



2020-1-UK01-KA227-YOU-094543

IO1- A3: Compendio di progettazione digitale Modulo: Programmazione informatica

**KA2 - Cooperation for innovation and the exchange of good practices
Partnerships for creativity**



Versione CRONOLOGIA DELLE REVISIONI	Data	Autore	Descrizione	Azione	Pagine
1.0	15/10/2021	CIVIC	Creazione	C	12

(*) Azione: C = Creazione, I = Inserisci, U = Aggiorna, R = Sostituisci, D = Elimina

ID DOCUMENTI DI RIFERIMENTO	Riferimento	Titolo
1	2020-1-UK01-KA227-YOU-094543	Proposta HerTour4Youth
2		

ID	Riferimento	Titolo
1	Consegnabile IO1.A2	Elaborazione della metodologia di formazione
2		

DOCUMENTI





Contenuti

1. introduzione5
 - 1.1 Risultati di apprendimento5
 - 1.2 Parole chiave5
 - 1.3 Tempo di seduta stimato5
 - 1.4 Glossario dei termini5
2. Programmazione computer8
 - 2.1 introduzione8
 - 2.2 Programmatore di computer10
 - 2.1.1 Linguaggi di programmazione12
 - 2.1.2 Processo di programmazione15
 - 2.1.3 Principi di programmazione19
 - 2.1.4 Come puoi diventare un programmatore di computer?20
3. Valutazione23
 - 3.1 Valutazione della conoscenza23
 - 3.2 Valutazione delle competenze25
4. Riferimenti26



1. Introduzione

Se ti stai chiedendo cosa sia la programmazione di un computer, immagina un molto semplice che puoi trovare a casa (può essere un termostato, il tuo router per la connessione ecc.). Ora, sappi solo che quel computer non funziona da solo, ma è stato programmato da qualcuno. Quel qualcuno è un programmatore di computer. Ogni computer ha bisogno di una serie di istruzioni per funzionare senza problemi. Questo è tutto ciò che riguarda la programmazione del computer: può consistere in un semplice insieme di istruzioni per facilitare azioni specifiche fino ad un insieme complesso di istruzioni che coinvolgono la lettura e lo smistamento dei dati.

1.1 Risultati di apprendimento

Dopo aver completato questo modulo, sarai in grado di:

- Avere una conoscenza di base dei principi di programmazione
- Avere una conoscenza più approfondita su come funzionano gli strumenti ICT
- Essere in grado di applicare i principi di programmazione

1.2 Parole chiave

- ICT (TIC in italiano)
- Programmatore di computer
- Linguaggi di programmazione

1.3 Tempo di seduta stimato

1 ora.

1.4 Glossario dei termini



TIC: Sta per "Tecnologie dell'informazione e della comunicazione". Le TIC si riferiscono alle tecnologie che forniscono l'accesso alle informazioni attraverso le telecomunicazioni. (Termini tecnici, 2021)

Programmatore di computer: un programmatore di computer è la persona responsabile della creazione di istruzioni per l'esecuzione di un computer scrivendo e testando il codice che consente alle applicazioni e ai programmi software di funzionare correttamente. (Tecnopedia, 2014)

Linguaggi di programmazione: i linguaggi di programmazione per computer sono uno qualsiasi dei vari linguaggi atti ad esprimere una serie di istruzioni dettagliate per un computer digitale. (D. Hemmendinger, 2021)

Analogico: Aggettivo che si riferisce a un meccanismo o dispositivo in cui l'informazione è rappresentata da grandezze fisiche continuamente variabili. È contrario al digitale. (Merriam Webster, 2021)

Intelligenza artificiale: "L'intelligenza artificiale sfrutta computer e macchine per imitare le capacità di risoluzione dei problemi e decisionali della mente umana, senza limitarsi a metodi biologicamente osservabili" (IBM, 2020)

Apprendimento automatico: L'apprendimento automatico è una branca di intelligenza artificiale (IA) e dell'informatica che si concentra sull'uso di dati e algoritmi per imitare il modo in cui gli esseri umani apprendono, migliorandone gradualmente l'accuratezza. (IBM, 2020)

Binario: Il binario è un sistema numerico in base 2 inventato da Gottfried Leibniz che è composto solo da due numeri o cifre: 0 (zero) e 1 (uno). Questo sistema di numerazione è la base per tutto il codice binario, che viene utilizzato per scrivere



**dati digitali come le istruzioni del processore del computer utilizzate ogni giorno.
(Computer Hope, 2021)**



2. Programmazione computer

2.1 Introduzione

Nel nostro mondo guidato dai computer, conosciamo le loro straordinarie capacità. Il rapido progresso tecnologico a cui abbiamo assistito negli ultimi decenni ha un effetto di vasta portata su come funziona il mondo. Devi guardare sotto la superficie per vedere dove sono avvenuti i cambiamenti più significativi. I computer hanno sostituito milioni di ore lavorative e magazzini di macchine analogiche con sistemi più veloci, più sicuri e più affidabili.

