

Si implementi ciascuno dei casi seguenti, integrandolo con un main program che ne verifichi e dimostri le funzionalità.

1) Il gestore di una ferrovia associa a tutti i suoi **treni** un codice univoco, la data del viaggio, una stazione di partenza e una di arrivo, ed il suo prezzo del viaggio. Realizzare una **classe Treno** con le opportune variabili d'istanza e metodi get. (se c'è una variabile int x, il corrispondente metodo sarà int getX() ).

2) Aggiungere alla classe Treno un metodo stampaBiglietto che specifica data, codice del treno, stazione di partenza e di arrivo.

3) Il gestore vuole fare una distinzione tra i **treni ad alta velocità** e quelli **regionali** . I treni ad alta velocità hanno un sovrapprezzo del biglietto del 30%. Ai quelli regionali viene associata una seconda data: la prima assume il significato di inizio validità del biglietto, la seconda quello di fine validità. Realizzare le **sottoclassi TrenoAV e TrenoReg** estendendo opportunamente la classe Treno, ed adeguando opportunamente il metodo stampaBiglietto.

4) Modificare le due sottoclassi menzionate aggiungendo il metodo **controllaValidità** che verifica, alla data odierna, se il biglietto sia valido, ed in caso contrario eroghi una multa pari al 50% del prezzo del biglietto.

---

Nota: per gestire le date si utilizzi la classe Data reperibile qui:  
[http://www.di.unipi.it/~lametti/Teaching\\_files/Data.java](http://www.di.unipi.it/~lametti/Teaching_files/Data.java)

Uso:

- Per avere la data corrente:

```
Data d=new Data();
```

- Per avere la data corrispondente al giorno g/m/ a dove g, m, a sono interi

```
Data d=new Data(g,m,a);
```

- Per confrontare due date:

```
Date d1= new Date(g1,m1,a1);  
Date d2= new Date(g2,m2,a2);  
int delta=d1.getDifference(d2)
```

Se d1 è anteriore, uguale o posteriore a d2 si avrà rispettivamente delta negativo, zero, positivo.

*NOTA: la classe data deriva da un esercizio proposto di Silvia Lametti, Università di Pisa.*