

# Analisi e Progettazione del Software

Prova scritta del 5 settembre 2019

Si vuole progettare una classe C++ per un sistema di domotica, cioè di controllo degli elettrodomestici casalinghi (ad es. lavapiatti). Il sistema è caratterizzato da una potenza massima erogabile espressa in KWatt (ad es. 3KW) e da un orizzonte temporale, espresso in minuti a partire dall'istante 0 (ad es. un orizzonte pari a 1440, rappresenta la gestione di una giornata).

Ciascun elettrodomestico è caratterizzato dalla durata totale in minuti del suo unico ciclo di lavoro e da un consumo. Il consumo può variare durante il ciclo di lavoro dell'elettrodomestico, per cui viene espresso da un vettore di lunghezza pari alla durata del ciclo dell'elettrodomestico in Watt. Ad esempio, il vettore  $\langle 50, 50, 50, 50, 100, 100, 100, 100, 70, 70 \rangle$  rappresenta la situazione per cui il ciclo dell'elettrodomestico ha durata 10 minuti e consuma 50W i primi 4 minuti, 100W i successivi 4 minuti e 70W gli ultimi 2.

Quando un elettrodomestico viene inserito nel sistema, questo è spento (cioè non ha un'accensione programmata). La programmazione dell'accensione di un elettrodomestico nel sistema consiste nell'assegnazione del minuto di accensione. Un elettrodomestico, all'interno dell'orizzonte può essere acceso una volta sola (quindi il minuto di accensione è unico) e il ciclo non può essere interrotto. Le operazioni principali del sistema sono le seguenti:

*InserisciElettrodomestico*( $e : \text{Elettrodomestico}$ )

L'elettrodomestico  $e$  viene registrato tra quelli del sistema, nello stato di spento.

*Precondizioni:* L'elettrodomestico  $e$  non è già registrato nel sistema.

*ProgrammaAccensione*( $e : \text{Elettrodomestico}, m : \text{intero}$ )

Viene programmata l'accensione di  $e$  al minuto  $m$ .

*Precondizioni:* L'elettrodomestico  $e$  è già registrato nel sistema ed è spento. Il minuto  $m$  è maggiore o uguale a 0. Il minuto  $m$  più la durata di  $e$  è inferiore all'orizzonte di pianificazione del sistema (il ciclo deve finire entro l'orizzonte).

*PosponiAccensione*( $e : \text{Elettrodomestico}, t : \text{intero}$ )

Viene riprogrammata l'accensione di  $e$  ritardandola di  $t$  minuti rispetto a quanto registrato precedentemente.

*Precondizioni:* L'elettrodomestico  $e$  è già registrato nel sistema ed è acceso. Il tempo  $t$  è maggiore o uguale a 0. Il tempo  $t$  più l'attuale minuto di accensione di  $e$  più la durata di  $e$  è inferiore all'orizzonte di pianificazione del sistema.

*VerificaSoglia*() : *coppia*(*intero*, *intero*)

Viene restituita una coppia di valori, in cui il primo rappresenta il numero di minuti per i quali il consumo totale del sistema è maggiore della sua soglia e il secondo è il massimo sfioramento in Watt tra il consumo in quel minuto e la soglia (0 se non c'è mai sfioramento).

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

**Esercizio 2 (punti 5)** Si scriva la definizione della classe **Sistema** e delle altre classi che compongono il diagramma UML.

**Esercizio 3 (punti 10)** Si scrivano le definizioni dei metodi della classe **Sistema** 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 altre classi del diagramma UML.

**Esercizio 4 (punti 4)** Si scriva l'operatore di output della classe **Sistema**, in modo che stampi anche tutti i dati ad essa collegati nel formato che si ritiene opportuno.

**Esercizio 5 (punti 6)** Si scriva una funzione esterna (non *friend*) che riceva come parametro un oggetto della classe **Sistema** ed un vettore di elettrodomestici. La funzione deve inserire e accendere gli elettrodomestici nell'ordine dato dal vettore fino a che si verifica uno

sforamento della soglia. Il minuto a cui va acceso ciascun elettrodomestico deve essere scelto casualmente tra quelli possibili. A questo scopo, si consideri disponibile la funzione `Random(int a, int b)` che restituisce un intero casuale tra `a` e `b`, estremi inclusi.