

idu0080. Ülesanne 3. Asükroonne andmevahetus. Sõnumivahetusserver (JMS – Java Messaging Service).

Praktikumis kasutatav kood asub SVNis Repository location:

https://cloud.ld.ttu.ee/svn/idu0080/praktikum/branches/praktikum3_lahendus

Tehke sellele koodi „checkout“, laadige ta svn-ist oma arvutisse kui tavaline Java projekt.

1. Esialgne lahendus	2
1.1. Käivitamise järjekord:.....	2
2. Teie lahendus.....	5
2.1. Aluseks olemasolev näide – seda saab laiendada.....	5
2.2. Uued tüübid – Tellimus, TellimuseRida, Toode.....	7
2.3. Käivitamine.Objektide „castimine“(teisendus) sõnumite vastuvõtul.....	10
2.4. „Producer“-i ja „Consumer“-i ümbernimetamine.....	13

Selle näite ja ülesande sisu mõistmiseks tuleks tutvuda materjalidega mis räägivad sõnumivahetuse-teenuse (Javas platvormil JMS – Java Messaging Service) olemusest.

Mida tähendab asünkroonsus sõnumivahetuses, miks seda vaja on, kuidas JMS-teenuse kasutamine selle asünkroonsuse annab?

Mida tähendab sõnumisaatja ja sõnumite saaja lahtisidustus JMS-i kasutamise kontekstis, miks see kasulik on?

viiteid:

Praktikumi slaidid:

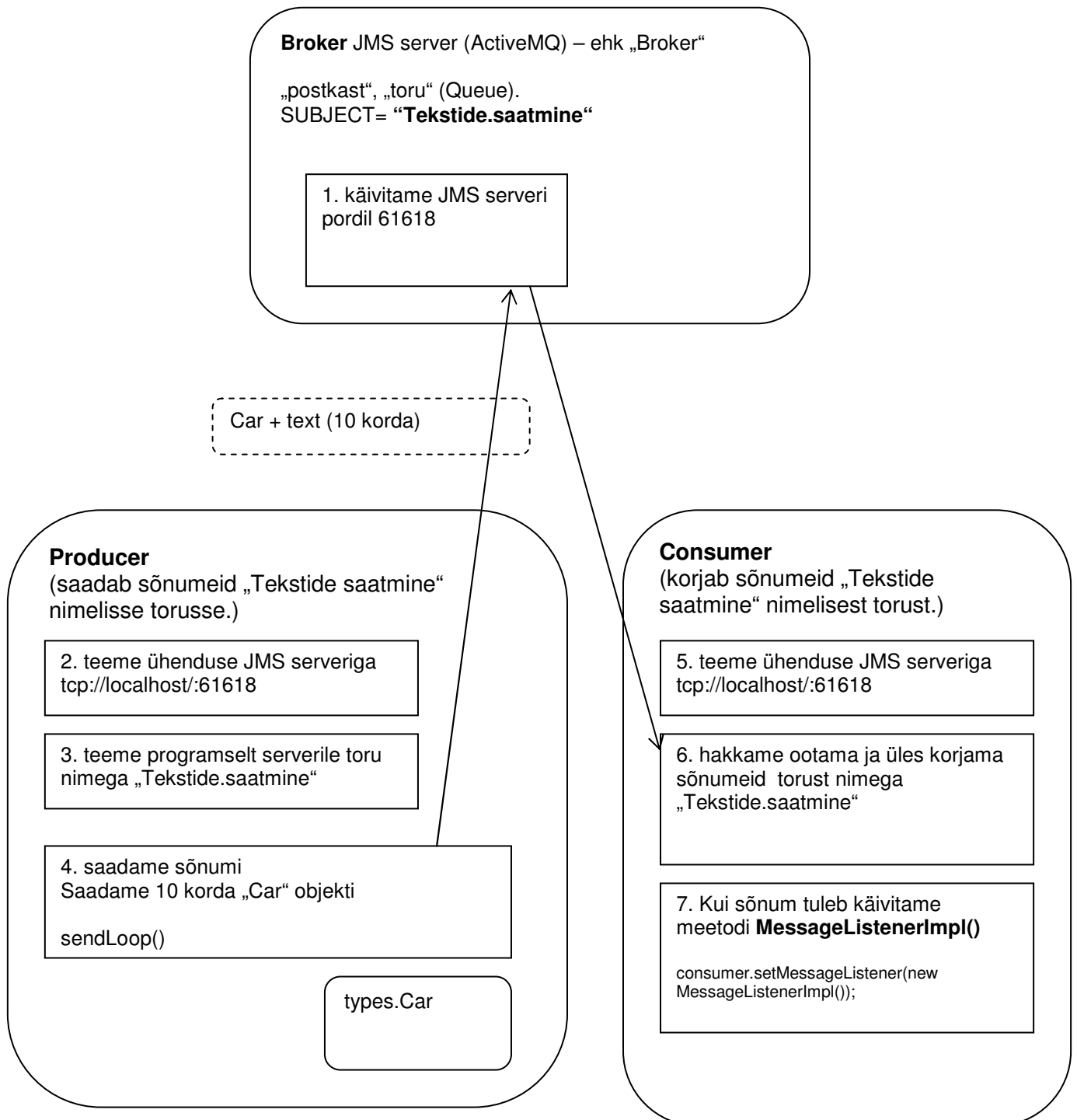
<http://cloud.ld.ttu.ee/IDU0080/LinkClick.aspx?fileticket=06CUvyVgzOc%3d&tabid=91>

Ülesande püstitus:

<http://cloud.ld.ttu.ee/IDU0080/LinkClick.aspx?fileticket=Pk-fbiW75aM%3d&tabid=91>

