

Analisi e Progettazione del Software

Prova scritta del 25 febbraio 2019

Si vuole progettare una classe C++ per la gestione di un centro assistenza per le riparazione di apparecchi elettronici. Le operazioni principali per il centro, di cui si memorizza il nome, sono le seguenti:

RegistraCliente(*c* : *Persona*)

La persona *c* viene registrata tra i clienti del centro. Se la persona è già stata registrata, l'operazione non ha effetti.

CreaRiparazione(*a* : *Apparecchio*, *d* : *Data*)

Viene creata una riparazione per l'apparecchio *a* da effettuare con data di consegna *d* e data di accettazione uguale alla *data corrente*.

Precondizioni: La data di consegna è successiva alla data corrente. L'apparecchio non è già in riparazione nel centro. Il proprietario dell'apparecchio è registrato come cliente (a ciascun apparecchio è associata una persona che ne è il proprietario).

ConsegnaApparecchi(*d* : *Data*)

Tutti gli apparecchi con data di consegna minore o uguale a *d* vengono riconsegnati, cioè eliminati dal centro.

Precondizioni: La data *d* è maggiore o uguale a tutte le data di accettazione degli apparecchi in riparazione nel centro.

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

Esercizio 2 (punti 5) Si scriva la definizione della classe **Centro** e delle eventuali altre classi che compongono il diagramma UML, assumendo però già disponibili e *immodificabili* le classi **Persona** e **Apparecchio** per la gestione di clienti e apparecchi, di cui non si conosce la rappresentazione. Si assuma soltanto che l'apparecchio abbia il metodo pubblico **Proprietario()** che restituisce il puntatore al proprietario.

Esercizio 3 (punti 10) Si scrivano le definizioni dei metodi della classe **Centro** che corrispondono alle operazioni sopra elencate e i selettori che si ritengono opportuni. Si gestiscano le precondizioni tramite il lancio dell'eccezione **invalid_argument**. Si definiscano i metodi (modificatori e selettori) delle eventuali altre classi del diagramma UML (escluse ovviamente **Persona** e **Apparecchio**). Si assuma disponibile anche la classe **Data** con tutti i metodi e gli operatori che si ritengono opportuni, compreso un costruttore senza argomenti che assegna all'oggetto la data corrente.

Esercizio 4 (punti 5) Si scriva un metodo della classe che riceva come parametro il nome di un file e scriva sul file stesso, uno per riga, i codice degli apparecchi in riparazione, con il nome del proprietario e data di consegna. Si assumano disponibili il metodo **Codice()** della classe **Apparecchio**, il metodo **Nome()** della classe **Persona** e l'operatore di output della classe **Data**.

Esercizio 5 (punti 5) Si scriva una funzione esterna (non *friend*) alla classe **Centro** che riceva come parametro un oggetto della classe **Centro** ed uno della classe **Persona** e restituisca la media delle durate degli interventi sugli apparecchi di quella persona.