


Ohjelmistotuotanto

Suunnittelu 2


1



Käyttötapaalähtöinen suunnittelu

- Lähtökohdat: käyttötapaamalli, määrittelytason luokkakaavio
- Tehtävät:
 - 1. Käyttöliittymän periaatesuunnittelu
 - Mitkä käytötavat hoidetaan saman käyttöliittymän kautta
 - Look and feel
 - Periaateratkaisu käyttöliittymän ja sovelluslogiikan suhteista
 - miten käyttöliittymä erotetaan sisäluokista
 - arkkitehtuuri (MVC/ komennot /....)
 - Käytettävä liittymäkirjasto
 - Syntyy liittymäluokkia jotka vastaanottavat syötteet ja välittävät tulokset


©Harri Laine 2



Käyttötapaalähtöinen suunnittelu

- 2. Tietojen säilytykseen liittyvät periaateratkaisut
 - tiedostot, tietokannat, ohjelman ja tietokannan kommunikointi - kerrostus, eristäminen
 - kirjastot
 - sisältöluokat määrittelytason luokkien pohjalta
- 3. Tee hajautukseen ja tietoliikenteeseen liittyvät periaateratkaisut
 - sanomanvälitys, kirjastot
- Hyödynnä valmiita ratkaisumalleja (pattern) periaatepäätöksiä tehdessäsi


©Harri Laine 3



Käyttötapaalähtöinen suunnittelu

- 4. Käy läpi käytötavat:
 - 4.1 Ratkaise miten käyttötapa hoituu käyttäjän kannalta
 - 4.2 Ratkaise käyttötavan kulku järjestelmän kannalta
 - Osallistuvat oliot, luokat, palvelut, attribuutit
 - Olioiden yhteistyö - mitä palveluja käytetään
 - Lisää tarvittaessa palveluita, luokkia attribuutteja, jotta saat yhteistyön sujumaan. Täsmennä
 - Simuloi käyttötapaa luokkamallin suhteen


©Harri Laine 4



Käyttötapaalähtöinen suunnittelu

- Pitäisi saada aikaan tietty toiminto
 - Minkä olion vastuulla toiminto on?
 - Luokka, instanssi, lisää tarvittaessa
 - Onko luokalla jo tarvittava palvelu?
 - Lisää tarvittaessa
 - Onko oliolla tarpeellinen tietosisältö palvelun hoitamiseksi?
 - Lisää tarvittaessa attribuutteja, yhteydet muihin olioihin
 - Hoitaako olio palvelun yksinään vai tarvitaanko yhteistyötä?
 - Mahdollista yhteistyö, vain tunnettujen olioiden palveluita voi käyttää

©Harri Laine 5



Käyttötapaalähtöinen suunnittelu

- 4.3. Tehdyt muutokset voivat vaikuttaa aiemmin tehtyihin ratkaisuihin - iteroi
- 5. Laadi täsmällinen kuvaus luokista, attribuuteista ja palveluista
- 6. Määrittele pakkaukset, osasysteemit, yms
- Hyödynnä ratkaisumalleja ja kirjastoluokkia

©Harri Laine 6

Olioyhteistyö

- Olioyhteistyö (UML)
 - **Sekvenssikaavio (yhteistyöpolut)** (sequence diagram)
 - keskitytään erityisesti kuvaamaan operaatioiden tapahtumajärjestystä ja toimintaan liittyvien viestien kulkua. Sekvenssikaavion toinen ulottuvuus on aika.
 - **Yhteistyörakennekaavion** (collaboration diagram)
 - keskitytään kuvaamaan, sitä miten yhteistyö hyödyntää olioiden välisiä kytkentöjä

©Harri Laine

7

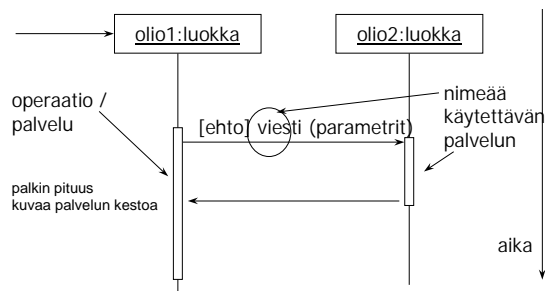
Olioyhteistyö

- **Sekvenssikaavio**
 - kuvaa sitä miten kontrolli ajallisesti etenee oliolta toiselle jotain toimintoa (esim käyttötapaa) suoritettaessa
 - mitäolioita palvelun suoritukseen osallistuu ja mitä näiden palveluja käytetään
 - missä järjestyksessä olioiden palveluita käytetään

©Harri Laine

8

Olioyhteistyö



©Harri Laine

9

Olioyhteistyö

- Viestit ovat yleensä palvelujen kutsuja (palvelupyynnöksi) =
 - palvelun nimi (parametrit)
 - käytännössä metodikutsu oliohjelmassa
 - ehto ei ole välttämätön
 - palveluun liittyvä paluunuuoli jätetään yleensä piirtämättä vaikka palaute saataisiinkin (kaavio tulee näin yksinkertaisemmaksi)

©Harri Laine

10

Olioyhteistyö

```
class A {
  B olioB;
  C olioC;
  D olioD;

  public int dolt() {
    olioB.assist1();
    olioC.assist2(olioD);
  }

  class D {...
    Public serviceD {...}
  }
}
```

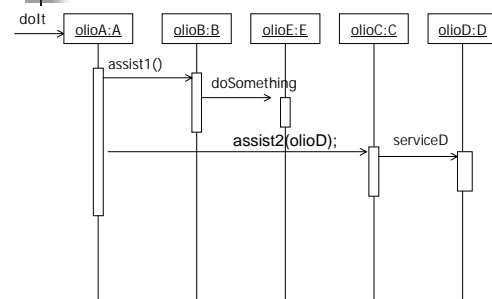
```
class B {
  E olioE;
  public void assist1() {
    olioE.doSomething();
  }
}
```

```
class C {...
  public void assist2(D od){
    od.serviceD()
  }
}
```

©Harri Laine

11

Olioyhteistyö



©Harri Laine

12