<http://cloud.ld.ttu.ee/idu0080/Praktikum3.aspx>

1. Esialgne lahendus



1.1. Käivitamise järjekord:

Esimesena käivitame Brokeri (JMS serveri)

Teisena käivitame Consumeri (sõnumite ootaja/üleskorjaja)

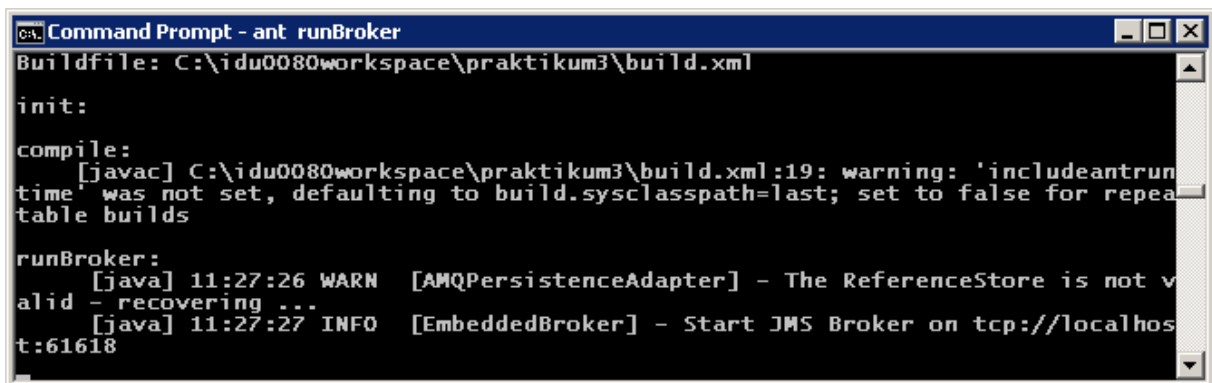
Kolmandana käivitame Produceri (sõnumite saatja)

Ülevaatlükum on käivitada kõiki kolme „tegelast“ Apache ant-i abil command line-st:

Teha lahti kolm eraldi „command line“ akent ja käivitada nii:

1. esimesest aknast käivitada JMS server

cd C:\idu0080workspace\praktikum3
ant runBroker



```
Command Prompt - ant runBroker
Buildfile: C:\idu0080workspace\praktikum3\build.xml

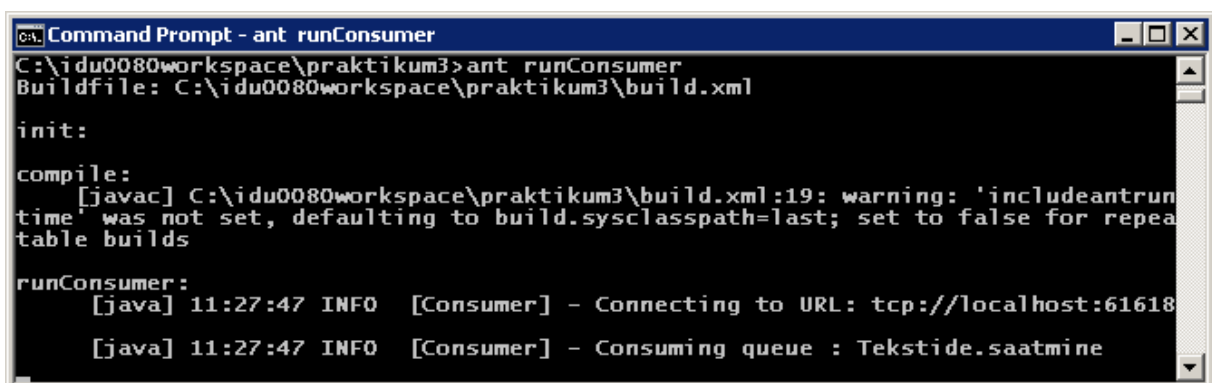
init:

compile:
[javac] C:\idu0080workspace\praktikum3\build.xml:19: warning: 'includeantrun
time' was not set, defaulting to build.sysclasspath=last; set to false for repea
table builds

runBroker:
[java] 11:27:26 WARN [AMQPersistenceAdapter] - The ReferenceStore is not v
alid - recovering ...
[java] 11:27:27 INFO [EmbeddedBroker] - Start JMS Broker on tcp://localhos
t:61618
```

2. teisest aknast käivitada sõnumite vastuvõtja.

cd C:\idu0080workspace\praktikum3
ant runConsumer



```
Command Prompt - ant runConsumer
C:\idu0080workspace\praktikum3>ant runConsumer
Buildfile: C:\idu0080workspace\praktikum3\build.xml

init:

compile:
[javac] C:\idu0080workspace\praktikum3\build.xml:19: warning: 'includeantrun
time' was not set, defaulting to build.sysclasspath=last; set to false for repea
table builds

runConsumer:
[java] 11:27:47 INFO [Consumer] - Connecting to URL: tcp://localhost:61618
[java] 11:27:47 INFO [Consumer] - Consuming queue : Tekstide.saatmine
```

3. Kolmandast aknast käivitada sõnumite saatja.

cd C:\idu0080workspace\praktikum3
ant runProducer

```
Command Prompt - ant runProducer
[java] 11:28:15 DEBUG [Producer] - Sending message: Message: 3 sent at: Sun Mar 11 11:28:15 EET 2012
[java] 11:28:16 DEBUG [Producer] - Sending message: Message: 4 sent at: Sun Mar 11 11:28:16 EET 2012
[java] 11:28:17 DEBUG [Producer] - Sending message: Message: 5 sent at: Sun Mar 11 11:28:17 EET 2012
[java] 11:28:18 DEBUG [Producer] - Sending message: Message: 6 sent at: Sun Mar 11 11:28:18 EET 2012
[java] 11:28:19 DEBUG [Producer] - Sending message: Message: 7 sent at: Sun Mar 11 11:28:19 EET 2012
[java] 11:28:20 DEBUG [Producer] - Sending message: Message: 8 sent at: Sun Mar 11 11:28:20 EET 2012
[java] 11:28:21 DEBUG [Producer] - Sending message: Message: 9 sent at: Sun Mar 11 11:28:21 EET 2012
```

