Esercitazione n° 4

Obiettivi

- Progettare e implementare per intero un componente software in Java
- Linguaggio Java:
  - Classi astratte
  - Utilizzo di costruttori e metodi di superclasse
  - Polimorfismo
- Esempio guida: simulazione di ingressi in un centro sportivo

Specifica del Problema (1)

Simulazione di ingressi in un Centro Sportivo

Si vuole progettare e implementare un componente software che simuli la gestione degli ingressi in un Centro Sportivo. L'ingresso al centro da parte degli utenti del centro è consentito solo ai possessori di un abbonamento a qualche attività sportiva del centro. All'ingresso gli utenti presentano la propria tessera e possono entrare solo se l'abbonamento è valido.

Dettaglio…

Gli abbonamenti sono di due tipi: a ingressi (**AbbonamentoIngressi**) o mensili (**AbbonamentoMensile**). Tutti gli abbonamenti sono però associati ad un numero di tessera che identifica l'utente.

Tutti gli abbonamenti hanno un metodo ***entra()*** e ***visualizza()*** usati rispettivamente per simulare l'ingresso e visualizzare le caratteristiche di un abbonamento. Questi metodi sono differenziati a seconda del tipo di abbonamento.

Specifica del Problema (2)

Dettaglio…

Il componente software che gestisce la simulazione chiamato **SimulaIngressi** dovrà :

1 Inizializzare lo stato iniziale del Centro Sportivo come necessario per la simulazione. Tale stato deve essere modellato attraverso una entità software chiamata **StatoCentro** che contiene informazioni circa lo 'stato corrente' del Centro Sportivo (es num. persone in piscina, mese corrente, ecc…)

2 Creare alcune istanze di abbonamento

3 Simulare l'ingresso degli abbonamenti creati al Centro. In particolare verrà testato se l'abbonamento autorizza all'ingresso e verrà visualizzato in ogni caso un messaggio su console

Specifica del Problema (3)

Dettaglio…

Per la simulazione si vogliono utilizzare gli abbonamenti :

PesiIngressi *(abbonamento alla palestra a ingressi)*

NuotoLibero *(abbonamento alla piscina a ingressi)*

CorsoNuoto *(corso mensile di nuoto in piscina)*

Esercitazione n°4

Come procedere?

1. Creare in `C:\TEMP` una directory **Esercitazione4** che conterrà tutti i sorgenti e i .class di questa esercitazione.
2. Progettare e implementare la classe **Abbonamento.java** seguendo la specifica della slide n°8, quindi compilarla.
3. Progettare e implementare la classe **AbbonamentoIngressi.java** seguendo la specifica delle slide N°9-10, quindi compilarla.
4. Scaricare il file **Pesi.java**, compilare e capire il codice anche osservando la specifica data nelle slide N°11-12.