Poiché i computer funzionano su codice, è evidente il motivo per cui puoi trovarlo ovunque. I computer continueranno a sostituire la tecnologia obsoleta in tutto, dai forni a microonde alle centrali elettriche. E la presenza del codice nella nostra vita quotidiana aumenterà.

Inoltre, quasi tutti oggi possiedono un laptop che può essere utilizzato per semplici processi informatici, come la creazione di un foglio di calcolo o la creazione di semplici documenti. Tuttavia, ci sono anche persone che usano il proprio laptop in modo più complesso e professionale. Ciò potrebbe essere necessario per completare milioni di transazioni finanziarie al giorno e controllare l'infrastruttura che rende possibile la vita moderna.

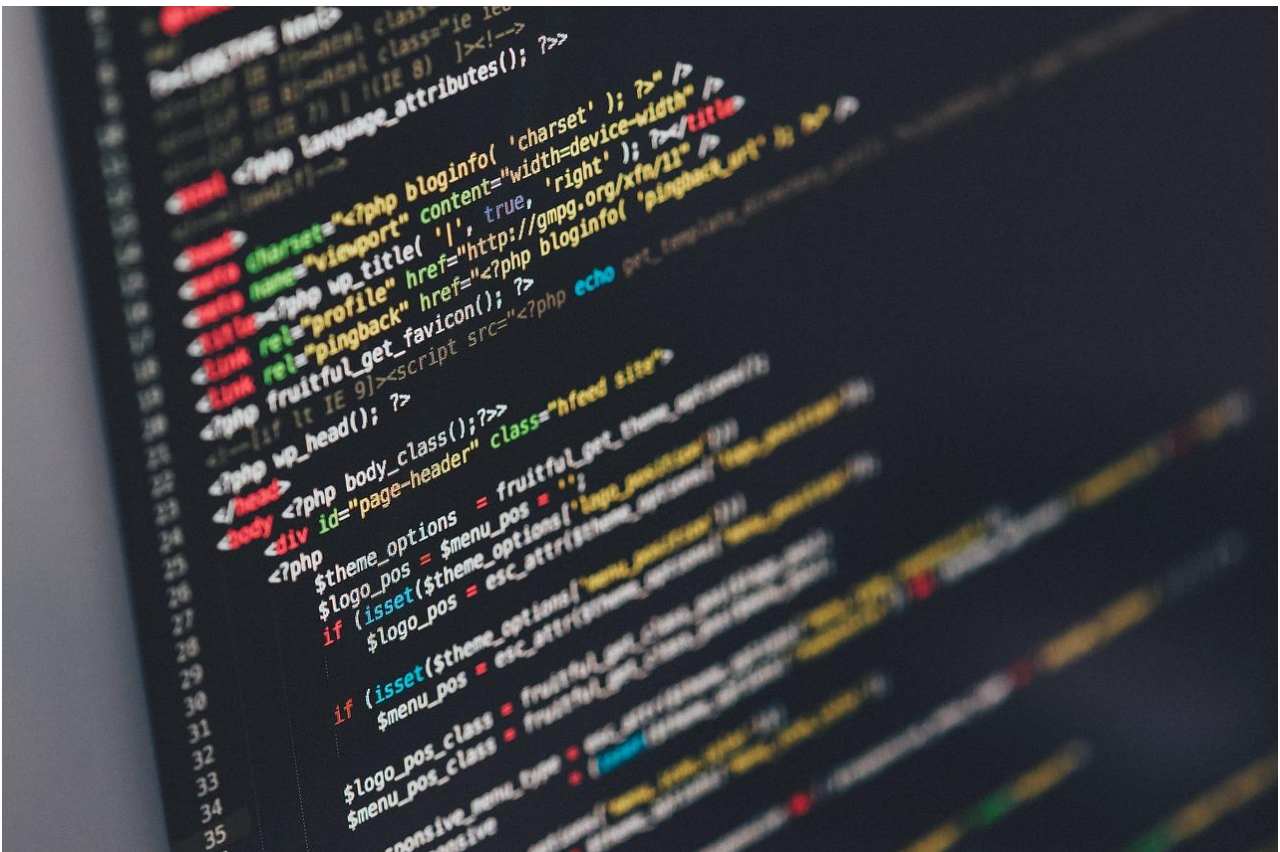
Detto questo, è importante ricordare che nessun computer può fare nulla se non è programmato da un essere umano. In questo caso si tratta della figura del programmatore di computer.