Producer-i käivitamisel saadetakse nüüd 10 korda „Car“ objekti koos täiendava tekstisõnumiga JMS serverile. Consumer korjab need sõnumid üles ja trükkib vastava info väljundisse:

```
Command Prompt - ant runConsumer
[java] 11:28:17 INFO [Consumer] - Received: Message: 5 sent at: Sun Mar 11 11:28:17 EET 2012
[java] 11:28:18 INFO [Consumer] - Received: car has 5 doors
[java] 11:28:18 INFO [Consumer] - Received: Message: 6 sent at: Sun Mar 11 11:28:18 EET 2012
[java] 11:28:19 INFO [Consumer] - Received: car has 5 doors
[java] 11:28:19 INFO [Consumer] - Received: Message: 7 sent at: Sun Mar 11 11:28:19 EET 2012
[java] 11:28:20 INFO [Consumer] - Received: car has 5 doors
[java] 11:28:20 INFO [Consumer] - Received: Message: 8 sent at: Sun Mar 11 11:28:20 EET 2012
[java] 11:28:21 INFO [Consumer] - Received: car has 5 doors
[java] 11:28:21 INFO [Consumer] - Received: Message: 9 sent at: Sun Mar 11 11:28:21 EET 2012
```

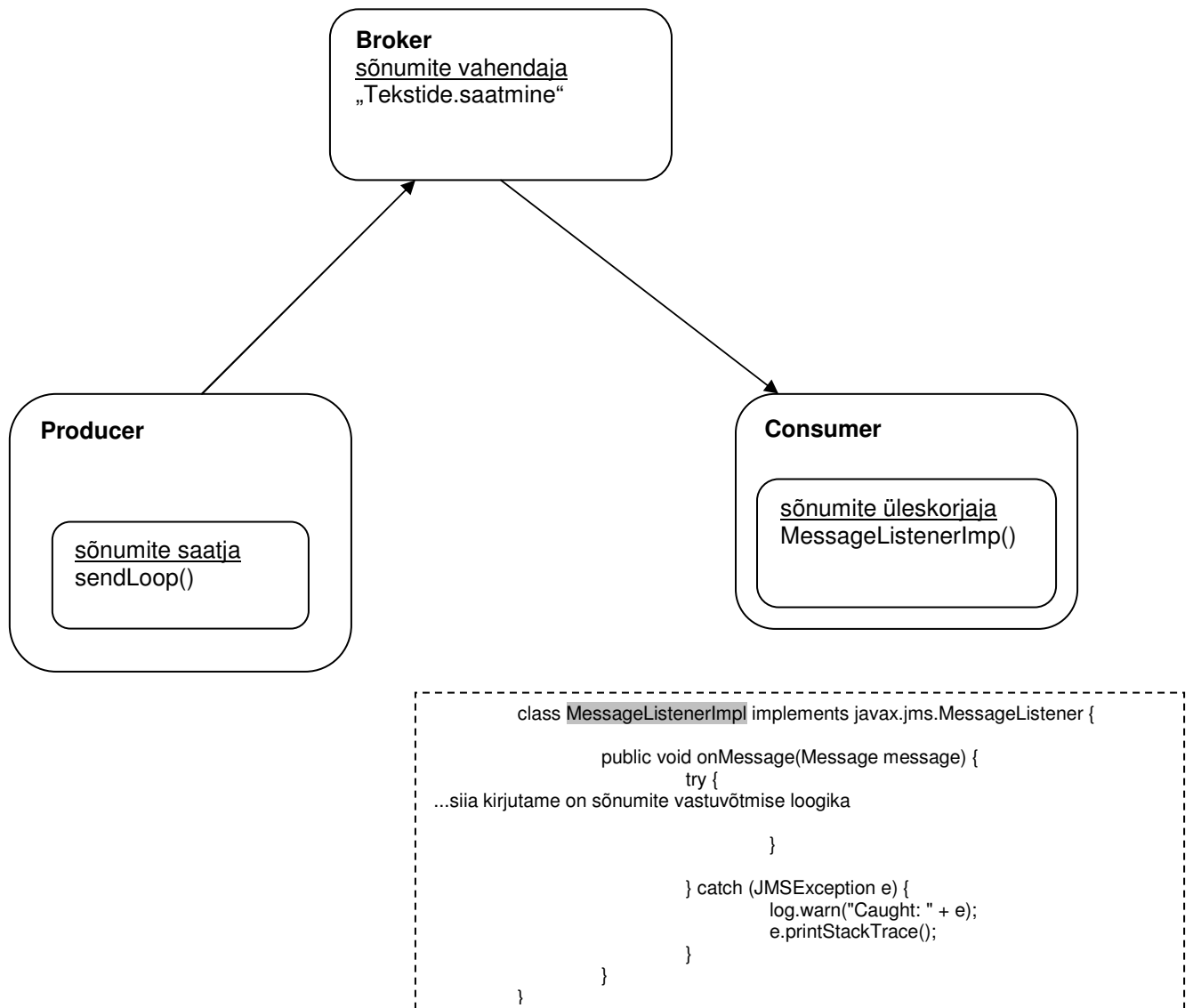
Projekt võib Eclipsest lahti olla, paralleelselt võib ant-iga käivitada kõiki kolme osapoolt.

