

Java tečaj

9. dio – Datoteke i datotečni sustav.

Zadaci za vježbu

V1.1

Zagreb, 2007.

9. Datoteke i datotečni sustav

U nastavku slijede zadaci vezani uz osmi dio tečaja. Sve zadatke potrebno je smjestiti u paket `hr.fer.zemris.java.tecaj_9.vjezba`.

9.2 Zadatak

Na adresi:

http://java.zemris.fer.hr/javaCourses/tasks/java_tecaj_09_zadaca.zip

nalazi se ZIP datoteka s java datotekama `CommandExecException` i `ICommand`. Skinite taj ZIP i raspakirajte ga u Vaš projekt.

U sklopu ovog zadatka potrebno je napraviti komandno-linijski program koji emulira rad jednostavne konzole (terminala). Pri tome naredbe koje se mogu zadati nisu unaprijed definirane, već se programu mogu dodavati “izvana”. Evo kako je to zamišljeno. Pretpostavimo da ste napravili projekt koji je smješten u direktorij:

```
c:\java_workspace_priprema\Tecajevi\
```

U tom direktoriju imate poddirektorije `src` (koji sadrži izvorne datoteke) i `bin` (koji sadrži class datoteke). Prilikom pokretanja bilo kojeg programa iz ovog projekta (`Tecajevi`), navedeni direktorij automatski postaje “radni” tj. trenutni direktorij pokrenutog programa – to je onaj direktorij u koji će se razriješiti staza “.”.

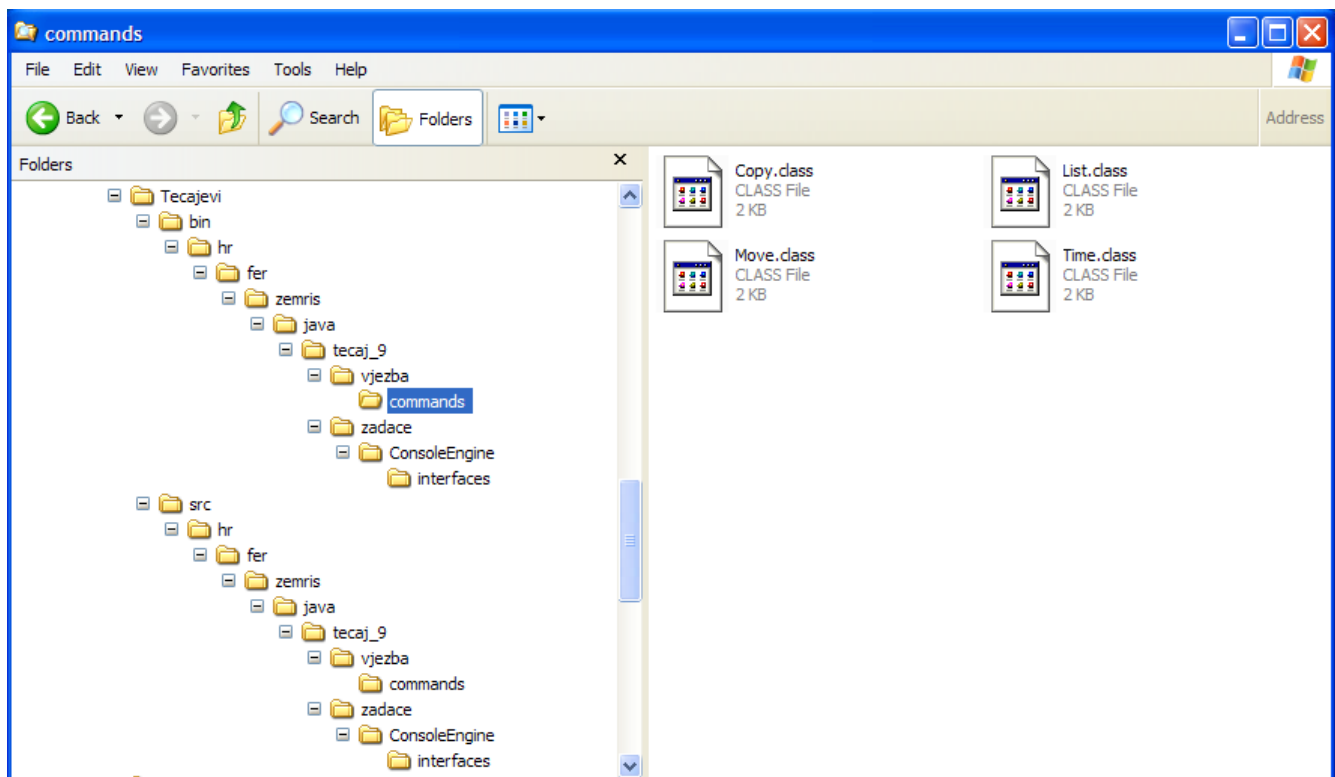
Napišite razred `hr.fer.zemris.java.tecaj_9.vjezba.Console` koji sadrži metodu `main` i predstavlja implementaciju same konzole. Sve naredbe konzole su zasebni razredi koji implementiraju sučelje:

```
hr.fer.zemris.java.tecaj_9.zadace.ConsoleEngine.interfaces.ICommand.
```

