

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corsi di Laurea in Ingegneria Informatica ed Automatica
Corso di Progettazione del Software
Esame del **20 ottobre 2022**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda la gestione di macchinari in un impianto industriale e dei programmi (ricette) di lavorazione. Di ogni *macchinario* interessa conoscere il numero di serie (una stringa), il produttore (una stringa), il modello (una stringa) e gli strumenti di lavorazione disponibili (almeno uno). Degli *strumenti di lavorazione* interessa il numero di serie (una stringa), il produttore (una stringa), il modello (una stringa) ed il colore del led di stato (una stringa). Uno strumento è installato su uno ed un solo macchinario. Esistono soltanto due tipi di strumenti di lavorazione: le frese ed i manipolatori. Delle *frese* interessa il tipo di materiale trattato (una stringa), mentre dei *manipolatori* interessa il peso massimo supportato in chilogrammi (un numero decimale). L'impianto industriale definisce per ogni macchinario un insieme di programmi di lavorazione (almeno uno). Ogni *programma* di lavorazione, caratterizzato da un nome (una stringa) è associato ad uno ed un solo macchinario e richiede, in sequenza ordinata, l'impiego di un sottoinsieme di strumenti di tale macchinario (almeno uno). Per ogni strumento utilizzato da un programma si vuole conoscere il tempo in secondi (un intero) per il quale lo strumento deve essere utilizzato. Per semplicità si ipotizzi che un programma non possa utilizzare lo stesso strumento più volte.

Siamo interessati al comportamento degli *strumenti di lavorazione*. Essi sono inizialmente nello stato *non alimentato*. Quando uno strumento sta per essere utilizzato, riceve un evento di *avvio* che lo porta nello stato di *inizializzazione* impostando il led di stato al valore 'giallo' ed avviando un thread di inizializzazione (per semplicità si faccia sì che questo thread attenda un certo numero di secondi). Nel momento in cui il thread di inizializzazione termina, esso genera un evento di *inizializzazione completa* destinato allo strumento stesso ed all'oggetto che ha richiesto l'inizializzazione. Alla ricezione dell'evento di *inizializzazione completa*, lo strumento imposta il led di stato a 'verde' e porta lo strumento in stato *pronto*. In stato pronto, uno strumento può ricevere un evento di utilizzo (con un opportuno parametro di durata) che lo porta in stato *in uso* avviando un thread di utilizzo (anche in questo caso si implementi questo thread come una semplice attesa pari all'argomento passato come parametro) ed impostando il led a 'rosso'. Quando il thread di utilizzo termina, esso genera un evento di *termine utilizzo* che è indirizzato allo strumento stesso ed a chi ne ha richiesto l'uso. Alla ricezione dell'evento di *termine utilizzo*, lo strumento ritorna in stato *pronto* impostando nuovamente il led a 'verde'. Nello stato *pronto* lo strumento può ricevere un evento di *spegnimento* che lo riporta in stato *non alimentato*.

Siamo interessati all'attività principale di esecuzione di un programma. L'attività prende come input un'istanza di programma. A questo punto, invia a tutti gli strumenti interessati dal programma un evento di *avvio* (azione atomica che avvia tutti gli strumenti insieme) e si mette in attesa di tutti gli eventi di *inizializzazione completa* (si supponga di avere un'azione atomica che permetta di attendere il completamento). A questo punto, per ogni strumento contenuto nel programma, in sequenza, l'attività principale avvia due sottoattività in parallelo. La prima sottoattività invia un evento di *utilizzo* allo strumento corrente passando come argomento la durata prevista da programma e si mette in attesa dell'evento di *termine attività*. La seconda sottoattività visualizza a schermo una percentuale di completamento (attività atomica di cui non interessano i dettagli) relativamente alla durata prevista di utilizzo dello strumento corrente. Una volta terminati tutti gli strumenti del programma, viene inviato un evento di *spegnimento* a tutti gli strumenti relativi al programma (azione atomica che spegne tutti gli strumenti insieme).

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare l'analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo di: diagramma delle classi (inclusi eventuali vincoli non esprimibili in UML); diagramma stati e transizioni per la classe *StrumentoDiLavorazione*; diagramma delle attività; specifica del diagramma stati e transizioni; segnatura dell'attività principale, sottoattività non atomiche, atomiche e segnali di input/output. Si noti che NON è richiesta la specifica delle attività. Motivare, qualora ce ne fosse bisogno, le scelte di progetto.

Domanda 2. Effettuare il progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte di progetto. In particolare definire SOLO le responsabilità sulle associazioni del diagramma delle classi. Per le responsabilità utilizzare la seguente notazione: M per molteplicità, O per operazioni, ed R per requisiti.

Domanda 3. Effettuare la realizzazione, producendo un programma JAVA e motivando, qualora ce ne fosse bisogno, le scelte di progetto. In particolare, realizzare in JAVA SOLO i seguenti aspetti dello schema concettuale:

- La classe *StrumentoDiLavorazione* (ed eventuali sottoclassi) con la classe *StrumentoDiLavorazioneFired*, le classi Java per rappresentare le associazioni di cui la classe *StrumentoDiLavorazione* ha responsabilità. Fornire inoltre l'implementazione dei thread per l'inizializzazione e l'uso di uno *StrumentoDiLavorazione*. Come da testo, essi devono aspettare un certo numero di secondi e poi inviare un evento allo strumento di lavorazione stesso ed all'oggetto che ha provocato la transizione di stato.
- L'attività principale e le sue eventuali sottoattività NON atomiche.