Il programmatore di computer è la figura professionale responsabile della scrittura di codici, o codifica, che istruiscono su come dovrebbe funzionare un computer, un'applicazione o un programma software.



Come accennato in precedenza, questo insieme di istruzioni può essere abbastanza semplice e l'intero processo di codifica potrebbe richiedere solo l'aggiunta di due numeri, ma potrebbe anche comportare un aspetto più complesso come la lettura dei dati dai sensori di temperatura per regolare un termostato, l'ordinamento dei dati per completare intricate pianificazioni o rapporti critici o accompagnare i giocatori attraverso mondi a più livelli e sfide nei giochi.

Mentre l'informatica si occupa di idee teoriche di alto livello, quasi ogni aspetto della vita moderna si basa sulla codifica. Ogni applicazione sul nostro telefono, tablet o computer utilizza i linguaggi del computer per l'esecuzione.



Fonte: <https://pixabay.com/photos/code-coding-computer-data-1839406/>

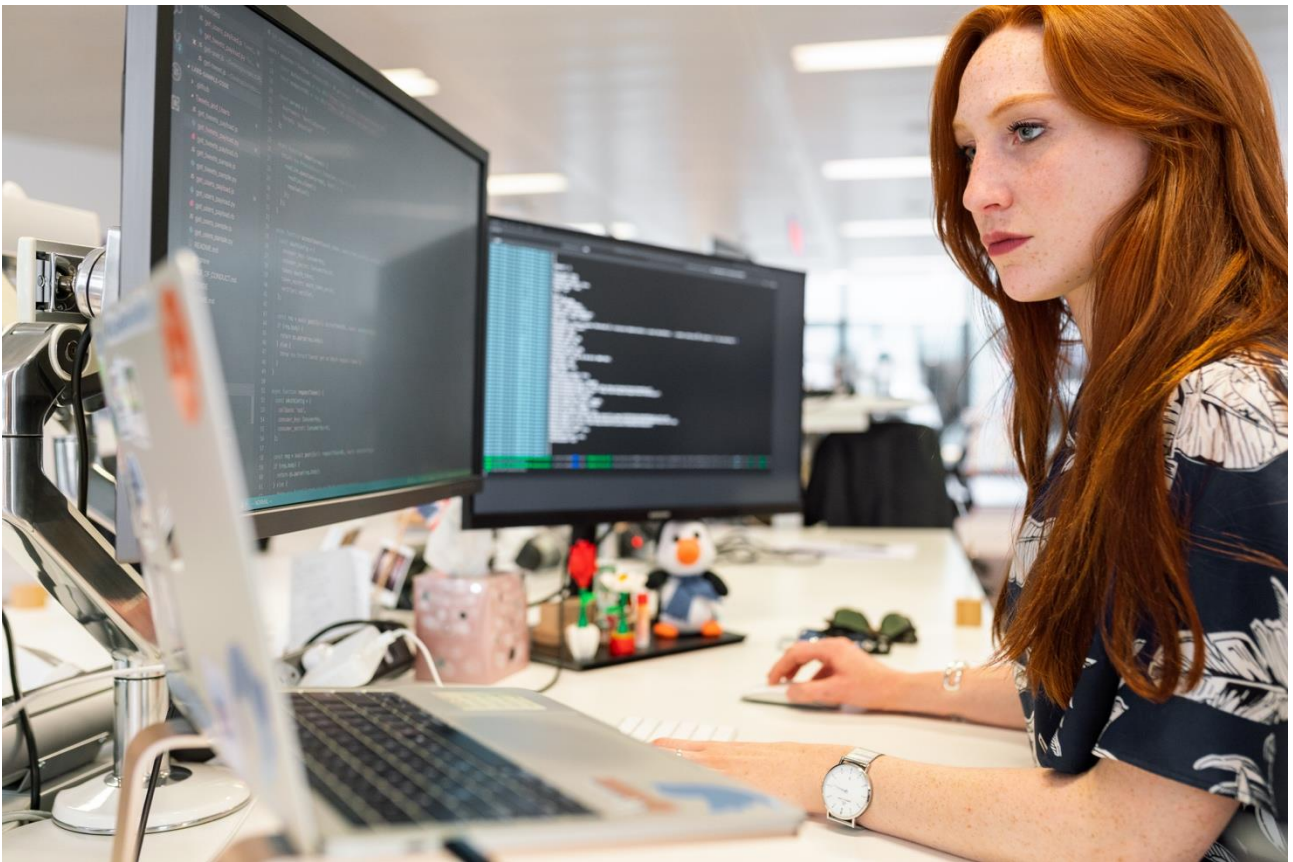
Lo usano anche altri sistemi digitali come smart TV e calcolatrici. Praticamente ogni nuova auto lo usa per controllare tutto, dai sistemi di aria condizionata agli iniettori di carburante. Le città utilizzano i computer per gestire i segnali stradali.

