

## Lab 6: Kodgranskning och refactoring

**Notera!** Denna uppgift är inte längre obligatorisk, men kommer att ge 4 bonuspoäng på tentan. Detta motsvarar ca 10% av tentans sammanlagda poäng.

Labben har flera målsättningar:

- Läsa Javakod, specifikt klasser och arvsmekanismen som tillåter en klassdeklaration att bygga vidare på en annan klassdeklaration.
- Använda grundläggande refactoring-mönster och refactoring-stödet i Eclipse.
- Fundera kring kvalitativa egenskaper hos kod utöver korrekthet, såsom namngivning, metoders längd, läsbarhet, etc.

Arbeta gärna i grupper om 2 personer, men inte fler.

Deltagande är *obligatoriskt* och uppgifterna skall redovisas för en handledare! *Användande av Eclipse-IDE:n är obligatoriskt!*

1. Skapa en arbetskatalog lab6 i din ioopm-katalog.
2. Kopiera de utdelade filerna för den aktuella labben från Mercurial-repositoryt till den katalogen.
3. Läs koden för programmet och reflektera över vad det gör och hur det fungerar. Kommentera det kortfattat med JavaDoc så att ni själva förstår vad de olika klasserna och metoderna gör.
4. Bland de utdelade filerna finns ett *kodgranskningsprotokoll*. Gå igenom koden med hjälp av granskningsprotokollet och identifiera eventuella fel och brister. Notera dem. (Du kommer att behöva redovisa dessa anteckningar för en handledare!)
5. Använd det inbyggda stödet för refactoring i Eclipse för att se till att variabler, klasser och metoder har vettiga namn som avspeglar deras innebörd eller användning. Fundera över lämpliga mönster/standarder som gör koden mer läsbar. Nu ligger alla klasser i en fil – det är inte lämpligt. Fundera över paketstruktur och inre/nästlade klasser samt åtkomstmodifikatorer.
6. Läs på om följande refactoring-mönster och applicera dem sedan på programmet (om nödvändigt):
  - “Move method”<sup>1</sup>
  - “Move field”<sup>2</sup>
  - “Replace Array with Object”<sup>3</sup>
  - “Replace Iteration with Recursion”
  - “Split Loop”
  - “Encapsulate Field”<sup>4</sup>
  - “Reverse Conditional”

Notera på ett separat papper vilka mönster ni använt och var så att ni enklare kan redovisa detta för handledaren senare.

---

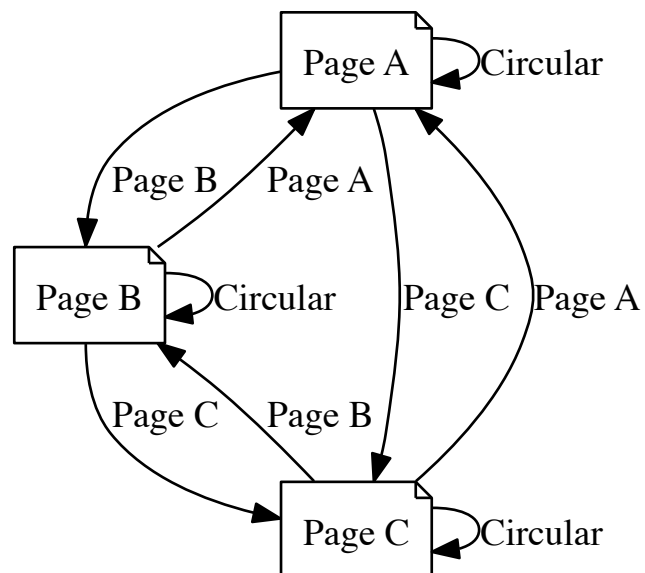
<sup>1</sup>newPage bör inte ligga i Webcrawler utan i Page – varför?

<sup>2</sup>Konsekvenser av move method!

<sup>3</sup>Ledning: links borde vara ett LinkSet analogt med PageSet

<sup>4</sup>T.ex. instansvariabeln crawlDepth

7. Kör programmet lämpligen så här: `java Webcrawler | dot -Tpdf -o test.pdf` (ändras om ni t.ex. lägger Webcrawler i ett paket). Då bör följande figur skapas i filen `test.pdf`:



där varje box avser en besökt webbsida, varje pil en länk från en webbsida till en annan, texten i boxarna deras `title`-attribut, och texten på pilarna länktextern.