

**Requisiti.** L'applicazione da progettare riguarda la gestione di trenini elettrici che girano su un sistema di rotaie condiviso. Ogni trenino è caratterizzato da un nome (una stringa) e da una sequenza non vuota ordinata di vagoni ad esso assegnati. Un vagone può ovviamente essere assegnato ad al più un trenino in ogni momento. Ogni vagone è caratterizzato da un codice (una stringa) e da una descrizione (una stringa). Ogni trenino è posto su un binario. Ciascun binario può essere caratterizzato da un codice (un intero) e può essere un binario morto oppure un percorso. Ogni binario morto è legato ad un singolo percorso. Ogni percorso può intersecare altri percorsi (si noti che se il percorso  $i$  interseca il percorso  $j$  allora ovviamente il percorso  $j$  interseca il percorso  $i$ ). Il trenino, che è inizialmente fermo, può essere fatto partire in avanti o indietro, una volta partito può essere fermato. E' di interesse conoscere lo stato del trenino. Inoltre il trenino può essere modificato, aggiungendo o eliminando vagoni solo quando è fermo. L'utente del sistema è interessato ad effettuare diversi controlli in particolare:

- dato un insieme  $I$  di trenini, verificare se almeno due di essi sono posti su percorsi che si intersecano;
- dato un insieme  $J$  di vagoni restituire il sottoinsieme di  $J$  costituito da vagoni non assegnati ad alcun trenino.

---

**Domanda 1.** Basandosi sui requisiti riportati sopra, svolgere la fase di analisi producendo lo schema concettuale in UML per l'applicazione e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

**Domanda 2.** Svolgere la fase di progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio solo progettare gli algoritmi e definire le responsabilità sulle associazioni.

**Domanda 3.** Svolgere la fase di realizzazione, producendo un programma JAVA e motivando, qualora ce ne fosse bisogno, le scelte effettuate.

È obbligatorio realizzare in JAVA solo i seguenti aspetti dello schema concettuale:

- le classe **Trenino** e **Vagone**, tutte le associazioni che le legano;
  - il primo use case.
-