I sistemi che prima erano analogici sono ora ottimizzati utilizzando sistemi computerizzati. Ciò consente agli ingegneri di costruire un sistema, una struttura e una macchina più efficienti e meno costosi. Inoltre, alcuni dei campi tecnici più avanzati (come l'intelligenza artificiale e l'apprendimento automatico) utilizzano la codifica.¹

2.2 Programmatore di computer

¹Weinstein, J. (24 gennaio 2021). Che cos'è la codifica? Definizione e usi della codifica. Karma di carriera.
<https://careerkarma.com/blog/what-is-coding-used-for/>





Snostra:<https://unsplash.com/photos/64YrPKiguAE>

Come accennato in precedenza, un programmatore di computer è la persona responsabile della creazione di istruzioni per l'esecuzione di un computer scrivendo e testando il codice che consente alle applicazioni e ai programmi software di funzionare correttamente.

Un programmatore di computer può anche essere definito programmatore, sviluppatore o ingegnere del software. Inoltre, il termine è spesso usato per riferirsi a uno sviluppatore di software autonomo, sviluppatore di applicazioni mobili, sviluppatore Web, analista di software, sviluppatore di firmware incorporato e così via.

Un programmatore di computer è un professionista qualificato che codifica, verifica, esegue il debug e mantiene le istruzioni complete note come programmi per computer che i computer dovrebbero seguire per eseguire le loro funzioni specifiche².

Alcuni dei compiti comuni che un programmatore di computer deve padroneggiare comprendono:

- Testare le prestazioni del software.
- Risolvere problemi del software del computer.
- Modificare programmi software per migliorare le prestazioni.
- Scrivere il codice di programmazione del computer.
- Collaborare con gli altri per risolvere i problemi di tecnologia dell'informazione.

È comune classificare i programmatori di computer in due tipi: programmatori di sistemi e programmatori di applicazioni.

Entrambi i ruoli codificano. La differenza tra i due è che i programmatori di applicazioni eseguono la codifica per gestire un'attività particolare, come la codifica di un programma per monitorare l'inventario all'interno di un'azienda.

D'altra parte, i programmatori di sistemi codificano i programmi per mantenere e controllare il software di sistema, inclusi i sistemi di gestione dei database e i sistemi operativi (OS).

2.1.1 Linguaggi di programmazione

²Tecnopedia. (2014, 30 luglio). Programmatore di computer. Techopedia.Com.
<https://www.techopedia.com/definition/6589/computer-programmer>

Proprio come le lingue basate sull'uomo, anche i computer hanno le loro lingue. In effetti, esistono più linguaggi di programmazione per computer che i programmatori possono utilizzare per comunicare con un computer per eseguire una serie di attività specifiche.

La parte del linguaggio che un computer può capire è chiamata codice "binario". La traduzione del linguaggio di programmazione in binario è nota come "compilazione". Ogni linguaggio di programmazione ha le sue caratteristiche distinte. Tuttavia, a volte ci sono somiglianze tra di loro³.

