

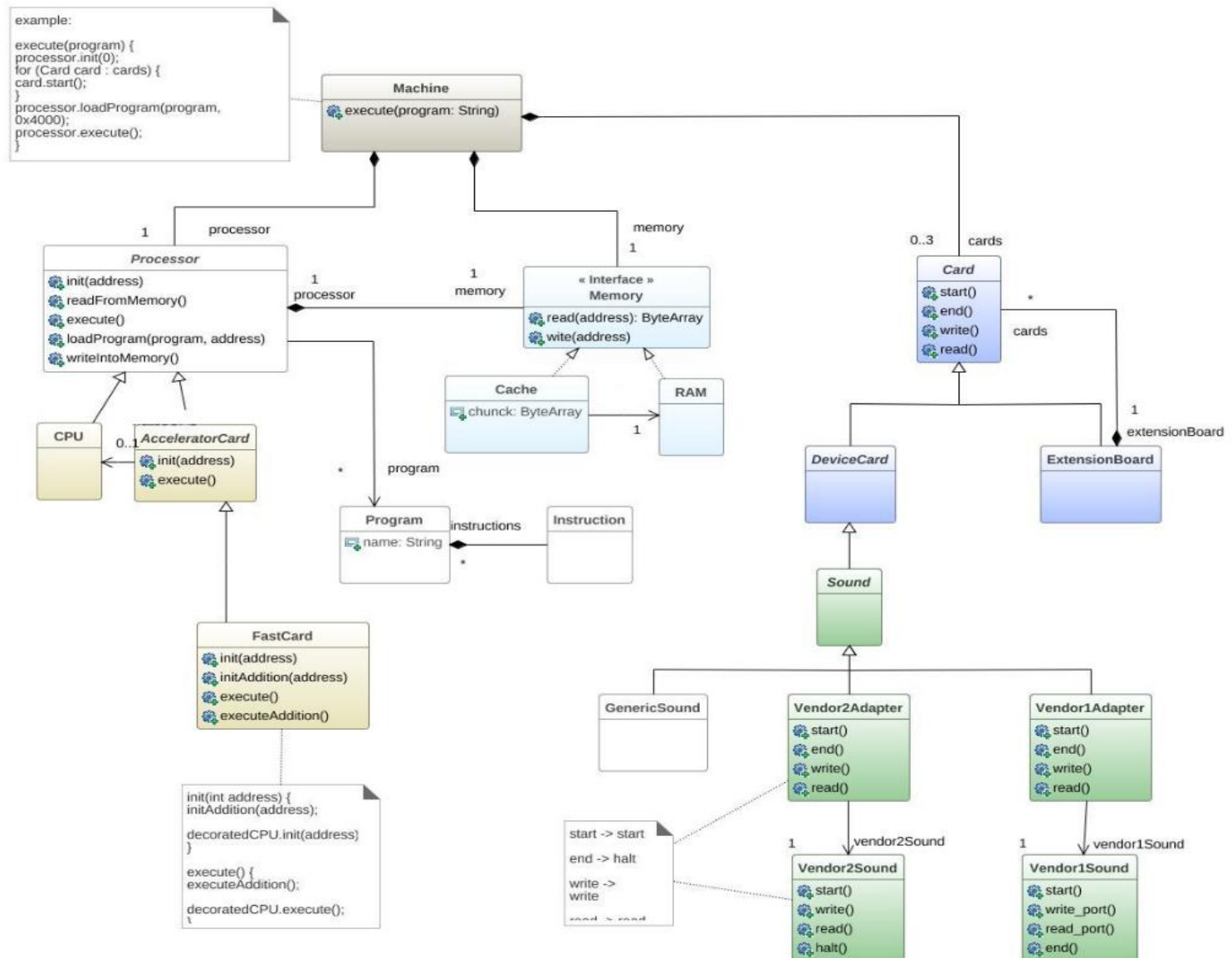
# Feb 2019

Un'app per smartwatch contiene un componente `ActivityTracking` che registra le attività dell'utente. Una attività (classe `Activity`) è caratterizzata da un tipo (nuoto, ciclismo, corsa, ...), una durata in minuti e una stima delle calorie bruciate. Il componente `ActivityTracking` contiene, tra gli altri, un metodo per terminare l'attività corrente. Quando un'attività termina, diversi altri componenti devono essere aggiornati: il componente `DailyCalories`, che salva il numero di calorie bruciate durante la giornata, il componente `DailyActivities`, che salva la lista dei tipi di attività completate durante la giornata, e il componente `DailyActiveTime`, che salva il numero totale di minuti di attività durante la giornata.

Si disegni una possibile architettura per questo sistema usando un diagramma delle classi UML e si fornisca una bozza di implementazione Java per le classi principali. Nello scrivere il codice Java si assuma che la classe `Activity` sia già implementata e si mostri l'implementazione della classe `ActivityTracking`, di una tra `DailyCalories`, `DailyActivities` e `DailyActiveTime`, e di ogni altra classe che possa essere rilevante.

Si favoriscano soluzioni che permettono future modifiche ed estensioni, considerando in particolare il caso in cui l'insieme dei componenti che devono essere aggiornati al termine di un'attività possa cambiare in futuro

# Lug-2018



# Lug-2018

- Indicare se e quali pattern UML sono stati utilizzati nel diagramma. Per ciascun pattern eventualmente individuato, indicare quali classi del diagramma sono coinvolte, quale è il ruolo del pattern, e secondo voi per quale motivo è stato applicato

# Lug-2015

Uno Zoo è composto da più gabbie. Una Gabbia può contenere animali. Esistono tre tipi di animali: Tigre, Leone e Pappagallo.

1. Fornire il diagramma delle classi UML che specifica il problema.
2. Si vuole garantire che una Gabbia possa contenere animali di uno stesso tipo, ma che questo tipo non sia noto a priori. Per questo si usa il pattern factory method. Quando viene invocato il costruttore di Gabbia, viene passato come parametro la factory che crea animali del tipo voluto. Questa factory viene chiamata quando viene invocato il metodo `addAnimal` su una Gabbia.

Mostrare le aggiunte al diagramma UML che rappresentano quanto detto sopra.