Broker-it ei saa korraga käivitada mitu korda (näiteks Eclipsest ja ant-iga, tekib pordikonflikt (pordil 61618))

2. Teie lahendus.

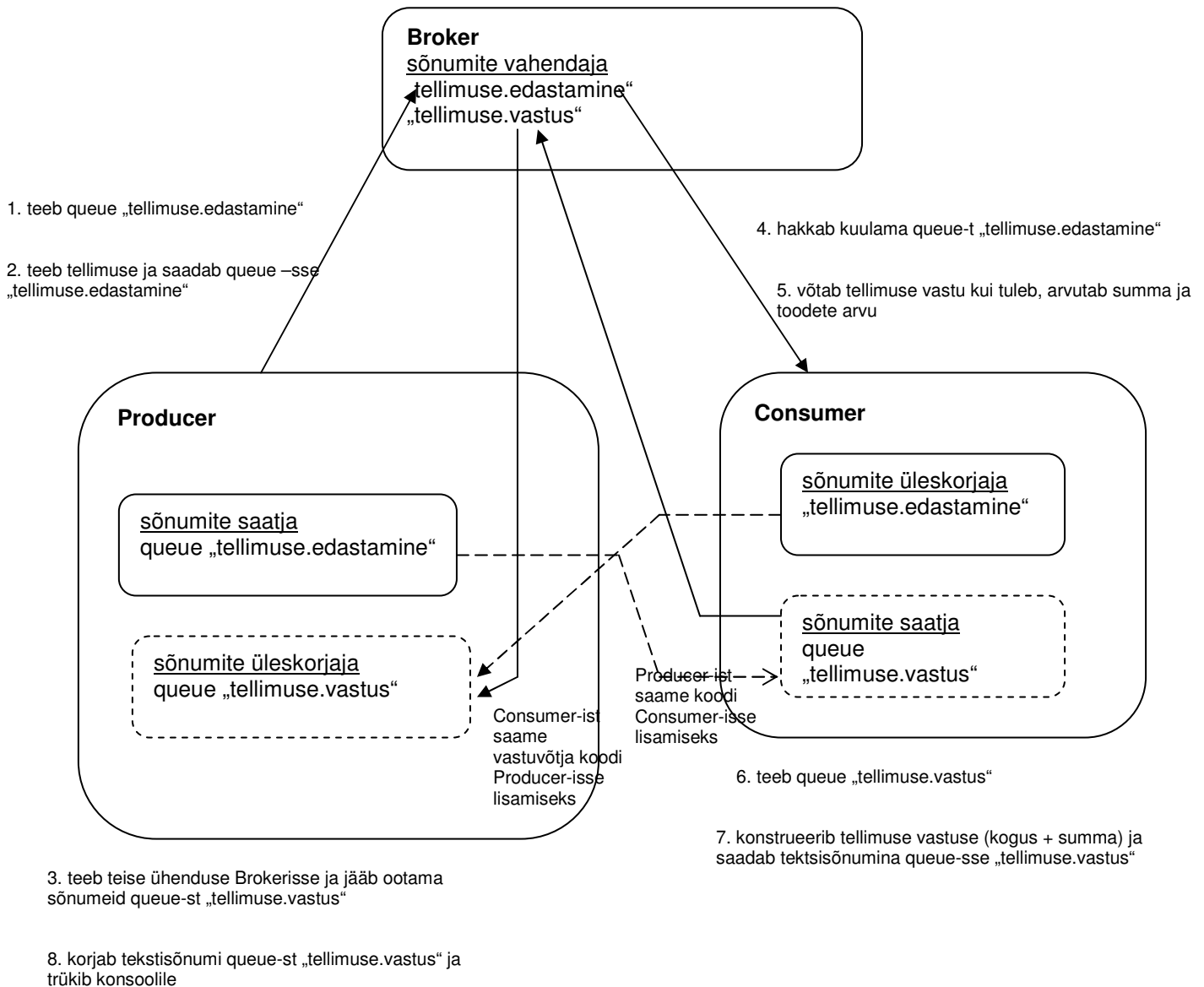
2.1. Aluseks olemasolev näide – seda saab laiendada.

Olemasolev lahendus koosneb sõnumiteenuse serverist (Broker), sõnumite saatjast (Producer) ja sõnumite vastuvõtjast (Consumer).



Teie peate tegema nii et sõnumite vastuvõtja (Consumer) saadaks (mingit teise nimega „toru“ Brokeril luues ja kasutades) sõnumite saatjale tema tellimuste kohta infot tagasi.

