Sopimuspohjainen olio-ohjelmointi

Jouni Smed

Kevät 2007

Yleistä

- Laajuus: 5 op. (3 ov.)
- Esitiedot: Olio-ohjelmoinnin perusteet (tai ent. Ohjelmointi I)
- Ilmoittautuminen: *https://www.it.utu.fi/kurssi-ilmo/*
  - ilmoittautuminen päättyy **8.3.2007**!
- Kotisivu: *http://vco.ett.utu.fi/moodle/course/view.php?id=13*
  - kurssiavain: spoo2007

Korvaavuus

- Ohjelmointi II poistunut nykyisestä tutkintorakenteesta
- Ohjelmointi II =
  - Sopimuspohjainen olio-ohjelmointi +
  - Olio-ohjelmoinnin jatkokurssi
- OOJ järjestetään syksyllä 2007

Luennot

- 5.3.–4.4.2007
- Luentoajat
  - maanantaisin 12–14 salissa α
    - 5.3, 12.3 ja 19.3
  - keskiviikkoisin 14–16 salissa α
  - torstaisin 10–12 salissa α

Demonstraatiot 1(2)

- Neljä demonstraatiokertaa
- Ryhmät maanantaisin 10–12, 12–14, 14–16 ja 16–18
- Päivät
  - demot 1: 26.3
  - demot 2: 2.4
  - demot 3: 16.4
  - demot 4: 23.4

Demonstraatiot 2(2)

- Ei osallistumispakkoa!
- Ei minimirajoja!
- Ei kirjallisia vastauksia!

Kurssin arvostelu

- Arvostelu perustuu 30 pisteeseen
- Tentissä jaossa 28 pistettä
- Demonstraatioissa jaossa 4 pistettä
  - kullakin kerralla 1 kokonainen piste
- Ei demopakkoa:
  - 0% demoja ⇒ max. 28 pistettä
  - 50% demoja ⇒ max. 30 pistettä
  - 100% demoja ⇒ max. 32 pistettä

Kurssin arvostelu 2(2)

- Hyväksytty kurssi vaatii yhteensä 15 pistettä
- Arvosana
  - [15, 18) ⇒ 1
  - [18, 21) ⇒ 2
  - [21, 24) ⇒ 3
  - [24, 27) ⇒ 4
  - [27, 32] ⇒ 5

Tentit

- Tenttipäivät
  1. toukokuussa 2007?
  2. kesäkuussa 2007?
  3. syyskuussa 2007?
- Varmista tenttiaika ja -paikka
  http://www.it.utu.fi/opetus/tentit/
- Muista ilmoittautua ajoissa!

Kurssikirja 1(2)

Smed, Hakonen, Raita:
Sopimuspohjainen olio-ohjelmointi Java-kielellä, 2007.
ISBN 978-952-92-1776-2

http://staff.cs.utu.fi/staff/jouni.smed/SHR07-SPOO.pdf

SPOO-kurssiin tulevat kirjan luvut

- §1: johdanto, Java-kielestä
- §2: rutiinin määrittely
- §3: luokka
- §4: luokkakonaisuus
- §5.1–§5.4: periytymisen käyttö
- §6.1–§6.3: alustus, samuus
- §7.1: geneerisyyden käyttö
- §8.2: kokoelmaluokat

Kurssiaikataulu 1(2)

Kerta	Pvm	Aihe
1.	5.3 ma	Alustus, Javasta, rutiinin määrittely
2.	7.3 ke	Sopimuspohjaisuus
3.	8.3 to	Erikoistilanteiden käsittely
4.	12.3 ma	Luokan muodostaminen
5.	14.3 ke	Sisäisen esitysmuodon eheys
6.	15.3 to	Luokkakokonaisuuden muodostaminen
7.	19.3 ma	Esimerkki, testauksesta
8.	21.3 ke	Periytymisen käyttö
9.	22.3 to	Perusoperaatiot

Kurssiaikataulu 2(2)

Kerta	Pvm	Aihe
(i)	*26.3 ma*	*1. demonstraatiot*
10.	28.3 ke	Geneerisyyden käyttö, kokoelmat
11.	29.3 to	Kokoelmat
(ii)	*2.4 ma*	*2. demonstraatiot*
12.	4.4 ke	Esimerkkejä, lopetus
(iii)	*16.3 ma*	*3. demonstraatiot*
(iv)	*23.3 ma*	*4. demonstraatiot*

1. Johdanto

1. Java-kielestä
2. Käytetyistä merkkinnöistä

Javan versiohistoria

Versio	Julkaistu	Luokkia	Pakkauksia	Erityistä
1.0	toukokuu 1996	212	8	
1.1	helmikuu 1997	504	23	sisäluokat
1.2	joulukuu 1998	1520	59	Collections
1.3	toukokuu 2000	1842	76	
1.4	helmikuu 2002	2991	135	`assert`
5.0	syyskuu 2004	3279	166	geneerisyys
6.0	joulukuu 2006	3777	202	