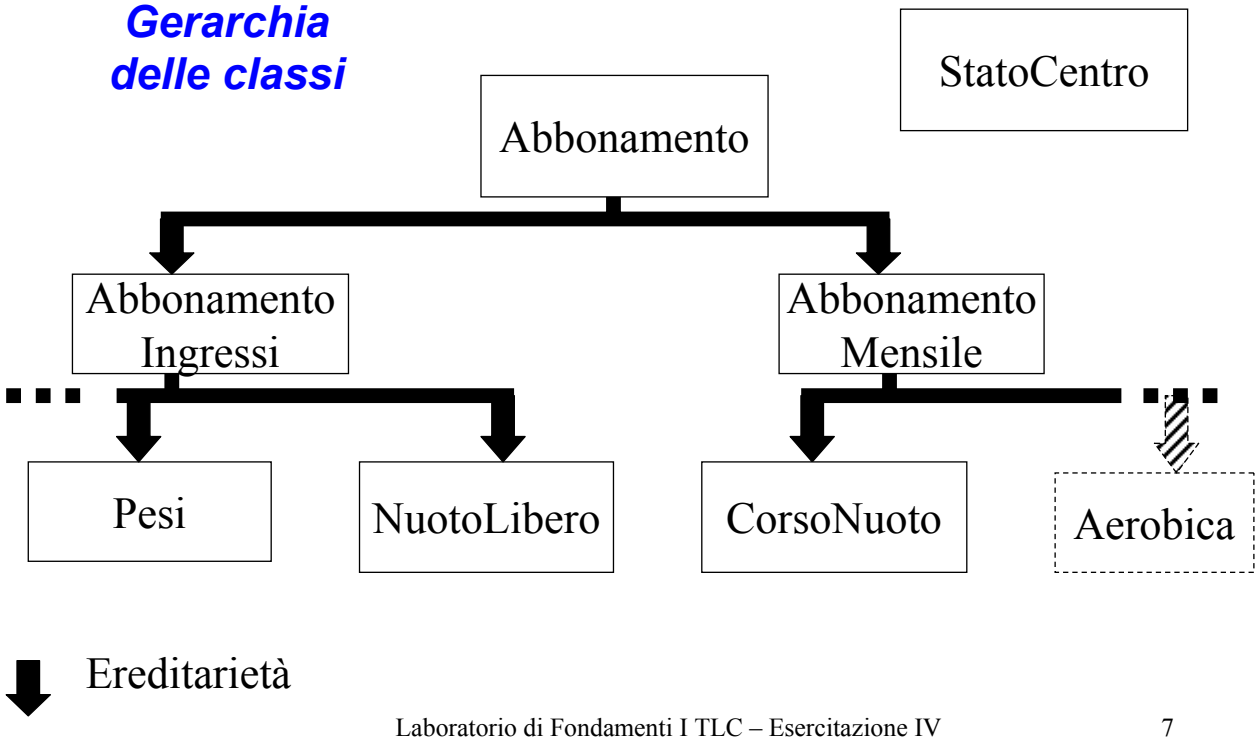
Esercitazione n°4

Come procedere?

5. Progettare e implementare la classe **StatoCentro.java** seguendo la specifica delle slide N°13-14, quindi compilarla.
6. Progettare e implementare la classe **NuotoLibero.java** seguendo la specifica delle slide N°15-16, quindi compilarla.
7. Scrivere il componente software incaricato di gestire la simulazione **SimulaIngressi.java** seguendo la specifica delle slide N°17-18, quindi compilare ed eseguire.

Il Package Esercitazione4

Gerarchia delle classi



La classe Abbonamento

Abbonamento.java **package Esercitazione4**

- Ogni abbonamento ha un numero di tessera associato **numTessera**. Tale numero deve essere intero e *private*.
- Il numero della tessera relativo ad ogni abbonamento è stabilito al momento della creazione dell'abbonamento e deve essere sempre specificato.
- Il numero di tessera associato è restituito dal metodo
  public int **visualizzaTessera()**
- Ogni abbonamento deve possedere i metodi:
  public boolean **entra()** *(per simulare un ingresso)*
  public void **visualizza()** *(per visualizzare a console le caratteristiche di un abbonamento)*

  Tali metodi hanno comportamento differenziato a seconda del tipo di abbonamento

La classe AbbonamentoIngressi (1)

AbbonamentoIngressi.java **package Esercitazione4**

- **AbbonamentoIngressi** estende la classe **Abbonamento**
- Posside due variabili intere e *private*:

  totIngressi *(totale degli ingressi acquistati)*

  effIngressi *(ingressi già effettuati)*
- Costruttori:

  public **AbbonamentoIngressi (** int tessera, int ingressi **)**

  public **AbbonamentoIngressi (** int tessera**)**

  Dove tessera è il numero di tessera dell'abbonamento, mentre ingressi il numero di ingressi acquistati. Se non viene indicato, il numero di ingressi di default è 12.

La classe AbbonamentoIngressi (2)

AbbonamentoIngressi.java **package Esercitazione4**

- public int **totIngressi()** *(restituisce il valore di* `totIngressi`*)*

  public int **effIngressi()** *(restituisce il valore di* `effIngressi`*)*
- protected int **incIngressi()** *(incrementa il valore di effIngressi di 1 e ritorna effIngressi)*
- Il metodo **visualizza()** produce una uscita di questo tipo:

```
Tessera: 33
Ingressi effettuati: 1
Ingressi rimanenti : 23
```

- Il comportamento del metodo **entra()** dipende dal tipo di abbonamento ad ingressi

La classe Pesi (1)

Pesi.java **package Esercitazione4**

Pesi è una classe concreta che modella un abbonamento alla palestra che è ad ingressi. Estende **AbbonamentoIngressi**.

- Costruttori

  public **Pesi (** int tessera, int ingressi **)**
  public **Pesi (** int tessera**)**

  Dove tessera è il numero di tessera dell'abbonamento, mentre ingressi il numero di ingressi acquistati. Se non viene indicato, il numero di ingressi di default è 24.

- Il metodo **entra()**, se si hanno ancora ingressi da effettuare ritorna `true` e aggiorna il numero di ingressi effettuati. Altrimenti ritorna `false` e scrive su console: "Non posso entrare!"

La classe Pesi (2)

Pesi.java **package Esercitazione4**

- Il metodo **visualizza()**, stampa in uscita un messaggio di questo tipo:

```
** ABB. PESI AD INGRESSI      **
   Tessera: 100
   Ingressi effettuati: 1
   Ingressi rimanenti : 23
**********************************
```

La classe StatoCentro (1)

StatoCentro.java **package Esercitazione4**

StatoCentro raccoglie staticamente alcune informazioni utili inerenti lo stato del Centro Sportivo

- **numInPiscina** è una variabile intera e *private* che contiene il numero delle persone che attualmente stanno facendo nuoto libero (e non il corso di nuoto)
- I metodi statici e pubblici void **incNum()** e void **decNum()** rispettivamente incrementano e decrementano di 1 il valore di **numInPiscina**
- **capienza** è una costante intera che ha valore 20 e rappresenta il numero massimo di persone che fanno nuoto libero che la piscina può ospitare.