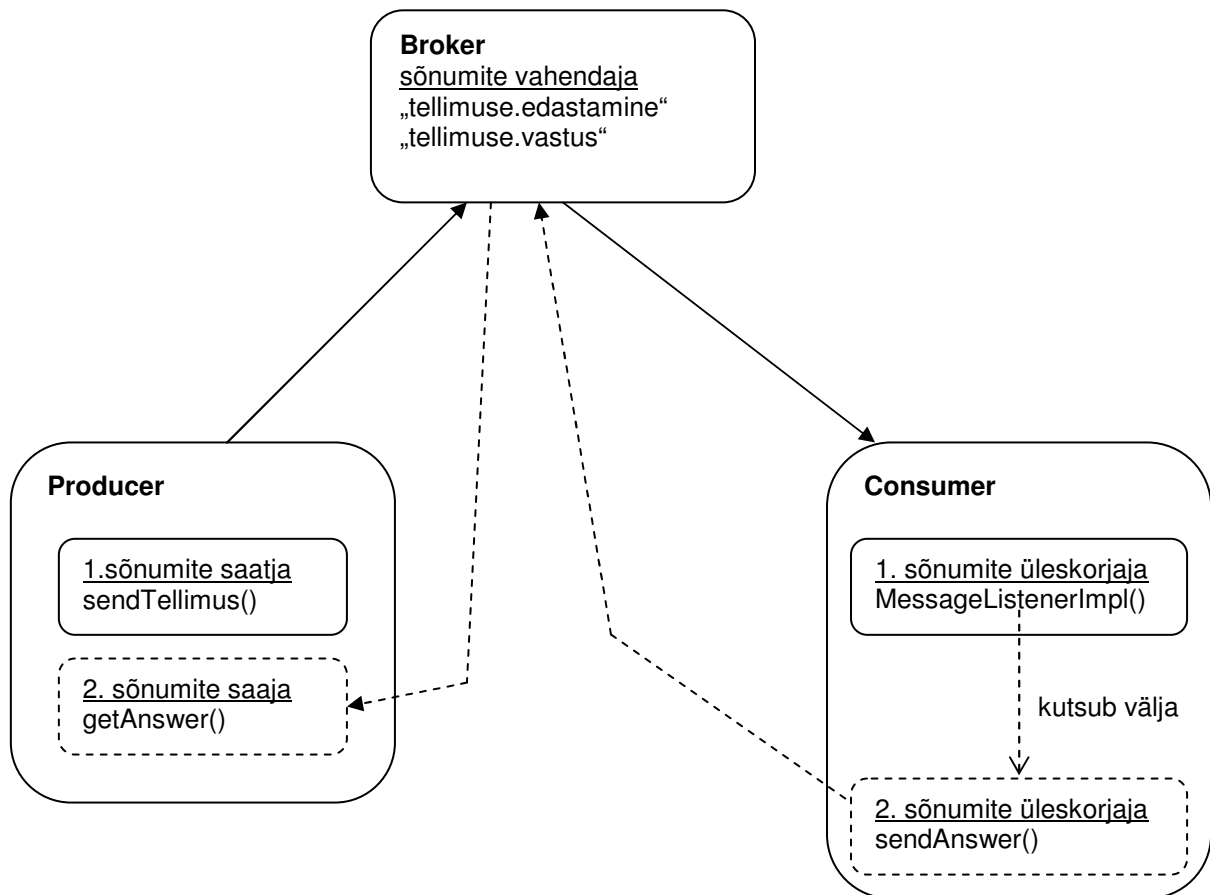
Sõnumite saatja (Producer) peab teie lahenduses selle tagasi saadetava info kätte saama ka läbi Brokeri.



Seega muutub teie lahenduses **Producer** ka sõnumite (tellimuse vastuste) vastuvõtjaks ja **Consumer** muutub ühtlasi sõnumite (tellimuse vastuse) saatjaks.

Nagu näha saate oma lahenduseks kasutada olemasolevat näidet, seda tuleb ainult laiendada. „Producer“-i kood tuleb lisada ka sõnumite vastuvõtmise funktsionaalsus ja „Consumer“-isse

tuleb lisada sõnumite saatmine. Saate põhimõtteliselt kopeerida sõnumite saatmise koodi Producer-ist ja panna see Consumer-isse ja sõnumite vastuvõtmise koodi kopeerida Consumer-ist ja panna see Producer-isse.



Mina tegin näiteks nii:

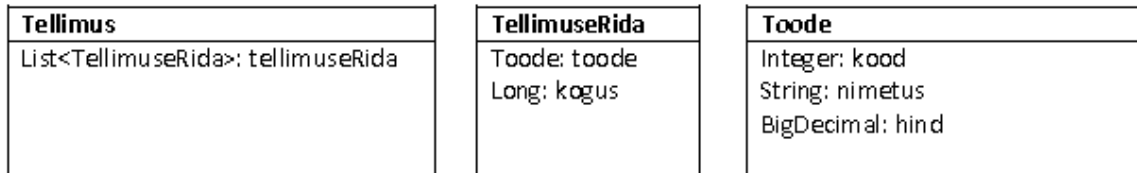
* Lisasin „**Producer**“-isse meetodi **getAnswer()** mis käivitatakse peale tellimise edastamist (**sendTellimus()**). See meetod ootab vastust queue-st „tellimuse.vastus“.

- Lisasin „**Consumer**“-isse meetodi **sendAnswer()** mida kutsutakse välja sõnumite üleskorjamise meetodist **MessageListenerImpl()** . Kui nüüd tellimus laekub siis peale tellimuse andmete konsoolile trükkimist käivitatakse iga tellimuse korral ka **sendAnswer()** mis saadab queue-sse „tellimuse.vastus“ tellimuse kogusumma ja tellitud toodete arvu.

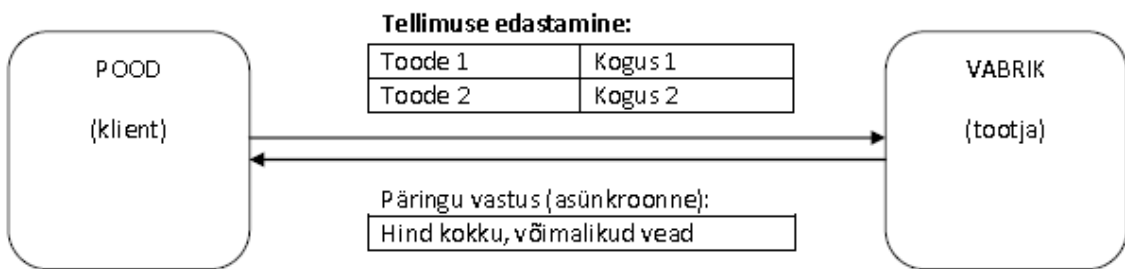
2.2. Uued tüübid – Tellimus, TellimuseRida, Toode.

