

EDA016 Programmeringsteknik för D

Läsvecka 13: Designexempel

Björn Regnell

Datavetenskap, LTH

Lp1-2, HT 2015

13 Designexempel

- Att göra denna vecka
- Riktlinjer inlämningsuppgift
- Repetition: Vad är en algoritm?
- Design av mjukvara
- Towers of Hanoi
- Inbjuden gäst: Patrik Persson lajvkodar androidapp

Att göra i Vecka 13: Studera designexempel.

- 1 Läs följande kapitel i kursboken: 10.4, 10.5, 12.7
Designexempel: Myntvändning, Nim-spel, Hanois torn
- 2 Gör extraövningar (inkl. kolla på lösningsförslag)
<http://fileadmin.cs.lth.se/cs/Education/EDA016/exercises/extraexercises.pdf>
- 3 Träffas i samarbetsgrupper och hjälp varandra
- 4 Diskutera **inlämningsuppgiftsval** med handledare
- 5 Gör Grupplabb 11: Image Filters

Riktlinjer inlämningsuppgift

Mål: Visa att du kan ska skapa ett större program.

1 Välj bland 3 alternativ eller hitta på en egen som uppfyller:

1 Minst ca 500 rader, minst 5 klasser, gärna mer.

2 Skapa egna klasser som samverkar.

3 Använda färdiga klasser.

4 Använda en datastruktur, till exempel ArrayList.

5 Avlusa och förbättra ditt program stegvis.

2 Diskutera val av uppgift med handledare denna vecka.

3 Förbered presentation till redovisningen.

Läs mer i kompendiet på sid 89.

Repetition: Vad är en algoritm?

En **algoritm** är en stegvis beskrivning av hur man löser ett problem.

Problemlösningssprocessens olika steg (inte nödvändigtvis i denna ordning):

- 1 identifiera (del)**problemet**
- 2 Kom på en **lösningssidé**
- 3 Formulera en **stegvis beskrivning** som löser problemet
- 4 Implementera en **körbar lösning** i "riktig" kod

Det krävs ofta **kreativitiet** i alla steg ovan – även i att **känna igen** problemet.

Delar i designprocessen för utveckling av mjukvara

- **Krav: Varför? Vad?**
Intressenter, önskelmål, produktstrategier, beslut
- **Arkitektur: struktur och principiell design**
- **Design: Hur?**
Uppdelning i delproblem, vilka klasser? vilka API?
- **Implementation: Hur?**
Algoritmer, kod, implementera API
- **Testning: Är det rätt kvalitet?**
Enhetstest, Modultest, Systemtest, Acceptanstest
- **Hantera byggprocessen och olika versioner**
- **Driftsättning (eng. *Deployment*)**
- **Drift (eng. *Operation*)**
- **Support och återkoppling**

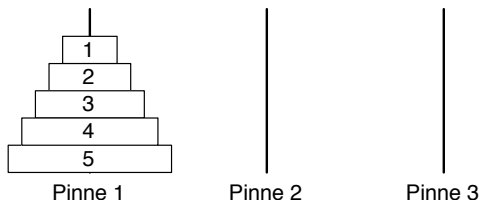
Designexempel i ankboken

- Kap. 10.4: Myntvändning – Läs själv!
- Kap. 10.5: Nim-spel – Läs själv!
- Kap. 12.7: **Hanois torn**
- (Kap. 16.6: Swing-program; mer om GUI i fk med JavaFX)

Towers of Hanoi

Designexempel: Hanois torn

Det finns tre pinnar numrerade 1, 2, 3. Från början finns n brickor av avtagande storlek på pinne 1 med den största brickan underst. Pinne 2 och pinne 3 är tomma.



De n brickorna ska flyttas så att de hamnar i **avtagande** storlek på en av de övriga pinnarna. Detta ska ske i en följd av drag där man i varje drag flyttar den **översta** brickan från en pinne till en annan pinne. **Den bricka som flyttas får aldrig placeras ovanpå en mindre bricka.**

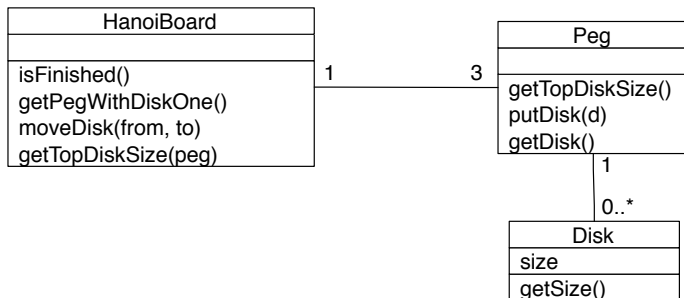
Krav

Krav: Skriv ett program som börjar med att läsa in antalet brickor som ska flyttas. Därefter ska brickorna flyttas enligt reglerna.

Utskrift (exempel med tre brickor):

```
Flytta bricka 1 från pinne 1 till pinne 2  
Flytta bricka 2 från pinne 1 till pinne 3  
Flytta bricka 1 från pinne 2 till pinne 3  
Flytta bricka 3 från pinne 1 till pinne 2  
Flytta bricka 1 från pinne 3 till pinne 1  
Flytta bricka 2 från pinne 3 till pinne 2  
Flytta bricka 1 från pinne 1 till pinne 2
```

Design: Klasser och operationer



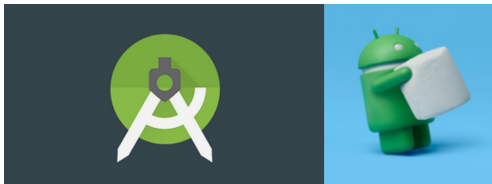
Lösningssidé

- I **udda** drag; drag nr 1, 3, ... flyttar man den minsta brickan, bricka 1, till pinnen närmast till höger. Pinne 1 anses därvid finnas till höger om pinne 3.
- I **jämna** drag; drag nr 2, 4, ... flyttar man en bricka mellan de två pinnar som inte innehåller bricka 1.

Inbjuden gäst: Patrik Persson lajvkodar androidapp

Designexempel: Skriv en app för Andorid

- Med de kunskaper ni tillgodogör er i denna kurs är det hyffsat lätt att komma i gång med utveckling av mobilappar i den integrerade utvecklingsmiljön **Android Studio**.
- Läs mer **på techworld** och **på officiella hemsidan**.
- Inbjuden gästföreläsare Patrik Persson lajvkodar androidapp i Android Studio...



Grumligtlådan

#Lappar	Ämne
6	StringBuilder
3	Vektorer, ArrayList
2	Implementering och användning av klasser
2	Sorteringsalgoritmer
2	Static
1	Arv
1	Generics
1	for-each-sats
1	Flera metoder med samma namn
1	Matris
1	När du säger "Java" exakt vad menar du då?
1	Iterator
1	Volatile Image

Övning: Dictionary

Implementera denna klass som har hand om en ordlista.
Använd en vektor `String[] words` för att spara orden.

Dictionary

```
/** Skapar en ny ordlista */  
Dictionary();  
  
/** Sätt in ett nytt ord på rätt plats i listan */  
void insertWord(String w);  
  
/** Returnerar listans ord som, skilda med mellanslag */  
String toString();  
  
/** Returnerar true om ordet finns i listan, annars false */  
boolean contains(String word);
```

Extraövning: Byt attributrepresentationen till `ArrayList<String>`