

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, concordo molto con il metodo presentato nell'articolo [1], anche perché è simile al modo in cui io stesso preferisco imparare nuovi argomenti. Nella sezione 4.2, subito dopo *Intent and Motivation*, viene presentata la sezione *Problem Examples*, in cui sono mostrati alcuni esempi di utilizzo del pattern. Trovo molto utile vedere prima un esempio di soluzione in atto quando si affronta un nuovo problema.

In futuro, gli studenti potranno far riferimento a questi pattern quando si troveranno di fronte a problemi simili, conoscendo già una possibile soluzione. Inoltre, questo metodo favorisce il riuso. Personalmente, durante il mio secondo anno di università, ho dovuto analizzare un file CSV molto complesso in Java e ho scritto del codice che ancora oggi utilizzo ogni volta che mi trovo ad avere in input un file con tale estensione.

Tutti i pattern presentati nell'articolo sono molto comuni, tanto che, prima di averli letti, li avevo già inseriti nella mia prima bozza dell'attività 2. In modo inconscio, ho scritto una domanda che richiede di risolvere due dei problemi

presentati nell'articolo, volendo presentare un problema "semplice" e "comune".

In aula, una collega ha sostenuto che questo metodo potrebbe ostacolare l'apprendimento del *problem solving* e, in generale, concordo con questa affermazione. 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 evitare la situazione "I do not even know where to start", ma spesso capire da dove partire per risolvere un problema è una skill a sé, un'abilità che va sviluppata e che risulta utile nella vita quotidiana di uno sviluppatore.

Tuttavia, l'articolo non propone questo metodo come una panacea per insegnare a tutti gli sviluppatori, ma, citando ancora la sezione 4.2, "*The patterns are written to guide a novice with little or no programming experience*". Quindi, si cerca di insegnare a programmare a persone con scarsa esperienza, che non hanno ancora le basi per iniziare a sviluppare la capacità di "cercare da soli una soluzione". In questo senso, credo che questo metodo sia 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

Riferimenti bibliografici

- [1] Viera K. Proulx. *Programming Patterns and Design Patterns in the Introductory Computer Science Course*. URL: <http://www.ccs.northeastern.edu/home/vkp/Papers/Patterns-sigcse2000.pdf>.