

# Orario Treni

Sviluppare l'applicazione che consente di gestire l'orario dei treni e di monitorare i passaggi effettivi alle stazioni.

Tutte le classi devono essere nel package "orario".

Progetto di partenza: [progetto](#)

---

## R1 - Definizione Percorsi

Il sistema agisce tramite la classe **Orari**.

La prima fase dell'uso consiste nel definire i percorsi dei treni.

Il metodo **creaPercorso()** che accetta come parametro il codice (es. IC2345) e la categoria e crea un oggetto **Percorso**. La categoria di un treno può essere "Intercity", "Eurostar", "Interregionale", o "Regionale". La classe **Percorso** che offre i metodi **getCodice()**, **getCategoria()** per leggere il codice e la categoria. I treni possono essere ordinari oppure straordinari, per definire il tipo si usa il metodo **setStraordinario()** che riceve un parametro boolean, inoltre è a disposizione il metodo **isStraordinario()** per conoscere il tipo. Un percorso per default non è straordinario.

E' possibile conoscere tutti i percorsi definiti nell'orario tramite il metodo **getPercorsi()** che restituisce una collezione di oggetti **Percorso**.

Inoltre è possibile usare il metodo **getPercorso()** che riceve come parametro il codice di un percorso e restituisce l'oggetto corrispondente.

## R2 - Fermate

Ad un percorso è possibile aggiungere delle fermate presso le stazioni.

La classe **Percorso** fornisce il metodo **aggiungiFermata()** che riceve come parametri il nome della stazione, e l'orario previsto in ore e minuti. Questo metodo restituisce l'oggetto **Fermata** corrispondente.

La classe **Fermata** offre i metodi **getStazione()**, **getOra()**, **getMinuti()**.

Per ottenere l'elenco di tutte le fermate di un percorso si usa il metodo **getFermate()** della classe **Percorso** che restituisce la lista degli oggetti **Fermata** ordinati in base all'orario.

## R3 - Treni

I treni seguono i percorsi stabiliti e fanno dei passaggi presso le stazioni a certe ore.

La classe **Orari** offre il metodo **nuovoTreno()** che riceve come parametri il codice del percorso e la data in cui viaggia il treno e restituisce l'oggetto **Treno** corrispondente.

Se il codice del percorso non corrisponde ad un percorso definito in precedenza viene generata l'eccezione di **PercorsoNonValido**.

La classe **Treno** offre i metodi **getPercorso()**, **getGiorno()**, **getMese()** e **getAnno()** per accedere alle informazioni sul percorso e alla data di viaggio.

Dato un percorso è possibile ottenere l'elenco dei treni che l'hanno percorso tramite il metodo **getTreni()** della classe **Percorso**: questo metodo restituisce la lista dei treni ordinata per data di viaggio decrescente.

## R4 - Passaggi

I treni fanno dei passaggi presso le stazioni del percorso a certe ore.

La classe **Treno** offre il metodo **registraPassaggio()** che riceve come parametri il nome della stazione, l'ora ed i minuti del passaggio; il metodo restituisce l'oggetto **Passaggio** creato.

Se il nome della stazione non corrisponde ad una delle fermate del percorso seguito viene segnalata l'eccezione di **StazioneNonValida**.

La classe **Passaggio** offre i metodo **getStazione()**, **getOra()**, **getMinuti()**.

Inoltre offre il metodo **ritardo()** che restituisce il ritardo rispetto all'orario previsto (definito nella fermata) espresso in minuti.

## R5 - Statistiche

La classe Treno permette di ottenere alcune informazioni statistiche.

Il metodo **arrivato()** restituisce un valore booleano che indica se il treno ha fatto la fermata presso l'ultima stazione.

I metodi **ritardoMassimo()**, **ritardoFinale()** permettono di avere informazioni sui ritardi.

E' possibile calcolare le statistiche anche a livello di percorso.

La classe Percorso offre i metodi **ritardoMedio()**, **ritardoMassimo()** che restituiscono rispettivamente la media ed il massimo tra i ritardi finali dei treni che hanno seguito il percorso.

---