La classe StatoCentro (2)

StatoCentro.java **package Esercitazione4**

- Il metodo *public static boolean* ***PostiDisponibili()*** ritorna `true` se il **numInPiscina** non supera e non satura la **capienza**, altrimenti ritorna `false`.
- **meseCorrente** è una Stringa *private* che contiene il mese corrente. Es: “Febbraio”.
- public static void **setMese** (String mese) imposta **meseCorrente** alla stringa ricevuta come argomento
- public static String **currentMese** () ritorna **meseCorrente**

La classe NuotoLibero (1)

NuotoLibero.java **package Esercitazione4**

NuotoLibero è una classe concreta che modella un abbonamento ad ingressi al nuoto libero in piscina. Estende **AbbonamentoIngressi**.

- Costruttori

  public **NuotoLibero (** int tessera, int ingressi **)**

  public **NuotoLibero (** int tessera**)**

  Dove tessera è il numero di tessera dell'abbonamento, mentre ingressi il numero di ingressi acquistati. Se non viene indicato, il numero di ingressi di default è 12.

La classe NuotoLibero (2)

NuotoLibero.java **package Esercitazione4**

- Il metodo **entra()**, controlla se in vasca ci sono posti disponibili. In caso negativo ritorna `false` e stampa su console: "la piscina è piena!" e "Non posso entrare!". Se invece ci sono ancora posti, controlla se si hanno ancora ingressi da effettuare. In caso affermativo ritorna `true` e aggiorna il numero di ingressi effettuati oltre che lo Stato del Centro. Altrimenti ritorna `false` e scrive su console: "Non posso entrare!"
- Il metodo **visualizza()**, stampa in uscita un messaggio di questo tipo:

```
** ABB. NUOTO AD INGRESSI     **
   Tessera: 30
   Ingressi effettuati: 1
   Ingressi rimanenti : 11
**********************************
```

La classe SimulaIngressi (1)

SimulaIngressi.java package Esercitazione4

SimulaIngressi è il componente software che gestisce la simulazione. La simulazione avviene in tre fasi qui di seguito riportate che devono essere implementate.

- **Inizializzazione stato del Centro**

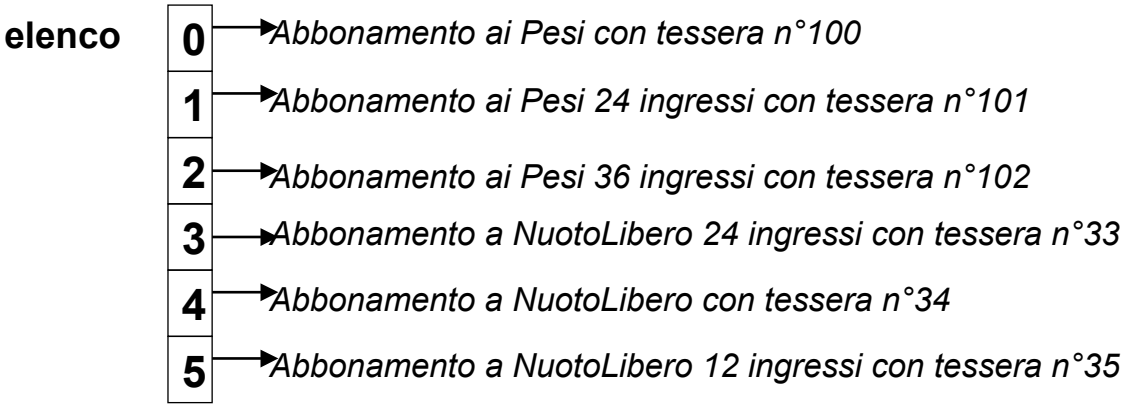
  Inserire 18 persone nella vasca del nuoto libero (**numINPiscina** di **StatoCentro** deve valere 18)

- **Creazione Abbonamenti**

  Creare un array di nome **elenco** di 6 elementi di tipo **Abbonamento**, come indicato nella prossima slide

La classe SimulaIngressi (2)

SimulaIngressi.java package Esercitazione4



- **Simulazione**

  Visualizzare lo stato dell'abbonamento con tessera n° 102, quindi simulare un suo ingresso al centro. Simulare in seguito l'ingresso anche di tutti gli altri abbonamenti creati (tranne quello con tessera n°102 già simulato).

Esercitazione n°4: facoltativo

8 Scaricare dal sito web i file **AbbonamentoMensile.java** e **CorsoNuoto.java**, capirne il funzionamento e compilarli.

9 Modificare il *main()* in **SimulaIngressi.java** in modo tale che all'inizio **meseCorrente** in **StatoCentro** venga inizializzato a "Febbraio".

Poi si creino due abbonamenti, uno con numero di tessera 40, e valido in Febbraio, l'altro con numero di tessera 41, valido in Febbraio e con Maestro "Carlo".

Si simuli quindi l'ingresso di entrambi gli abbonamenti al Centro Sportivo.