

Kontrollskrivning 2, Programmeringsteknik för D/C

2005-11-18, 15.00-18.00

Tillåtet hjälpmedel: Java-snabbreferens. *Preliminärt* ger uppgift 1 12 poäng, uppgift 2 14 poäng, uppgift 3 14 poäng.

Jag meddelar på kurshemsidan när rättningen är klar. Du får ett papper med ditt resultat i ditt fack, och senare blir det visning av skrivningen.

1. Många tillämpningsprogram, till exempel webbläsare, sparar filer som de skapar med förhoppningen att samma fil skall behövas igen och då inte behöva skapas på nytt. Detta kallas att filerna *cachas*. En fil-cache får inte bli hur stor som helst utan måste städas regelbundet, till exempel genom att de äldsta filerna tas bort.

Skriv en metod med följande rubrik:

```
/** tag bort den äldsta "vanliga filen" (inte katalogen) ur katalogen med
    namnet directoryName. Ger false om directoryName inte existerar eller
    inte är en katalog, true annars */
public boolean cleanCache(String directoryName);
```

Använd standardklassen `java.io.File` (du får förutsätta att klassen har importerats). Klassen beskriver alla slags filer, både kataloger och vanliga filer. Den har följande (förkortade och något förenklade) specifikation (metoden `getName` behöver du inte utnyttja för rän i uppgift 2):

```
specification File {
    /** skapa ett objekt som beskriver filen med namnet name */
    File(String name);

    /** undersök om filen existerar */
    boolean exists();

    /** undersök om filen är en "vanlig fil" (true) eller en katalog (false) */
    boolean isFile();

    /** tag bort denna fil */
    void delete();

    /** tag reda på filnamnet */
    String getName();

    /** tag reda på tiden då filen skapades eller modifierades (millisekunder
        efter klockan 00:00 1 januari 1970) */
    long lastModified();

    /** tag reda på filerna i denna katalog */
    File[] listFiles();
}
```

2. Även denna uppgift handlar om kataloger och filer, och du skall också här utnyttja klassen `File` som beskrivs i uppgift 1.

Skriv en metod som räknar hur många filer med olika tillägg som det finns i en given katalog och producerar en utskrift enligt följande exempel (raden "other" är antalet filer som inte har något av tilläggen i tabellen):

```
.java 14
.class 10
.jpg 5
.tex 1
other 4
```

Metoden skall ha följande rubrik:

```
/** räkna filer med olika tillägg i katalogen med namnet directoryName,
    skriv ut resultatlistan. Tilläggen finns i vektorn extensions. Du
    får förutsätta att directoryName beskriver en katalog som existerar */
public void countFiles(String directoryName, String[] extensions);
```

Exempel på anrop av metoden:

```
String[] extensions = { ".java", ".class", ".jpg", ".tex" };
countFiles("pfd/inluppg2", extensions);
```

För att undersöka om en fil har ett visst tillägg bör du använda metoden `endsWith` i klassen `String`. Metoden undersöker om en sträng avslutas med en annan sträng — om strängen `s` har värdet "Test.java" så är `s.endsWith(".java")` `true`, `s.endsWith(".class")` `false`.

3. En klass som beskriver en följd av tal sorterade i växande ordning har följande specifikation:

```
specification SortedSequence {
    /** skapa en tom talföljd */
    SortedSequence();

    /** lägg in talet nbr på rätt plats i följderna */
    void insert(int nbr);

    /** tag bort dubletter ur följderna, lägg de borttagna elementen i
        en ny lista som returneras */
    ArrayList<Integer> moveDuplicates();

    /** skriv ut talen i följderna på en rad, med blanktecken mellan talen */
    void print();
}
```

Implementera klassen fullständigt, *förutom* metoden `insert`. Talen *skall* lagras i `Integer`-objekt som lagras i en `ArrayList`.

Exempel på olika fall som kan inträffa i metoden `moveDuplicates`:

```
1 5 5 19 55 => 5 tas bort, returnerar en lista med talet 5
4 8 9 13 24 => inget tas bort, returnerar en tom lista
4 6 6 6 7 9 9 => 6 6 9 tas bort, returnerar en lista med talen 6 6 9
```