Teie lahendus ei tegele enam objekti „Car“ saatmisega üle sõnumiteenuse, teie peate tegelema tellimustega.

Pood esitab tellimuse vabrikule saates JMS teate, mille teema on „tellimus.edastamine“. Teate sisuks on ObjectMessage **Tellimus**.



Vabrik töötleb tellimust 2 sekundit ja saadab vastu JMS teate, mille teema on „tellimus.vastus“. Teate sisuks on TextMessage, kus on kirjas mitu toodet telliti ja mis on tellimuse hind kokku.



Realiseeri da vähemalt kah e toote tellimuse edastamine.

Seega tuleks lisada tüüpide hulka klassid Toode, Tellimus, TellimuseRida.

Need võib teha samasse paketti kus on juba ees Car.java



Näide selle kohta milline võiks välja näha klass TellimuseRida:


```
package ee.ttu.idu0080.raamatupood.types;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

public class Tellimus implements Serializable {
    private static final long serialVersionUID = 1L;
    public List<TellimuseRida> tellimuseRead;

    public Tellimus()
    {
        tellimuseRead = new ArrayList<TellimuseRida>();
    }

    public void addTellimuseRida(TellimuseRida tellimuseRida)
    {
        this.tellimuseRead.add(tellimuseRida);
    }

    public List<TellimuseRida> getTellimuseRead()
    {
        return tellimuseRead;
    }
}
```

2.3. Käivitamine. Objektide „castimine“ (teisendus) sõnumite vastuvõtul.

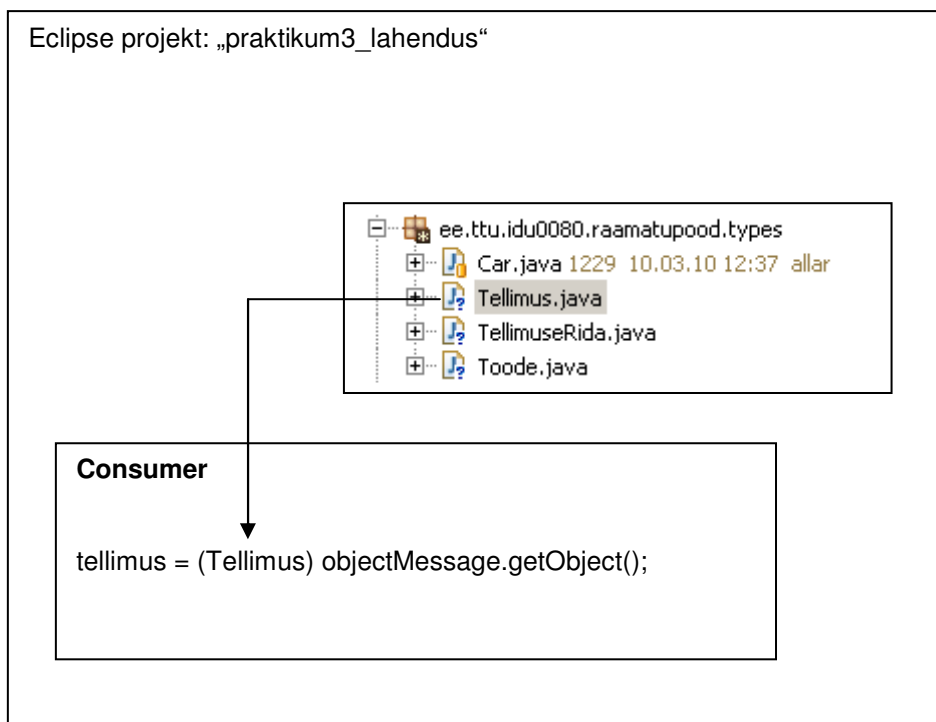
Kui laiendasite olemolevat näidet siis käib käivitamine samas järjekorras nagu esialgsel lahenduselgi: esimesena Broker, teisena Producer ja kolmandana Consumer.

Kõigi kolme osapoole kood on ühes Eclipse projektis, käivitada on parem mitte eclipsest vaid ant-iga, iga osaline oma aknas.

Kuna tegemist on ühe projektiga siis on nii saatjale (Producer) kui vastuvõtjale (Consumer) kättesaadavad tüübid

ee.ttu.idu0080.raamatupood.types.Tellimus, **ee.ttu.idu0080.raamatupood.types.TellimuseRida**, **ee.ttu.idu0080.raamatupood.types.Toode**. (Teie võite loomulikult kasutada teisi pakettide ja klasside nimetusi).

