

Consegna 4

progettazione soluzione/algoritmo

Lorenzo Dentis, lorenzo.dentis@edu.unito.it

31 marzo 2023

1 Domande

- Leggete l'articolo Programming Patterns and Design Patterns in the Introductory Computer Science Course e scrivete un breve resoconto sull'approccio proposto: siete d'accordo con l'approccio proposto? lo adottereste? immaginate di proporlo in una classe in cui insegnate: quali sono le potenziali difficoltà?
- riportate il testo del problema analizzato per l'attività 2) Progettazione soluzione/algoritmo e la soluzione che avete progettato: suddivisione in sottoproblemi, pattern algoritmici, elementari e ruoli delle variabili.
- descrivete anche eventuali critiche (es. sulla formulazione del problema), problemi e considerazioni emerse durante l'attività

2 Risposte

2.1 1. Commento sull'articolo

In generale mi trovo molto d'accordo con il metodo presentato nell'articolo [1], anche perchè è un metodo molto simile a come mi trovo bene ad imparare nuovi argomenti. Come scritto in sezione 4.2 subito dopo *Intent and Motivation* viene la sezione *Problem Examples* in cui vengono mostrati un paio di esempi di utilizzo del Pattern.

Trovo che, dovendo affrontare un nuovo problema, sia molto d'aiuto vedere subito un esempio di soluzione in atto.

Quando in futuro gli studenti affronteranno un problema potranno ricondursi a questi pattern, conoscendo già una possibile soluzione. Oltretutto questo metodo favorisce il riuso, io stesso il mio secondo anno di università ho dovuto analizzare in Java un CSV molto complesso ed ho scritto del codice che ancora oggi vado a riusare ogni volta che mi trovo ad avere in input un file con tale estensione.

Tutti i Pattern presentati dall'articolo sono inoltre molto ricorrenti, tanto che prima di aver letto l'articolo e non conoscendoli li ho comunque inseriti nella prima bozza dell'attività 2. Inconsciamente, volendo presentare un problema "semplice" e "comune", ho scritto una domanda che richiede proprio di risolvere due dei problemi presentati nell'articolo.

In aula una collega sosteneva che questo metodo potrebbe essere di intralcio all'apprendimento del *problem solving* ed in generale sono d'accordo. In effetti cercare di categorizzare ogni problema e fornire la soluzione agli studenti rende la soluzione molto "meccanica" e non aiuta a sviluppare la capacità di "cercare" un modo di risolvere il problema. Gli autori cercano di guidare gli studenti per non incorrere nella situazione "I do not even know where to start", ma molto spesso capire da dove partire per risolvere un problema è una skill a sè, una abilità che bisogna sviluppare e che torna utile nella realtà quotidiana di uno sviluppatore.

Però l'articolo non propone questo metodo come panacea per insegnare a tutti gli sviluppatori, bensì, citando di nuovo la sezione 4.2 "*The patterns are written to guide a novice with little or no programming experience*". Quindi si sta cercando di insegnare a programmare a persone con scarsa esperienza, decisamente senza la basi per necessarie per iniziare a sviluppare la capacità di "cercare da soli una soluzione", credo che in questo rimanga un metodo eccellente.

2.2 2. Attività 2

2.2.1 Problema posto da Dentis

2.2 2 Algoritmo di esempio 2.2.1 Algoritmo Dentis Lo Chef Tony vuole che la sua cucina sia sempre in ordine e soprattutto sapere quali ingredienti ha a disposizione.

Quindi ha suddiviso gli ingredienti in 5 categorie: *pasta, sugo, carne, verdure e spezie*.

Quando arriva un nuovo carico di materie prime (il numero di ingredienti in un carico può variare) Tony vuole che queste vengano separati automaticamente nelle 4 categorie e gli venga restituito in output quanti elementi di ogni categoria ci sono nel carico appena giunto.

Il problema può essere complicato come nella seguente variante: Il programma deve tenere conto che Tony non è l'unico Chef di Cat&Ring, nella dispensa potrebbero esserci ingredienti rimasti, quindi deve chiedere allo Chef la quantità rimasta e comportarsi di conseguenza.

2.2.2 Soluzione del problema posto da Dentis