Kuva 17: Konfiguraatioikkuna

A Eclipse-liitännäisten asennus

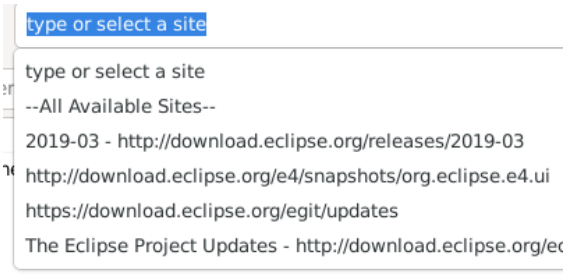
Eclipseen on mahdollista asentaa useita liitännäisiä, joiden asennus toimii pääpiirteissään samoin. Maven-liitännäisen asennus on esitelty esimerkkinä seuraavassa kappaleessa. Maven on asennettu valmiiksi virtuaalikoneisiin, joiden versio on suurempi kuin 308. Nämä ohjeet ovat esimerkkinä liitännäisten asentamisesta.

A.1 Maven-liitännäisen asennus

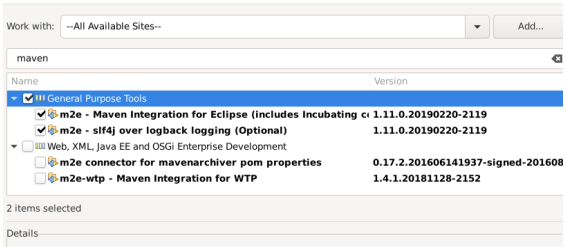
Seuraavassa kuvasarjassa käydään vaiheittain läpi Maven-liitännäisen asennus Eclipseen.



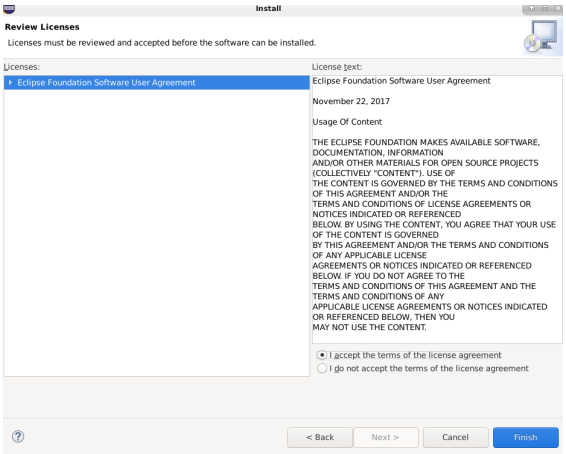
Kuva 18: Valitse *Help*-valikosta *Install New Software*



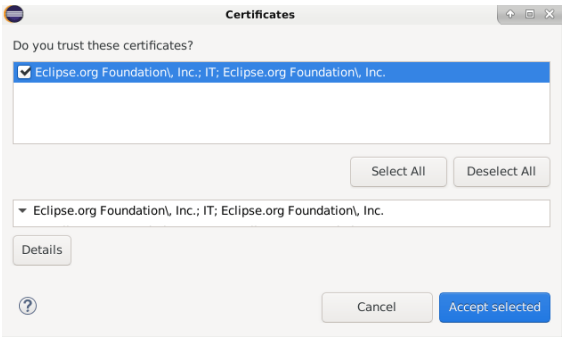
Kuva 19: Valitse *–All Available Sites–* ja kirjoita hakukenttään: maven



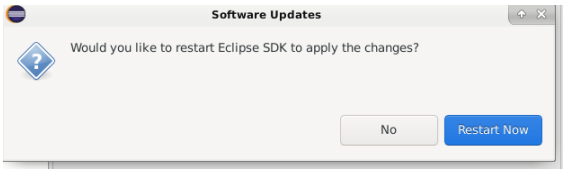
Kuva 20: Valitse General Purpose Tools



Kuva 21: Hyväksy lisenssi(t)



Kuva 22: Hyväksy sertifikaatti



Kuva 23: Käynnistä Eclipse uudelleen