

La programmazione in modo alternativo:
Haskell

Luca Barra

Anno accademico 2023/2024

INDICE

CAPITOLO 1	INTRODUZIONE	PAGINA 2
1.1	Livello di scuola, classe e indirizzo	2
1.2	Motivazioni e finalità	2
1.3	Prerequisiti	2
1.4	Contenuti	3
1.5	Traguardi e obiettivi Obiettivi di apprendimento — 3	3
1.6	Materiali e strumenti necessari	3
1.7	Linguaggio	3
CAPITOLO 2	SVILUPPO DEI CONTENUTI	PAGINA 6
2.1	Materiale didattico per fasi	6
2.2	Attività unplugged	6
2.3	Esercizi di programmazione	6
CAPITOLO 3	GUIDA PER GLI INSEGNANTI	PAGINA 8
3.1	Consigli sull'utilizzo del materiale didattico	8
3.2	Fasi dell'attività	8
3.3	Snodi e indicatori per fasi	8
CAPITOLO 4	INDICAZIONI PER LA VALUTAZIONE	PAGINA 10
4.1	Valutazione durante le lezioni/attività	10
4.2	Riguardo gli esercizi di programmazione... Block Model — 10 • Misconceptions — 10	10
4.3	Rubrica valutativa	10

Capitolo 1

Introduzione

1.1 Livello di scuola, classe e indirizzo

Domanda 1

A chi è rivolta questa attività?

Risposta: A studenti del quarto/quinto anno di una scuola secondaria di secondo grado di un indirizzo **scientifico**¹.

Domanda 2

Può essere adattata/rivolta a studenti di diverse età e indirizzi?

Risposta: Quest'attività può essere somministrata a studenti al quinto anno di superiori senza alcuna modifica. Può, altresì, essere eseguita da studenti di età inferiore con alcune correzioni (che indicherò nel documento).

1.2 Motivazioni e finalità

Domanda 3

Perché si è scelta proprio quest'attività?

Risposta: L'attività nasce dalla necessità di insegnare ai ragazzi a ragionare in un modo diverso da quello a cui sono abituati. Spesso gli studenti, quando si trovano davanti a un problema che non sanno affrontare, tendono a procedere per tentativi ed errori, ma ciò non è possibile con un linguaggio funzionale. Ciò porta gli studenti a ingegnarsi per trovare una soluzione al problema, che sia adatta ad essere espressa in un modo ricorsivo. La ricorsione è un concetto che spesso viene trascurato nelle scuole superiori, in quanto si tende a prediligere uno stile imperativo o a oggetti.

1.3 Prerequisiti

- ⇒ Utilizzo di base del computer;
- ⇒ Concetto di algoritmo;
- ⇒ Concetto di variabile;
- ⇒ Logica booleana;
- ⇒ Propensione al ragionamento astratto.

¹Liceo Scientifico Op. Scienze Applicate

1.4 Contenuti

I contenuti che presentano un asterisco blu (*) sono adatti anche a studenti di altri indirizzi o di età inferiore (quarta superiore). I contenuti che presentano un asterisco rosso (*) sono opzionali per via della maggiore difficoltà, ma se gli alunni reagiscono bene al resto dell'attività si potrebbero considerare come "bonus".

- ⇒ * Tipi primitivi di Haskell;
- ⇒ * Funzioni;
- ⇒ * Ricorsione;
- ⇒ Inferenza di tipo;
- ⇒ Funzioni a più argomenti;
- ⇒ Funzioni con guardie;
- ⇒ Liste;
- ⇒ Pattern matching;
- ⇒ * Funzioni anonime;
- ⇒ * Alberi.

1.5 Traguardi e obiettivi

1.5.1 Obiettivi di apprendimento

Note:-

I seguenti obiettivi di apprendimento sono stati scritti usando la tassonomia di Bloom rivisitata e sono divisi in traguardi di competenze e obiettivi di conoscenze. Il segno {+} indica un obiettivo procedurale, il segno {++} indica un obiettivo concettuale e il segno {-} indica un obiettivo metacognitivo.

Traguardi di competenze	Obiettivi di conoscenze e abilità
{+} Implementare semplici algoritmi in Haskell.	{+} Saper utilizzare ghci.
{++} Capire l'utilizzo di un determinato costrutto.	{++} Conoscere la sintassi di Haskell.
{-} Costruire una funzione Haskell per risolvere un problema generico.	{++} Utilizzare, in modo opportuno, i costrutti.
{++} Analizzare un codice che utilizza la ricorsione.	{++} Valutare quando convenga utilizzare l'if e quando convenga usare le guardie.
{+} Saper scrivere un programma in modo che sia immediatamente comprensibile da altre persone.	

1.6 Materiali e strumenti necessari

- ⇒ Lavagna;
- ⇒ Gesso;
- ⇒ Computer con installato Haskell/ghci.

1.7 Linguaggio

Linguaggio scelto: *Haskell*.

Domanda 4

Perchè si è scelto questo linguaggio?

Risposta: Haskell è un linguaggio funzionale puro, il che lo rende ideale per insegnare agli studenti un concetto importante come la ricorsione. Inoltre Haskell presenta una sintassi molto semplice e pulita, che non distrae gli studenti dal concetto che si sta insegnando: i codici risultano molto più compatti rispetto ad altri linguaggi (C, Java, ecc...). Infine, Haskell è un linguaggio unico nel suo genere, in quanto "lazy" e "strongly typed", il che lo rende un linguaggio molto interessante da studiare e da approfondire.

Capitolo 2

Sviluppo dei contenuti

2.1 Materiale didattico per fasi

Per gli insegnanti 1

Bisogna inizialmente introdurre in linea generale il contenuto dell'attività agli studenti, facendo attenzione a non perdersi nei dettagli che verranno approfonditi durante le fasi dell'attività. Questo aiuta ad accendere la curiosità degli studenti e a farli partecipare attivamente all'attività.

Note:-

In questo documento ho inserito note specifiche per gli insegnanti (come quella sopra). Tuttavia sono solo brevi suggerimenti, in quanto la guida per gli insegnanti vera e propria è contenuta nella sezione seguente.

Fase 1:

- ⇒ **Consegna:**
- ⇒ **Svolgimento:**
- ⇒ **Discussione:**
- ⇒ **Conclusione:**

Fase 2:

- ⇒ **Consegna:**
- ⇒ **Svolgimento:**
- ⇒ **Discussione:**
- ⇒ **Conclusione:**

Per gli insegnanti 2

2.2 Attività unplugged

2.3 Esercizi di programmazione

Capitolo 3

Guida per gli insegnanti

3.1 Consigli sull'utilizzo del materiale didattico

3.2 Fasi dell'attività

3.3 Snodi e indicatori per fasi

Capitolo 4

Indicazioni per la valutazione

4.1 Valutazione durante le lezioni/attività

4.2 Riguardo gli esercizi di programmazione...

4.2.1 Block Model

4.2.2 Misconceptions

4.3 Rubrica valutativa