Alcuni dei linguaggi di programmazione per computer più comuni sono:

- C
- SQL
- PHYTON
- JAVA
- R
- HTML⁴

³Scrittori, S. (2021, 27 settembre). Guida ai linguaggi di programmazione | ComputerScience.org. Ottieni un'istruzione di cui il mondo ha bisogno | Informatica.Org. <https://www.computerscience.org/resources/computer-programming-languages/>

⁴Gallagher, J. (2021, 4 maggio). I linguaggi di programmazione più popolari. Karma di carriera. <https://careerkarma.com/blog/top-programming-languages-2021/>





Fonte: <https://pixabay.com/illustrazioni/linguaggi-di-programmazione-icon-898961/>

Molti linguaggi di programmazione sono relativamente semplici, ma fanno cose diverse. Ad esempio, uno dei linguaggi più popolari, JavaScript, viene utilizzato principalmente per le pagine Web e lo sviluppo front-end. D'altra parte, Python viene utilizzato sia per programmi software completi che per siti Web.

Qualunque cosa scelga uno sviluppatore di software è solitamente a sua discrezione, poiché più di un linguaggio di programmazione può spesso funzionare per lo stesso scopo.

Conoscere uno o più di questi linguaggi di programmazione è essenziale se vogliamo raggiungere una carriera non solo come programmatore di computer ma anche come ingegnere del software o qualsiasi lavoro nel campo della scienza dei dati che richieda conoscenze high tech.

2.1.2 Processo di programmazione

L'obiettivo principale della programmazione è generalmente quello di creare una soluzione per risolvere un problema. Il problema può essere qualcosa di grande su scala internazionale o semplicemente qualcosa di piccolo per sollevarci dalla noia.





Fonte: <https://pixabay.com/illustrazioni/linguaggi-di-programmazione-icon-898961/>

Lo sviluppo di un programma comporta passaggi simili a qualsiasi attività di risoluzione dei problemi. Ci sono quattro fasi principali del processo di programmazione:

- **Definire il problema:** in questa fase stai cercando di identificare il problema. Tuttavia, sarebbe saggio dire che ciò che stai veramente cercando di identificare è la soluzione del problema, dal momento che stai cercando di definire ciò che vuoi ottenere.

La prima cosa da fare è analizzare i requisiti. In effetti, devi scoprire cosa dovrà fare il tuo programma.

La seconda cosa da fare quindi è osservare l'elenco dei requisiti e decidere esattamente cosa dovrebbe fare la tua soluzione per soddisfarli. A volte un singolo problema potrebbe richiedere più soluzioni, quindi è importante capire qual è la soluzione migliore per ogni problema.

Quindi, stai cercando di specificare, ancora una volta, quale soluzione dovresti adottare per il tuo problema.

- Pianificazione e progettazione della soluzione: questo è generalmente il compito più difficile di tutti perché richiede di capire come trasformare tali specifiche e requisiti in un programma funzionante e funzionale.

Progettare una soluzione significa descrivere a un livello superiore tutti i passaggi che il computer deve seguire per funzionare correttamente o svolgere un compito specifico.

Questo compito è il più difficile per diversi motivi: per esempio potrebbe essere necessario imparare un po' di più sulle funzioni del tuo computer e sui linguaggi di programmazione scelti per vedere quali cose possono essere facili o difficili; inoltre, come accennato in precedenza, un problema può richiedere diverse soluzioni e una volta individuata una soluzione, è importante analizzarne tutti i punti di forza e di debolezza e fare la scelta giusta.

Tuttavia, una volta completata la parte difficile, la parte di progettazione dovrebbe essere completata. Ciò significa che hai un'idea molto chiara di come il computer soddisferà le tue specifiche, soddisferà i tuoi requisiti e infine risolverà il tuo problema originale.

