package and import

Paket och sökvägar i Java

Vad betyder import java.util.Arrays?

Vi har många gånger redan skrivit `import` och så något “paket” för att få tillgång till olika klasser med mera. Men vad betyder det?

Paket består av namn avskilda med “.” t ex

```
import java.util.Arrays;
```

Paketets delar är kataloger

`import java.util.Arrays;` talar om för java och javac att vi använder namn som går att hitta i sökvägen som motsvarar paketets fulla namn. I detta fall `Arrays` och det är namnet på en klass.

Klassen ligger paketerad i en katalog:

`java/util/` - jämför `java.util.`

Men var ligger klassen egentligen?

Alla paket i Javas API är i sin tur paketerade i en så kallad jar-fil. Den ligger i ert system och heter `rt.jar` I min dator ligger den mer exakt i:

```
/usr/lib/jvm/java-7-openjdk-amd64/jre/lib/rt.jar
```

Vi kan verifiera att klasserna ligger i den.

Vi inspekterar rt.jar

```
jar tf /usr/lib/jvm/java-7-openjdk-amd64/jre/lib/rt.jar|
grep Arrays.class
java/util/Arrays.class
```

Kommandot `jar` fungerar väldigt likt kommandot `tar` för dem som är bekanta med det. Man använder alltså `jar` för att skapa en fil med en massa kataloger och filer inuti (lite som en ZIP-fil, alltså).

Leta upp filen rt.jar i ert system!

Leta upp filen rt.jar i er Java-installation och kör kommandot:

```
jar tf sökväg-till-filen
```

Alla Java-klasser med sökväg listas!

Många klasser blir det...

src.zip - källkoden till klasserna

I en del Java-installationer så medföljer även källkoden till alla klasser i API:et. I så fall ligger de i filen `src.zip` som på mitt system ligger i:

`/usr/lib/jvm/openjdk-7/src.zip`

Prova gärna att packa upp den och titta på några klasser, t ex `java/lang/String.java`

`Strings` klassdefinition

Klassen String är deklarerad så här:

(håll i er)

```
package java.lang;

...(massa import-satser)

public final class java.lang.String implements java.io.
Serializable, java.lang.Comparable<java.lang.String>,
java.lang.CharSequence {

 ...(Konstruktorer, metoder och variabler)...

}
```

Att vara medlem i ett paket

Första satsen i String.java är:

```
package java.lang;
```

Filen är lagrad i katalogen java/lang/ och raden betyder: “Jag är med i paketet `java.lang`”

Filen måste ligga i en katalog `/lang/` i katalogen `/java/` alltså:
`/java/lang/String.java`

Vi skapar ett eget paket

jag skapar en katalogstruktur enligt följande:

```
.
└── tig058
    └── examples
```

. betyder “aktuell katalog” - där jag “står”.

I examples skapar jag filen Packat.java

package-dekaration måste vara först

Hela klassen är nu:

```
// Bara kommentarer får komma före package!
package tig058.examples;
// Om package finns med, måste det komma först!

public class Packat{
    public static void main(String[] args){
        System.out.println("Hello package world!");
    }
}
```

Hur kompilerar vi klasser i paket?

För att kompilera tig058.examples.Packat.java så måste vi stå i katalogen där tig058 ligger:

```
$ ls -l
tig058
$ javac tig058/examples/Packat.java

(windows: javac tig058\examples\Packat.java )
```

Hur vi kör Packat (som har en main)

För att köra måste vi ange hela klassnamnet (qualified name) med punktnotation:

```
$ java tig058.examples.Packat
Hello package world!
```

(samma i windows)