Konkretna implementacija naredbi potrebno je smjestiti u paket:

```
hr.fer.zemris.java.tecaj_9.vjezba.commands
```

Ako pretpostavimo da smo implementirali naredbe `ls`, `date`, `cp` i `mv`, na disku ćemo imati sljedeću strukturu datoteka:



Primjerice, naredbu time možemo implementirati na sljedeći način:

```
package hr.fer.zemris.java.tecaj_9.vjezba.commands;

import java.io.PrintStream;
import java.util.Date;

import hr.fer.zemris.java.tecaj_9.zadace.ConsoleEngine.interfaces.CommandExecException;
import hr.fer.zemris.java.tecaj_9.zadace.ConsoleEngine.interfaces.ICommand;

public class Time implements ICommand {

    public void execCommand(String[] parameters, PrintStream out)
        throws CommandExecException, IllegalArgumentException {
        if(parameters!=null && parameters.length>0) {
            throw new IllegalArgumentException("Unexpected arguments!");
        }
        out.println(new Date().toString());
    }

    public String getCommandDescription() {
        return "Returns current date and time. Expects no arguments.";
    }

    public String getCommandName() {
        return "date";
    }
}
```

Prilikom pokretanja konzole program najprije mora istražiti cjelokupni sadržaj direktorija commands (točnije: ./bin/hr/fer/zemris/java/tecaj_9/vjezba/commands). Za svaku class

datoteku koju pronade u tom direktoriju program mora pretpostaviti da se radi o implementaciji neke naredbe. Stoga mora dinamički stvoriti jedan primjerak tog razreda, pitati ga za ime naredbe koju implementira, i zapamtiti i ime i stvoreni primjerak razreda. Primjerice, ako smo otkrili da postoji datoteka `Time.class` u navedenom direktoriju, dinamičko učitavanje možemo postići sljedećim isječkom koda:

```
ICommand cmd = (Icommand)Class
    .forName("hr.fer.zemris.java.tecaj_9.vjezba.commands.Time")
    .newInstance();
```

Pri tome ova naredba izaziva čitav niz različitih iznimki koje treba uhvatiti (a zbog jasnoće koda nisu napisane).

Obratite pri tome pažnju na razliku između lokacije `.class` datoteke (u direktoriju `bin!`), te naziva razreda koji se predaje metodi `Class.forName()`. Ova metoda očekuje puno ime razreda, a to znači čitav paket plus naziv samog razreda (ali bez ekstenzije `class`, ekstenzija ne pripada imenu razreda). Jednom kada to zatražimo, aktivni *ClassLoader* (poseban dio Java okruženja zadužen za učitavanje datoteka potražiti će `.class` datoteku na svim definiranim lokacijama, a to u našem slučaju znači u `bin` direktoriju).

Primjer interakcije s konzolom (kada su dostupne samo naredbe `date` i `ls`):

```
> help
Command: date
-----
Returns current date and time. Expects no arguments.

Command: ls
-----
Lists directory.
If no argument is given, lists current directory.
If one argument is given, lists that directory.

> date
Sun May 20 10:34:27 CEST 2007
> ls
[FIL] .classpath
[FIL] .project
[DIR] .settings
[DIR] bin
[DIR] examples
[DIR] src
> ls bin/hr/fer/zemris/java/tecaj_09/vjezba/commands
Directory bin/hr/fer/zemris/java/tecaj_09/vjezba/commands
does not exists.
```

```
> ls bin/hr/fer/zemris/java/tecaj_9/vjezba/commands
[FIL] List.class
[FIL] Time.class
>
```

U okviru zadaće potrebno je implementirati sljedeće naredbe:

- `date` – ispis trenutnog datuma i vremena
- `ls` – ispis trenutnog direktorija (za format pogledajte prethodni primjer)
- `ls direktorij` – ispis zadanog direktorija (za format pogledajte prethodni primjer)
- `cp izvornaDatoteka odredisnaDatoteka` – kopira datoteku (samo ako odredište ne postoji)
- `mv izvornaDatoteka odredisnaDatoteka` – premješta datoteku
- `rm datotekaIliDirektorij` – briše datoteku ili direktorij (zajedno sa svim poddirektorijima)
- `conv izvKodStr odredKodStr izvDat odredDat` – konvertira kodnu stranicu datoteke

Konzola također treba imati sljedeće ugrađene naredbe (ne moraju biti izvedene kao `ICommand`, ali mogu):

- `help` – ispis svih naredbi i pomoći za njih (vidi prethodni primjer)
- `help naredba` – ispis pomoći samo za zadanu naredbu
- `exit` – izlazak iz konzole