- **Codificare il programma:** In questa attività dovrai istruire il tuo computer su come eseguire il tuo progetto. Questa fase comprende 3 fasi:
 - **Codifica:** Codificare significa tradurre il tuo progetto in un programma funzionale, utilizzando un linguaggio di programmazione. Questa fase implica praticamente che ti siedi e inizi a digitare i codici sul tuo computer. Proprio come un saggio scritto, la codifica di un programma richiede di includere cose come titoli, pagina dei contenuti, introduzione e riferimenti. Una volta terminata la codifica, è necessario inviarla al computer per vedere come reagisce il computer e cosa ne fa.
 - **Compilare:** La fase di compilazione richiede di trasformare il programma scritto in un linguaggio comprensibile al computer, da qui il codice macchina binario. Ci sono alcuni programmi chiamati compilatori che possono farlo per te. Tuttavia, devi essere molto preciso quando scrivi il tuo programma in un compilatore, perché rileverà ogni minimo errore.
 - **Debug:** come accennato prima il compilatore rileverà ogni singolo errore commesso nel programma. Non preoccuparti, è una cosa molto comune. Ecco perché questa fase è inserita. Una volta identificati tutti i bug nel tuo programma, grazie all'aiuto dei compilatori, guarda di nuovo il programma originale, individua gli errori, correggi il codice e ricompilalo. Questo ciclo di codice -> compila -> debug verrà spesso ripetuto molte volte prima che il compilatore sia soddisfatto e avrai finalmente un programma che funziona. Questo potrebbe richiedere del tempo.
Puoi anche decidere di scrivere piccole sezioni del tuo programma, compilarle ed eseguirne il debug, piuttosto che concentrarti sull'intero programma tutto in una volta.
- **Testare il programma:** La parte finale del processo di programmazione consiste nel testare la tua creazione e verificare che funzioni esattamente



come volevi. So che potresti pensare che il compilatore ti abbia già fornito il programma corretto, tuttavia ciò non significa che il codice corretto possa risolvere quello che era il tuo problema originale. Questo è ciò che ti resta da controllare.

Se vengono rilevati eventuali errori, l'unica opzione è quella di rivisitare il programma, correggere l'errore modificando il codice e ricompilarlo. Ricorda di fare attenzione quando modifichi qualcosa nel codice, perché la modifica di una piccola cosa potrebbe influire su altre cose nell'intero programma. Questa fase può essere paragonata a un'altra fase di debug.

Infine, dopo aver compilato di nuovo, correggi gli errori, esegui il debug e il test e dopo esserti assicurato che il tuo programma funzioni secondo i tuoi requisiti e le tue specifiche, avrai una soluzione per il tuo problema!⁵

2.1.3 Principi di programmazione

A questo punto sappiamo che una delle caratteristiche principali della programmazione informatica è la scrittura di codici. Sia che lavori da solo o all'interno di un team, ricorda che i tuoi codici devono essere facili da leggere e da mantenere per le altre persone.

Ecco perché ogni programmatore dovrebbe conoscere alcuni principi base di programmazione per scrivere codici facili, con variabili pulite e in grado di resistere anche dopo test ed eventuali modifiche.

Alcuni di questi principi sono:

⁵Il processo di programmazione. (2020). Cs.Bham.Ac.Uk.
<https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html>



- **KISS: Mantienilo semplice, stupido:** Il principio KISS è uno dei principi più importanti da rispettare nel mondo della programmazione. Significa che dovresti scrivere il codice il più semplice possibile. Non devi provare a metterti in mostra con il codice avanzato. Se puoi scriverlo in una riga, usa una riga. E ricorda che alcuni codici devono essere rivisitati dopo mesi, quindi tienilo a mente e rendilo semplice.
- **ASCIUTTO: Non ripeterti:** Questo principio ti aiuta a ricordare un errore molto comune nella codifica: le ripetizioni. Evita qualsiasi duplicazione di dati e usa la logica.
- **Aperto/chiuso:** Questo principio ti assicura di ricordare di mantenere i tuoi codici aperti per le estensioni ma chiusi per le modifiche. Questa regola è utile quando stai fornendo una libreria di servizi aperta per il suo utilizzo da parte di altri. In parole molto semplici, se qualcun altro modifica il tuo codice, il codice si interromperà e non funzionerà come dovrebbe. Ecco perché dovresti rilasciare solo codici che impediscono la modifica diretta e incoraggiano l'estensione
- **Unica responsabilità:** Questo principio afferma che ogni classe o modulo in un programma dovrebbe fornire solo una funzionalità specifica. Ecco perché in questo modo il codice viene mantenuto semplice ed evita anche un processo di debug più complicato; In secondo luogo, diventa più difficile creare funzionalità aggiuntive per un modulo specifico.
- **YAGNI (You Aren't Going to Need It) Non ne avrai bisogno:** Questo principio afferma che non dovresti mai provare a scrivere codici per una funzionalità di cui potresti aver bisogno in futuro. Ciò significherebbe che stai cercando di risolvere un problema che non esiste.

