

### MODALITÀ DI LABORATORIO ONLINE

Per le esercitazioni di oggi, troverete il meeting nel canale “Lezioni” del Teams del corso. Il Docente presenterà gli esercizi da svolgere: per ogni esercizio avremo una breve introduzione seguita dallo svolgimento da parte degli studenti. Potete usare la chat per richiedere aiuto da parte del Docente. Usare il microfono solo per comunicare qualche urgenza.

## Esercizio 1: BookStore

Scrivere un programma Java usando JPA che simuli le operazioni principali di un negozio di libri (aggiunta, rimozione e lista dei libri).

1. Da Derby, creare il database con un utente associato. Fate attenzione a specificare user name e password non vuote.
2. Creare una classe Entity Book caratterizzata da:
  - titolo, prezzo, descrizione, isbn e categoria
3. Creare una classe Main in cui l'utente può aggiungere e rimuovere un libro e permetta di fare una ricerca per filtri (titolo, categoria, isbn).

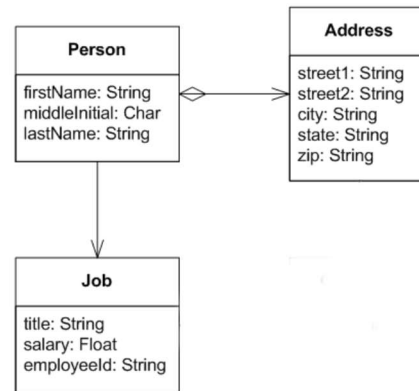
Note:

- Attenzione al file persistence.xml. Si consiglia di creare la persistence unit manualmente, specificando il nome della classe mappata come entity, e indicando login, password, url (come indicato dalla connessione di riferimento nella view Services -> Databases di NetBeans)
- Ricordatevi di aggiungere le seguenti dipendenze (in NetBeans: Projects -> Nome del progetto -> Dependencies) al progetto
  - o `javaee-api` per Java Enterprise
  - o `javax.persistence` per JPA
  - o `eclipselink` per l'implementazione JPA EclipseLink
- Se utilizzate come driver JDBC il Client Driver di Derby, ricordatevi di aggiungere i seguenti jar alle dipendenze del progetto:
  - o `derby.jar`, `derbyclient.jar`, `derbynet.jar`
- Se utilizzate come schema generation della persistence unit “drop-and-create”, non occorre che creiate manualmente le tabelle.

## Esercizio 2: Job Scheduling

Scrivere un programma Java usando JPA che permetta di assegnare un Job ad un Dipendente.

Avanzato: associare più di un Job ad un Dipendente. Verificate quali tabelle vengono create da JPA nel caso le relazioni siano one-to-many, provando con e senza l'annotazione JoinColumn. Fate attenzione alla rimozione automatica “drop-and-create” nei casi in cui le tabelle generate passino da join column a join table (potrebbe essere necessario rimuovere manualmente le tabelle).



## Esercizio 3: Job Scheduling

Una Compagnia ha un insieme di Job da assegnare ai propri Dipendenti.

Una compagnia può avere più job da assegnare ai propri dipendenti; la relazione Dipendente - Job è 1 a 1.

Scrivere un programma usando JPA che permette alla Compagnia di:

1. Assegnare o rimuovere un Job ad un Dipendente
2. Aggiungere un nuovo Dipendente
3. Creare un nuovo Job
4. (Avanzato) Assegnare più di un Job ad un Dipendente

