

EDA016 Programmeringsteknik för D

Läsvecka 13: Designexempel

Björn Regnell

Datavetenskap, LTH

Lp1-2, HT 2015

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.

Problemlösningens processens 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**

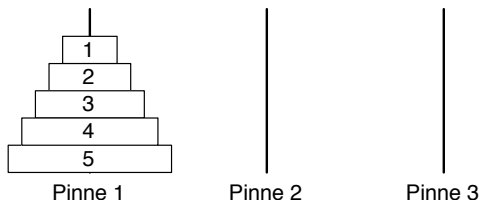
Designexempel i ankboken

- 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
```

Lösningssidé

Det visar sig att **Tornen i Hanoi** kan lösas med **denna strategi**:

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

Lösningssidé

Det visar sig att **Tornen i Hanoi** kan lösas med **denna strategi**:

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

Man kan bevisa att minsta antalet drag N med n brickor är:

$$N = 2^n - 1$$

Lösningssidé

Det visar sig att **Tornen i Hanoi** kan lösas med **denna strategi**:

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

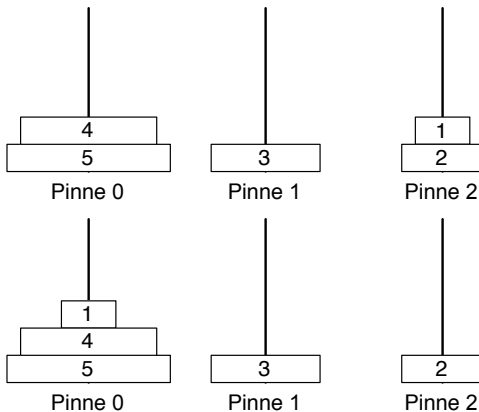
Man kan bevisa att minsta antalet drag N med n brickor är:

$$N = 2^n - 1$$

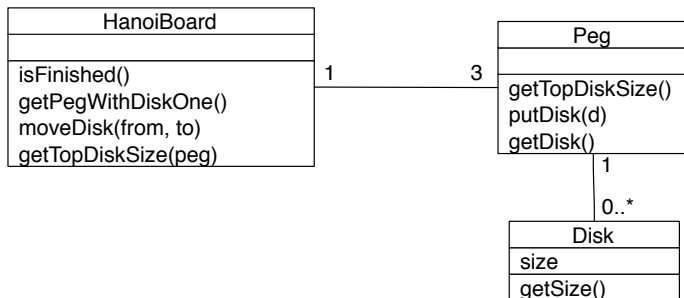
Det finns många **filmer** på **nätet** som **visar** hur **lösningen** går till.

Udda drag: flytta minstingen åt höger "cirkulärt"

Udda drag: Minstingen på pinne i flyttas till pinne $(i + 1) \% 3$



Design: Klasser och operationer



Towers of Hanoi: huvudprogrammet

```
import java.util.Scanner;

public class TowersOfHanoi {
    public static void main(String[] args) {
        System.out.print("Antal brickor: ");
        Scanner scan = new Scanner(System.in);
        int nbrDisks = scan.nextInt();
        scan.close();

        HanoiBoard board = new HanoiBoard(nbrDisks);
        HanoiStrategy strategy = new HanoiStrategy(board);
        strategy.moveDisks();
    }
}
```


Specifikationer Disk och Peg

Övning: Implementera Disk och Peg enligt specifikationerna.

Disk

```
/** Skapar en bricka med storleken size */  
public Disk(int size);  
  
/** Tar reda på brickans storlek */  
public int getSize();
```

Peg

```
/** Skapar en tom pinne */  
public Peg();  
  
/** Tar reda på numret på den översta brickan, Integer.MAX_VALUE om pinnen är tom */  
public int getTopDiskSize();  
  
/** Läger brickan d överst på pinnen */  
public void putDisk(Disk d);  
  
/** Hämtar och tar bort den översta brickan */  
public Disk getDisk();
```

Specifikation HanoiBoard

HanoiBoard

```
/** Skapar en spelplan med tre tomma pinnar, lägger nbrDisks brickor på den
 * första pinnen
 */
public HanoiBoard(int nbrDisks);

/** Tar reda på numret på pinnen som innehåller bricka 1 */
public int getPegWithDiskOne();

/** Undersöker om spelet är slut dvs om alla brickorna ligger på en annan
 * pinne än den första
 */
public boolean isFinished();

/** Tar reda på storleken av den översta brickan på pinne nummer peg,
 * Integer.MAX_VALUE om pinnen är tom
 */
public int getTopDiskSize(int peg);

/** Flyttar den översta brickan från pinne nummer from till pinne nummer to
 */
public void moveDisk(int from, int to);
```

Övning: Implementera HanoiBoard enligt specifikationen.

Specifikation HanoiStrategy

HanoiStrategy

```
/**
 * Skapar en strategi för att flytta brickor på spelplanen board
 */
public HanoiStrategy(HanoiBoard board);

/**
 * Flyttar brickorna på pinne 1 till en annan pinne
 */
public void moveDisks() ;

/**
 * Flyttar en bricka enligt reglerna i drag nummer moveNbr, skriver ut
 * flyttningen
 */
private void moveOneDisk(int moveNbr);
```

Övning: Implementera HanoiStrategy enligt specifikationen.

Ledning: se pseudo-kod på nästa bild.

Lösning, pseudo-kod

```
public void moveDisks()
```

- `moveNbr = 1`
- så länge spelet inte är slut
 - `moveOneDisk(moveNbr)`
 - `moveNbr++`

```
private void moveOneDisk(int moveNbr)
```

- om `moveNbr` är udda:
 - tag reda på numret på pinnen där bricka 1 finns
 - flytta den översta brickan från denna pinne till pinnen närmast till höger modulo 3
- annars:
 - tag reda på numret på pinnen där bricka 1 finns
 - räkna ut numren på de båda andra pinnarna
 - flytta den minsta brickan mellan dessa pinnar

Lösning, implementation

Se hela lösningen här:

[https://github.com/bjornregnell/lth-eda016-2015/
tree/master/lectures/examples/eclipse-ws/
lecture-examples/src/week13/hanoi](https://github.com/bjornregnell/lth-eda016-2015/tree/master/lectures/examples/eclipse-ws/lecture-examples/src/week13/hanoi)

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...