2.1.4 Come puoi diventare un programmatore di computer?



Essere un programmatore di computer può aiutarti a trovare un lavoro nella progettazione di sistemi informatici e nel settore dei servizi correlati. Tuttavia, molti campi dell'informatica richiedono la figura di un programmatore di computer.

Questi includono ad esempio:

- Ingegneria di software
- Sviluppo di software
- Ingegneria Informatica
- Computer grafica
- Intelligenza artificiale

In genere, ai programmatori di computer è richiesto un titolo universitario in dati e informatica, che può darti le basi per l'inizio della tua carriera. Tuttavia, molti programmatori possono anche essere appassionati autodidatti pieni di un interesse persistente per la programmazione.

Che tu decida di iniziare la tua carriera con una laurea o da autodidatta, l'importante è migliorare le tue conoscenze sui linguaggi di programmazione, interessarti alla programmazione informatica e anche tenerti informato e non smettere mai di imparare, perché la programmazione informatica è una disciplina in costante aggiornamento.





Fonte: <https://pixabay.com/photos/gioco-da-tavola-scrabble-4370255/>

Un'altra cosa importante che potrebbero richiedere molte posizioni lavorative nella programmazione di computer sono le certificazioni professionali e senza scopo di lucro disponibili.

Alcuni di essi includono:

- CISCO - Certified Network Associate, Certified Network Profession Routing and Switching, Certified Network Associate Security Credential
- Microsoft - Sviluppatore di soluzioni certificato per applicazioni Web, Certified Solutions Associate Windows Server
- Associazioni professionali - Certificazione Software Development Associate, Comptia's Security+, Comptia's A+ Certification, Comptia's Linux+

- **Senza scopo di lucro: professionista certificato per la sicurezza dei sistemi informatici, responsabile certificato per la sicurezza delle informazioni, credenziali certificate per il professionista del ciclo di vita del software sicuro⁶**

3. Valutazione

3.1 Valutazione della conoscenza

Valutazione simile a un quiz basata sul contenuto principale. Si prega di contrassegnare la risposta corretta in grassetto quando richiesto. Includi 10 domande per il tuo modulo. Aumenta gradualmente il livello di difficoltà.

Domanda 1: Il programmatore di computer è la figura professionale responsabile della scrittura di codici, o codifica, che istruiscono su come dovrebbe funzionare un computer, un'applicazione o un programma software.

[Vero] [falso]

Domanda 2: Il processo di codifica può solo far funzionare un computer.

⁶Cote, J. (2021, 19 agosto). Che cos'è la programmazione informatica? Come diventare un programmatore di computer. SNHU.Edu. <https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming>



[Vero] [falso]

Domanda 3: quali compiti deve padroneggiare un programmatore di computer?

[Risoluzione dei problemi del software del computer] [Modifica dei programmi software per migliorare le prestazioni] [Scrittura del codice di programmazione del computer] [Testare le prestazioni del software] [Tutti i precedenti]

Domanda 4: La parte del linguaggio che un computer può capire è chiamata "binario". Come viene chiamato il processo di traduzione del linguaggio di programmazione in binario?

[Codifica] [Compilazione] [Debug]

Domanda 5: Qual è l'ordine corretto delle attività del processo di programmazione?

[Codificare il programma, Testare il programma, Definire il problema, Pianificazione e progettazione della soluzione]
[Pianificare e progettare la soluzione, Definire il problema, Codificare il programma, Testare il programma]
[Definire il problema, Pianificare e progettare la soluzione, Codificare il programma, Test del programma]

Domanda 6: Quali sono le tre fasi della fase di Codifica all'interno del processo di Programmazione?

[Testing] [Codifica] [Sistemazione] [Compilazione] [Debug]

Domanda 7: cosa i principi di programmazione ricordano ai programmatori di fare durante la programmazione?

[usare codici che altri programmatori non possono modificare in futuro] [usare codici complessi che migliorano le funzioni] [scrivere codici facili] [usare variabili pulite che sono solide dopo il test]

Domanda 8 (corrispondenza): Abbina i termini con le loro definizioni.

Binario: Un sistema numerico in base 2 inventato da Gottfried Leibniz che è composto solo da due numeri o cifre: 0 (zero) e 1 (uno).

TIC: Questa parola si riferisce alle tecnologie che forniscono l'accesso alle informazioni attraverso le telecomunicazioni.