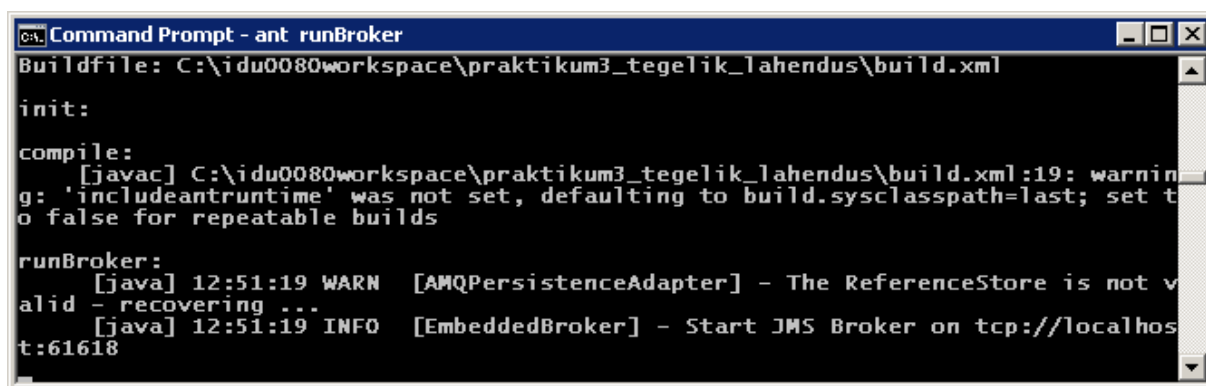
Sõnumite vastuvõtmisel tuleb Broker-ilt saadud objekt teisendada õigesse tüüpi (seepärast on oluline et saadetavate tüüpide klassid oleksid kättesaadavad ka vastuvõtu poolel, kui jookсутame nii saatjat kui vastuvõtjat ühe projekti siis siis klasside probleemi ei teki. Kui saatja ja vastuvõtja oleksid eraldi Java rakendused siis tuleks paketi **ee.ttu.idu0080.raamatupood.types** klassid teha kuidagi kättesaadavaks ka vastuvõtva rakenduse poolel (kopeerida klassid, teha kättesaadavaks .jar-failidena)



Lahenduse käivitamine võiks välja näha umbes nii:
(ant-i build.xml-i põhimõtteliselt muuta ei ole vaja, sobib näite oma – juhul kui ei ole muudetud klasside Producer ja Consumer nimesid.)

1. Käivitame Broker-i (vaatame et me ei oleks Broker-it mõnes teises aknas või Eclipse's juba eelnevalt käivitanud).

```
cd C:\idu0080workspace\praktikum3_tegelik_lahendus
ant runBroker
```

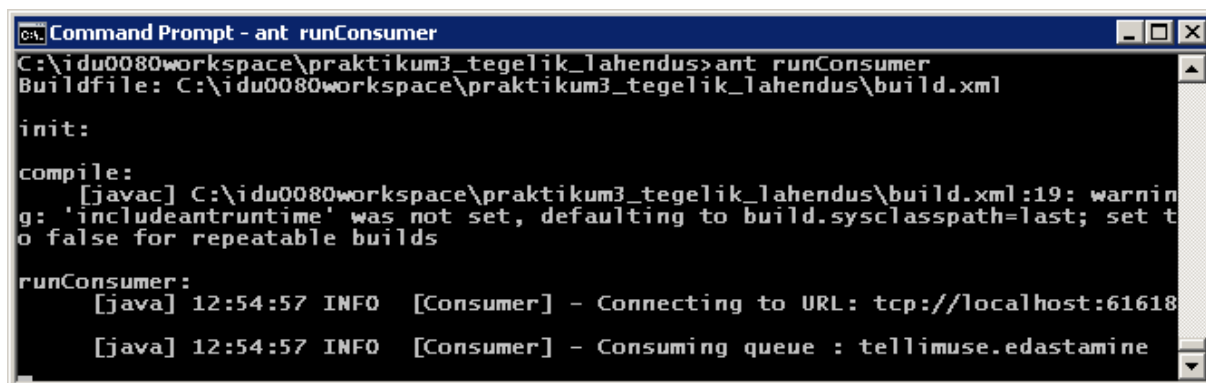


```
Command Prompt - ant runBroker
Buildfile: C:\idu0080workspace\praktikum3_tegelik_lahendus\build.xml
init:
compile:
[javac] C:\idu0080workspace\praktikum3_tegelik_lahendus\build.xml:19: warnin
g: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set t
o false for repeatable builds
runBroker:
[java] 12:51:19 WARN [AMQPersistenceAdapter] - The ReferenceStore is not v
alid - recovering ...
[java] 12:51:19 INFO [EmbeddedBroker] - Start JMS Broker on tcp://localhos
t:61618
```

2. Käivitame „Consumer“-i.

Consumer jääb nüüd ootama queue-st „tellimuse.edastamine“ saabuaid sõnumeid. Kui sõnum (tellimus) tuleb – loetakse kokku kaubad ja summad ning saadetakse tellimuse esitajale teise queue-sse nimega „tellimuse.vastus“ tekstiüüpi vastus.

```
cd C:\idu0080workspace\praktikum3_tegelik_lahendus
ant runConsumer
```



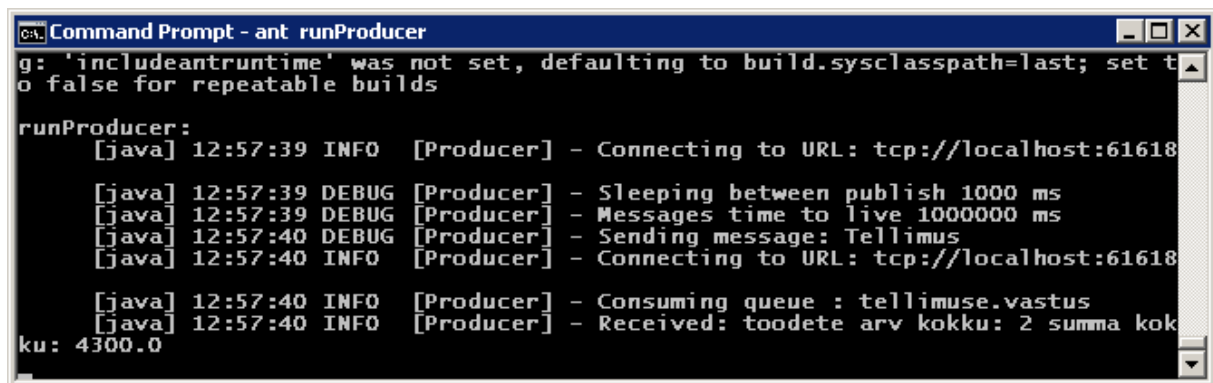
```
Command Prompt - ant runConsumer
C:\idu0080workspace\praktikum3_tegelik_lahendus>ant runConsumer
Buildfile: C:\idu0080workspace\praktikum3_tegelik_lahendus\build.xml
init:
compile:
[javac] C:\idu0080workspace\praktikum3_tegelik_lahendus\build.xml:19: warnin
g: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set t
o false for repeatable builds
runConsumer:
[java] 12:54:57 INFO [Consumer] - Connecting to URL: tcp://localhost:61618
[java] 12:54:57 INFO [Consumer] - Consuming queue : tellimuse.edastamine
```

Esiialgu jääb kuulama.

