

Analisi e Progettazione del Software

Prova scritta del 26 gennaio 2018

Si vuole progettare un'applicazione per la gestione di una *stazione sciistica*. Di una stazione sciistica interessano il nome e i suoi impianti di risalita. Ciascun impianto ha un nome, un tipo (seggiovia, funivia, bidonvia o sciovia), gli orari di apertura e di chiusura e una capacità in termini di persone all'ora. Per ciascuna persona che si registra alla stazione si memorizzano l'ora di inizio e di fine del suo diritto di accesso agli impianti. Per ogni ingresso di una persona ad un impianto viene registrato l'orario. Non viene invece memorizzata la data, in quanto i dati dell'applicazione si riferiscono ad una sola giornata (e vengono cancellati quotidianamente).

Si considerino le operazioni di aggiunta di un impianto, di registrazione di una persona e di inserimento di un ingresso di una persona ad un impianto. Si considerino inoltre gli usuali selettori, aggiungendone uno che restituisce, per una persona, il numero di risalite effettuate per un dato impianto. Si preveda inoltre l'operazione che restituisce per una persona l'impianto che ha usato più spesso. Si consideri infine un'operazione che, dato un orario, cancella tutte le persone la cui registrazione è scaduta in quell'orario.

Si gestiscano le precondizioni tramite l'eccezione `invalid_argument`, identificando quelle che si ritengono opportune; si aggiunga la precondizione che gli inserimenti degli ingressi agli impianti avvengano in ordine strettamente cronologico.

Si consideri già disponibile la classe `Orario` che gestisce un orario della giornata (ore, minuti e secondi) con tutte le funzioni e gli operatori che si ritengono opportuni (ad es. `+`, `-`, `==`, `<`).

Si consideri già disponibile anche la classe `Persona` che contiene tutti i dati della persona incluso un *codice* (alfanumerico) identificativo, di cui però non è nota la definizione. Si assumano disponibili per la classe `Persona` anche l'operatore di output e il selettore `Codice()` per il codice.

Esercizio 1 (punti 5) Si disegni il diagramma UML delle classi per l'applicazione descritta.

Esercizio 2 (punti 10) Si scrivano le definizioni delle classi C++ per l'applicazione.

Esercizio 3 (punti 10) Si scrivano le definizioni delle funzioni delle classi, incluso l'operatore di output.

Esercizio 4 (punti 5) Si scriva una funzione esterna che prenda come parametro una stazione e il codice di una persona e restituisca il tempo minimo (in secondi) intercorso tra due suoi ingressi a due seggiovie qualsiasi.