Apprendimento automatico: un ramo di intelligenza artificiale (IA) e dell'informatica che si concentra sull'uso di dati e algoritmi per imitare il modo in cui gli esseri umani apprendono, migliorandone gradualmente l'accuratezza

Analogico: Aggettivo che si riferisce a un meccanismo o dispositivo in cui l'informazione è rappresentata da grandezze fisiche continuamente variabili. È contrario al digitale.

Domanda g (corrispondenza): Abbina i concetti alle loro spiegazioni.

Programmatore di computer: persona responsabile della creazione di istruzioni per l'esecuzione di un computer scrivendo e testando il codice che consente alle applicazioni e ai programmi software di funzionare correttamente.

Compilazione: processo di traduzione del linguaggio di programmazione in binario.

Debug: Processo che prevede l'identificazione di tutti i bug del programma, grazie all'aiuto dei compilatori, rivedendo il programma originale, identificando gli errori, correggendo il codice e ricompilarlo.

Codifica: Processo di traduzione del progetto in un programma funzionale, utilizzando un linguaggio di programmazione.

Test: Processo che prevede la revisione del programma per verificare che funzioni esattamente nel modo desiderato.

3.2 Valutazione delle competenze

Il pensiero analitico è una caratteristica importante in un programmatore. Viene anche definito ragionamento analitico, pensiero astratto o pensiero critico. Le persone che possono pensare in modo logico sono in grado di analizzare i problemi ed escogitare soluzioni. Questo non è solo prezioso quando si sviluppano programmi, ma è vitale in qualsiasi situazione che richieda un pensiero razionale. Le persone che sono logiche:

- analizzano le informazioni o le risorse relative a un'attività
- osservano attentamente cosa sta succedendo
- studiano le informazioni in modo obiettivo per determinare se sono pertinenti o vere
- si concentrano sui fatti, non sulle emozioni



- **sviluppano soluzioni ai problemi basate sui fatti**
- **delineano chiaramente le idee scomponendole in parti**
- **prestano attenzione ai dettagli**
- **testano l'efficacia di una soluzione e apportano revisioni**

Esistono molti esercizi e attività per praticare queste abilità. Alcuni di loro sono molto semplici, come leggere libri, giocare a giochi di logica o entrare in un club di dibattito per mantenere il cervello stimolato.

Nel caso di un programmatore, un modo semplice per migliorare le tue capacità di pensiero analitico è seguire e tenere a mente questa guida all'approccio in 5 fasi:

- 1. Analizzare il problema: include la raccolta di informazioni, l'esame delle risorse e la determinazione delle lacune nelle abilità o nelle conoscenze;**
- 2. Formulare un piano: raccogliere idee e organizzare i pensieri in un piano;**
- 3. Sviluppare un codice per risolvere il problema: iniziare a scrivere il codice (è essenziale la conoscenza dei linguaggi informatici);**
- 4. Valutare la soluzione e rivedere il codice: rivedere la soluzione e identificare le aree di miglioramento, se necessario;**
- 5. Giustificare le decisioni: presentare prove che spieghino perché il programma è una soluzione accettabile.**

Maggiori informazioni qui: <https://www.technokids.com/blog/teaching-strategies/its-easy-to-improve-logical-thinking-with-programming/>

4. Riferimenti

**Cote, J. (2021, 19 agosto). Che cos'è la programmazione informatica? Come diventare un programmatore di computer. SNHU.Edu.
<https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming>**

Weinstein, J. (24 gennaio 2021). Che cos'è la codifica? Definizione e usi della codifica. Karma di carriera.
<https://careerkarma.com/blog/what-is-coding-used-for/>

**Scrittori, S. (2021, 27 settembre). Guida ai linguaggi di programmazione | ComputerScience.org. Ottieni un'istruzione di cui il mondo ha bisogno | Informatica.Org.
<https://www.computerscience.org/resources/computer-programming-languages/>**



Gallagher, J. (2021, 4 maggio). I linguaggi di programmazione più popolari. Karma di carriera. <https://careerkarma.com/blog/top-programming-languages-2021/>

Tecnopedia. (2014, 30 luglio). Programmatore di computer. Techopedia.Com.
<https://www.techopedia.com/definition/6589/computer-programmer>

Il processo di programmazione. (2020). Cs.Bham.Ac.Uk.

<https://www.cs.bham.ac.uk/%7Erxb/java/intro/2programming.html>

