

- 1) Nella gerarchia descritta nell'esercitazione 6 e relativa a **Shape**, **Circle**, **Rectangle**, **Square**, introdurre una interfaccia **Scalable** che dichiara un metodo *scale(double factor)*. Il metodo consente di modificare le dimensioni dei lati/raggio di una figura in base al fattore di proporzionalità *factor*.  
Introdurre anche l'interfaccia **Drawable** che dichiara un metodo *draw()* per visualizzare i dati dell'oggetto. Rendere **Shape** classe astratta e verificare che per questa classe non si possono istanziare oggetti.
- 2) Si scriva un'applicazione che consenta di simulare il lancio di un **Dado** e di una **Moneta**. Si utilizzi un'opportuna classe astratta per rappresentare le caratteristiche comuni delle due classi.  
Simulare, attraverso un opportuno metodo *lancia()*, la visualizzazione del dado che rotola o della moneta che gira durante il lancio. In particolare, il metodo *lancia()* di **Dado** visualizza i numeri da 1 a 6 per un numero intero e casuale di volte (max 10) fino al valore che è uscito (per es. supponendo che dal lancio del dado si ottenga 3, una possibile visualizzazione potrebbe essere: "1 2 3 4 5 6 1 2 3 4 5 6 1 2 3"). Per il metodo *lancia()* di **Moneta** si considerino i simboli T e C per indicare Testa e Croce che si alternano un numero casuale di volte fino al valore che è uscito (per es. se è uscito Testa una visualizzazione potrebbe essere "T C T C T C T C T").
- 3) In un'applicazione per la gestione di una videoteca, i clienti sono memorizzati in oggetti della classe **Persona**. Ogni **Persona** è caratterizzata da una *dataDiNascita*, nome, cognome, *codiceFiscale*, indirizzo, città, cap. Creare una classe **Abbonato** che estenda la classe **Persona** memorizzando in un'opportuna variabile d'istanza sconto la percentuale di sconto a cui l'abbonato ha diritto su ogni acquisto effettuato. Prevedere opportuni metodi per l'incapsulamento di questa variabile.  
Creare inoltre una classe **AbbonatoPremium** che, oltre ad aver diritto ad uno sconto, ha diritto ad un acquisto gratuito ogni volta che accumuli una spesa complessiva superiore a 100€. Scrivere una classe per testare le classi **Abbonato** e **AbbonatoPremium** che memorizzi una lista di oggetti e abbia funzionalità per aggiungere nuovi abbonati, gestire il costo degli acquisti in base al tipo di abbonato e stampare lo stato degli abbonati.
- 4) Creare una gerarchia di classi che possa rappresentare le seguenti entità: **Persona**, **Professore**, **Studente**, **StudenteTriennale** e **StudenteMagistrale**.  
Ogni **Professore** ha una *dataAssunzione*, un ruolo (Ricercatore, Professore Associato o Professore Ordinario), un dipartimento di appartenenza (es. DIID, DICAM, ...). Ogni Professore percepisce uno *stipendio* (prevedere quindi i metodi *getStipendio()* e *setStipendio()*).  
Ogni **Studente** ha una *dataIscrizione*, una *matricola*, un *corsoDiLaurea* a cui è iscritto. Ogni **Studente** paga un contributo d'iscrizione al corso.  
Uno **StudenteTriennale** deve conseguire 180 CFU e proviene da una *scuolaSuperiore* (una stringa per memorizzare la scuola di provenienza). Uno **StudenteMagistrale** deve conseguire 120 CFU e proviene da un *corsoTriennale* (una stringa per memorizzare il corso di laurea triennale di provenienza).  
Prevedere opportuni metodi per l'incapsulamento dei dati.  
Laddove possibile, riutilizzare classi sviluppate in esercitazioni precedenti (per es. **Persona**). In ogni classe prevedere il metodo *toString()* che restituisce una stringa descrittiva dell'oggetto (es. per **Persona**: "<nome> <cognome> <dataDiNascita> <indirizzo>", per **Studente**: "<matricola> <nome> <cognome> <dataDiNascita> <indirizzo> <corsoDiLaurea>...", etc).
- 5) In occasione di una gara canora si vuole gestire il televoto. Il pubblico da casa può votare per uno dei 15 partecipanti ma può votare al massimo 5 volte. Il sistema deve poter raccogliere i voti in forma anonima e alla fine delle operazioni stampare la classifica risultante dal televoto.  
Scrivete quindi un programma che permette di inserire un nuovo voto, verifica se da quel numero telefonico sono stati effettuati meno di 5 voti e, in caso affermativo, aggiorna la classifica. Si osservi che non si possono memorizzare i singoli voti (il voto deve restare anonimo) ma bisogna memorizzare l'elenco dei votanti e del numero di voti effettuati da ciascuno.  
Scrivete un programma che all'interno di un menu testuale, permette di 1) simulare l'arrivo di un nuovo voto tramite inserimento di numero di telefono del votante e numero del cantante votato, 2) stampare il totale dei voti ricevuti fino a quel momento e 3) stampare il numero di voti ricevuti da ciascun cantante.

NOTE PER COMPILAZIONE E TEST A RIGA DI COMANDO IN AMBIENTE LINUX:

```
javac -d ../classes -cp ../classes nomeClasse.java compila e genera il bytecode
java -cp ../classes nomePackage.nomeClasse esegue il bytecode sulla JVM
```