Kääntäminen ja ajaminen

```
javac MunKoodi.java

javac -Xlint MunKoodi.java

java -enableassertions MunKoodi
```

Javadoc

- Hae kurssin kotisivulta tiedostot `common.jd` ja `project.jd`

```
javadoc @common.jd @project.jd
```

Javadocin perustäkyt

`@author`	tekijä
`@version`	versio
`@since`	mukana versiosta
`@throws`	poikkeuksen esittely
`@param`	parametrin esittely
`@return`	paluuarvon esittely
`@see`	ristiviittaus

common.jd-tiedostossa esiteltyjä lisätäkyjä

@.pre	alkuehto
@.post	loppuehto
@.classInvariant	luokkainvariantti
@.abstractionFunction	abstraktiofunktio
@.correspondence	tekijän yhteystiedot
@.download	lähdekoodilinkki
@.toDo	työn alla

project.jd-tiedosto

```
-windowtitle 'Projektin otsikko'
-doctitle 'Dokumentin otsikko'
-overview mahdollinen-projektin-yleiskuvaus.html
-bottom '&copy; 2007 Etunimi Sukunimi. All
  Rights Reserved.<p align="right">Conforms to
  Java<sup><small>TM</small></sup> 2 API
  Specification Version 6.0'

EsimerkkiTiedosto1.java
EsimerkkiTiedosto2.java
```

Käytetyistä määrittelymerkinnöistä

- Loogiset operaatiot
- Implikaatio: ==>
- Ekvivalenssi: <==>
- Universaalikvanttori: FORALL
- Eksistenssikvanttori: EXISTS
- Arvo ennen rutiinikutsua: OLD
- Rutiinin paluuarvo: RESULT

Loogiset operaatiot

Operaatio	Merkintä
negaatio	!
konjunktio	&
disjunktio	|
poissulkeva disjunktio	^
oikosulkeva konjunktio	&&
oikosulkeva disjunktio	||

Implikaatio ==>

- Ilmaisee riittävää tai välttämätöntä edellytystä

$$p \Longrightarrow q =_{\text{def}} (!p) \mid q$$

Ekvivalenssi <==>

- Tosi jos ja vain jos ehtojen totuusarvot ovat samat

$$p \Longleftrightarrow q =_{\text{def}} !(p \wedge q)$$

Universaalikvanttori: FORALL

FORALL(*alkio* : *kokoelma*; *totuusarvolauseke*)

FORALL(*muuttuja* : *muuttujan totuusehto*; *totuusarvolauseke*)

Esimerkki

```
/**
 * @.pre  FORALL(mj : lauma;
 *               mj.equals("Cow"))
 * @.post true
 */
public void muut(String[] lauma)

/**
 * @.pre  FORALL(i : 0 <= i < lauma.length;
 *               lauma[i].equals("Cow"))
 * @.post true
 */
public void muut(String[] lauma)
```

Eksistenssikvanttori: EXISTS

EXISTS(*alkio* : *kokoelma*; *totuusarvolauseke*)

EXISTS(*muuttuja* : *muuttujan totuusehto*; *totuusarvolauseke*)

Esimerkki

```
/**
 * @.pre  EXISTS(mj : parvi;
 *               mj.equals("Chicken"))
 * @.post true
 */
public void kotkot(String[] parvi)

/**
 * @.pre  EXISTS(i : 0 <= i < parvi.length;
 *               parvi[i].equals("Chicken"))
 * @.post true
 */
public void kotkot(String[] parvi)
```

Arvo ennen rutiinikutsua: OLD

```
/**
 * @.pre  true
 * @.post this.equals(OLD(this))
 */
public void konservoi()
```

Rutiinin paluuarvo: RESULT

```
/**
 * @.pre  s != null
 * @.post RESULT.length == t.length &
 *        FORALL(i : 0 <= i < t.length;
 *               RESULT[i] ==
 *                   t[(t.length - 1) - i])
 */
public int[] käännä(int[] t)
```

UML-merkinnöistä 1(2)

Notaatio	Merkitys
---------->	riippuvuus
——————>	assosiaatio (suunnattu)
---------▷	toteutus
——————▷	yleistys

UML-merkinnöistä 2(2)

Notaatio	Merkitys	Virallinen notaatio
Kana	luokka	
Lintu	abstrakti luokka	«abstract» Lintu
Lentäväinen	rajapintaluokka	«interface» Lentäväinen

Esimerkki