Vi lägger till ett sub-paket

Vi skapar en katalog till under tig058:

och skapar en klass StringMethods där:

```
.
└── tig058
    ├── examples
    │   ├── Packat.class
    │   └── Packat.java
    └── util
        ├── StringMethods.class
        └── StringMethods.java
```

tig058.util.StringMethods

Paketdeklarationen blir då:

```
package tig058.util;
```

I StringMethods lägger vi lite util-metoder för strängar, t ex ucFirst(String) - en metod för att göra om första bokstaven i en sträng till stor bokstav, bra att ha för namn t ex.

tig058.util.StringMethods

```
package tig058.util;
public class StringMethods{
  public static String ucFirst(String s){
    char first = s.charAt(0);
    char upper = Character.toUpperCase(first);
    if(Character.isAlphabetic(first)){
      return s.replaceFirst(""+first, ""+upper);
    }else{
      return s;
    }
  }
}
```

För att använda StringMethods

Vill vi använda StringMethods så måste vi importera antingen

```
tig058.util.*; //Alla klasser i paketet
tig058.util.StringMethods; //StringMethods
```

Användande av StringMethods

```
import tig058.util.*;
public class Test{
  public static void main(String[] args){
    System.out.println(StringMethods.ucFirst("rikard"));
    System.out.println(StringMethods.ucFirst("88rikard"));
  }
}
```

import static

Vi som är lata gillar att slippa skriva så mycket.

Om jag bara vill importera ucFirst(String), hur gör jag då? Metoden är static och det finns en särskild syntax för detta:

```
import static tig058.util.StringMethods.ucFirst;
//... klassdekl. och t ex mainmetod
politeName = UcFirst(rawName);
```

Exempel med import static

```
import static tig058.util.StringMethods.ucFirst;

public class TestImportStatic{
  public static void main(String[] args){
    System.out.println(ucFirst("rikard"));
    System.out.println(ucFirst("88rikard"));
  }
}
// Denna fil, TestImportStatic.java ligger
// i katalogen där tig058 ligger.
```

Man kan importera från valfritt paket

Katalogstruktur:

```
.
├── TestImportStatic.java
├── Test.java
└── tig058
    ├── examples
    │   └── Packat.java
    └── util
        └── StringMethods.java
```

Kombination - paket och import

Vi skulle kunna lägga TestImportStatic i paketet tig058.examples:

```
package tig058.examples;
import static tig058.util.StringMethods.ucFirst;
public class TestImportStatic{
  public static void main(String[] args){
    System.out.println(ucFirst("rikard"));
    System.out.println(ucFirst("88rikard"));
  }
}
//javac tig058/examples/TestImportStatic.java
//java tig058.examples.TestImportStatic
```

Ny katalogstruktur

```
.
└── tig058
    ├── examples
    │   ├── Packat.java
    │   └── TestImportStatic.java
    └── util
        └── StringMethods.java
// Om vi står i ".":
$ javac tig058/examples/TestImportStatic.java
$ java tig058.examples.TestImportStatic
Rikard
88rikard
```

Troubleshooting - kompilering

1. Vi kan inte stå i samma katalog:

```
$ cd tig058/examples/
tig058/examples$ javac TestImportStatic.java
TestImportStatic.java:2: error: package tig058.util does
not exist
import static tig058.util.StringMethods.ucFirst;
                           ^
# vi måste stå i samma katalog som tig058 ligger i!
```

Troubleshooting - körning

2. Vi kan inte köra från samma katalog:

```
$ cd tig058/examples
tig058/examples$ java TestImportStatic
Exception in thread "main" java.lang.NoClassDefFoundError:
TestImportStatic (wrong name:
tig058/examples/TestImportStatic)
# Hint: den letar efter katalogen tig058/examples!
```

Classpath

Lär er först var ni ska “stå” för att kunna kompilera och köra. CLASSPATH-variabeln kan fixa problem när man inte kan stå på “rätt” ställe (av någon anledning):

```
tig058/examples$ CLASSPATH=sökvägtillrättkatlog
tig058/examples$ javac TestImportStatic.java
tig058/examples$ java TestImportStatic
Rikard
88rikard
```