

Miniprojekt i programmering (MIP)

dat1/inf1 3. - 5. august 2004

Obligatorisk del

MyHardware A/S er et firma, der sælger enkelt komponenter (bundkort, CPU, harddiske, RAM-blokke) og deres egen fabrikerede komplette pc'er. Lav et program der holder styr på både de enkelt komponenter og komplette pc'er MyHardware A/S sælger. Der er givet et eksempel for hver af de enkelt komponenter herunder. Selve data i eksemplerne er ikke væsentlige disse skal blot vise, hvad der gemmes for hvor komponent.

For en CPU skal følgende information registreres.

- *Fabrikant:* "Intel", *Model:* "Celeron", *Frekvens:* "1700 MHz", *Pris:* "419,00".

For et bundkort skal følgende information registreres.

- *Fabrikant:* "Asus", *Model:* "ASUS A7N266-VM", *Information:* "NVidia nForce 220-D", *Pris:* "478,00".

For en harddisk skal følgende information registreres.

- *Fabrikant:* "Hitachi", *Model:* "Hitachi DeskStar 180GXP", *Størrelse:* "40GB", *Pris:* "435,00".

For RAM-blokke skal følgende information registreres.

- *Fabrikant:* "Kingston", *Model:* "Kingston 1 GB", *Størrelse:* "1GB", *Pris:* "16.072,00".

En komplette pc består af et bundkort, en eller flere CPU'er, en eller flere harddisk og en eller flere RAM-blokke. Prisen for en pc er summen af dens enkelt komponenter. Udover de fire typer af enkelt komponenter og prisen skal følgende information gemmes for en pc.

- *Fabrikant:* "MyHardware A/S", *Model:* "Thinking PC S6".

Bemærkninger til obligatorisk del

- Det antages, at en komponent/pc kan identificeres vha. kombinationen af fabrikant og model navnene f.eks. er der kun en harddisk fra fabrikanten "Hitachi" med model navnet "Hitachi DeskStar 180GXP" og kun en RAM blok fra Kingston med model navnet "Kingston 1 GB".
- Det udviklede programmet skal blot kunne *vise* hvad MyHardware A/S sælger.

Programstruktur

Programmet bør som et minimum indeholde følgende klasser.

- **Component:** En eller flere klasser der repræsenterer de enkelt komponenter og komplette pc'er. Klassen (eller klasserne) skal som et minimum indehold følgende.
 - En metode der returnere en tekstuel repræsentation af komponenten/pc'en
 - En metode der kan sammenligne om to komponenter/pc'er er ens
- **Warehouse:** En klasse der repræsenterer de enkelt komponenter og pc'er MyHardware A/S sælger. Klassen skal som et minimum indeholde følgende.
 - En passende data struktur til at håndtere alle komponenter og pc'er
 - En metode til at tilføje og fjerne komponenter og pc'er
 - En metode til at lede efter komponenter og pc'er vha. fabrikant og model navn

- **UserInterface:** En klasse der repræsenterer et simpelt tekstbaseret menu system. Der skal være mulighed for at foretage følgende operationer. Bemærk brugergrænsefladen skal være funktionel den behøver ikke at være flot.
 - Tilføje og fjerne en komponent og en pc
 - Liste en enkelt eller alle komponenter og pc'er
 - Søge efter en komponent/pc vha. fabrikant og model navn og vis de komponenter der findes
- **RunMe:** En klasse der indeholder en main metode.

Bruges der klasser som ikke er en del af Java Standard Edition version 1.4 skal dette fremgå klart af jeres besvarelse.

For klassen UserInterface kan klassen `Console.java` fra bogen "Core Java" til håndtering af input fra tastaturet med fordel bruges. Klassen kan hentes på følgende URL:

<http://www.cs.auc.dk/~torp/Teaching/E03/OOP/Console.java>

Frivillig del

Her kan programmet udvides på alle de måder i finder relevante. Som eksempler kan nævnes.

- Gem og hent data til og fra en fil f.eks. vha. `Serializable` interfacet.
- Lav unit tests for alle klasser vha. JUnit værktøjet.
- Lav precondition og postcondition checks vha. Java 1.4's `assert` mekanisme.
- Lav en gennemført fejl håndtering (exception håndtering) i hele programmet således at der tages højde for de fejl en bruger af jeres program måtte lave.
- Lav søgning på flere felter så som fabrikant og pris eventuelt ved brug af regulære udtryk.

Krav til dokumentation

Der skal være en forside med kort beskrivelse af designet af program og status af jeres arbejde. Anvendte kilder udover lærebogen og kursushjemmeside **skal** klart angives.

Krav til programudskrift

For udskriften af programmet, der skal afleveret, er det væsentligt at den er letlæselig. Det betyder følgende.

- At variable, metoder, klasser og interfaces har gode og sigende navne.
- At kildekoden er pænt formateret.
- At ingen programlinierne er afkortet eller ombrudt i udskriften, sæt f.eks. maksimum linjebredde til 80 karakterer.
- At der er JavaDoc dokumentation på alle constructors, klasser og interfaces. Dette dokumentationskravet gælder også for alle offentlige variable og metoder. Alle øvrige variable/metoder må også gerne dokumenteres.

Disse krav kan alle overholdes ved at følge Sun's kode konventioner, der kan findes på følgende URL <http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>.