2. Käivitame „Producer“-i.

Producer saadab nüüd tellimuse milles on vähemalt 2 toodet ja jääb ootama mida vastab sellele „Consumer“.

```
cd C:\idu0080workspace\praktikum3_tegelik_lahendus
ant runProducer
```



```
Command Prompt - ant runProducer
g: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set to
false for repeatable builds

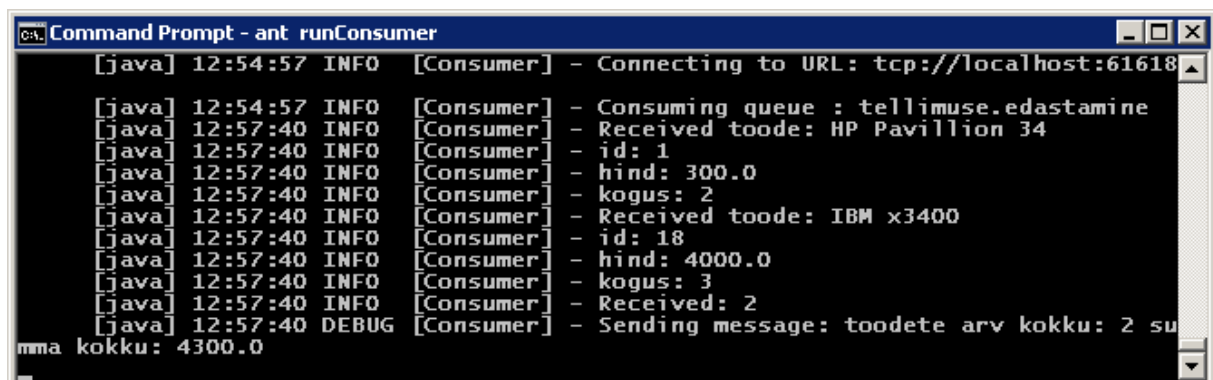
runProducer:
[java] 12:57:39 INFO [Producer] - Connecting to URL: tcp://localhost:61618
[java] 12:57:39 DEBUG [Producer] - Sleeping between publish 1000 ms
[java] 12:57:39 DEBUG [Producer] - Messages time to live 1000000 ms
[java] 12:57:40 DEBUG [Producer] - Sending message: Tellimus
[java] 12:57:40 INFO [Producer] - Connecting to URL: tcp://localhost:61618
[java] 12:57:40 INFO [Producer] - Consuming queue : tellimuse.vastus
[java] 12:57:40 INFO [Producer] - Received: toodete arv kokku: 2 summa kokku:
4300.0
```

Tellimus saadeti ja kohe saadi kätte teisest queue-st ka vastus sellele tellimusele:



```
[java] 12:57:40 INFO [Producer] - Consuming queue : tellimuse.vastus
[java] 12:57:40 INFO [Producer] - Received: toodete arv kokku: 2 summa kokku:
4300.0
```

„Consumer“-i aknas on samal ajal ka näha liikumist:



```
Command Prompt - ant runConsumer
[java] 12:54:57 INFO [Consumer] - Connecting to URL: tcp://localhost:61618
[java] 12:54:57 INFO [Consumer] - Consuming queue : tellimuse.edastamine
[java] 12:57:40 INFO [Consumer] - Received toode: HP Pavillion 34
[java] 12:57:40 INFO [Consumer] - id: 1
[java] 12:57:40 INFO [Consumer] - hind: 300.0
[java] 12:57:40 INFO [Consumer] - kogus: 2
[java] 12:57:40 INFO [Consumer] - Received toode: IBM x3400
[java] 12:57:40 INFO [Consumer] - id: 18
[java] 12:57:40 INFO [Consumer] - hind: 4000.0
[java] 12:57:40 INFO [Consumer] - kogus: 3
[java] 12:57:40 INFO [Consumer] - Received: 2
[java] 12:57:40 DEBUG [Consumer] - Sending message: toodete arv kokku: 2 su
mma kokku: 4300.0
```

Saadi kätte tellimus milles sisaldub 2 toodet, trükiti nende toodete andmed konsoolile ja konstrueeriti vastus mis saadetu queue-sse „tellimuse.vastus“.

2.4. „Producer“-i ja „Consumer“-i ümbernimetamine.

Teie lahenduse puhul on klasside „Producer“ ja „Consumer“ nimed mõnevõtta eksitavad – teie lahenduses ei saada „Producer“ ainult sõnumeid vaid võtab neid ka vastu. Teie lahenduses ei võta „Consumer“ ainult sõnumeid vastu vaid saadab neid ka ise Brokerile. Kuna lahendus tegeleb tellimuste saatmise ja vastuvõtmisega siis võiks klasside nimedeks olla näiteks vastavalt „Tellija“ ja „Raamatupood“.

Selline ümbernimetamine ei ole kohustuslik, võivad jääda ka Producer ka Consumer. Aga oleks ilus. Ümber nimetada.

