
GIT

Cosa è GIT, clonazione di una repo sul computer, creazione di commit

Carlo Tacchella

CHI SONO

Carlo Tacchella - tacchella.carlo@gmail.com

Cosa è GIT?

Git è un sistema di controllo della versione distribuita, serve agli sviluppatori per collaborare tra loro attraverso repository comuni. Questi repository locali completamente funzionali semplificano il funzionamento offline o remoto e integrano il lavoro di vari programmatori su un progetto comune.

GIT



IL CODICE OPEN SOURCE

- libera condivisione e circolazione delle informazioni relative ai programmi
- qualità elevata
- maggiore stabilità e protezione
- supporto dalla community
- trasparenza

Ci sincronizziamo con Github Desktop alla repo del corso

Proviamo ad eseguire un commit

Proviamo a ripristinare una vecchia versione della repo

Uso Gitignore

Il gitignore è un semplice file di testo nascosto, inizia infatti con .

Serve per indicare quali file della cartella git non devono essere più seguiti e condivisi da GIT

Uso Gitignore

Un esempio può essere:

a.txt <- ignora il file di testo a.txt

**/.idea/

**/build/

**/*.class <- ignora i file con estensione .class in tutte le cartelle e sottocartelle

**/nbproject/private/

Versioning

Versioning

Il versioning del software si riferisce alle diverse versioni rilasciate dalla società di software. Ogni versione di un software che viene rilasciato riceve un numero identificativo.

Come denominare le versioni del software?

Alcuni dei metodi più comuni di numerazione delle versioni software sono:

Numerazione semantica: tecnica di numerazione a tre cifre basata su una sequenza Major.Minor.Patch

Data di rilascio: il numero della versione del software è la data del rilascio. Ad esempio, 20.06 o 06.20 per giugno 2020.

Codici alfanumerici

Numerazione sequenziale

Cosa è il versioning semantico?

Questo metodo utilizza tre numeri: Major.Minor.Patch. Quando viene effettuato un aggiornamento maggiore, minore o patch, aumenta il numero corrispondente.

- Le modifiche alla versione **Major** si riferiscono alle modifiche delle API incompatibili.
- Le modifiche alla versione **Minor** si riferiscono all'aggiunta di nuove funzionalità in modo retrocompatibile.
- Le modifiche alla versione **Patch** si riferiscono alla correzione di bug in modo anch'esso retrocompatibile.

Esempio

Il numero aumenta di una cifra a seconda del tipo di modifiche apportate nella nuova versione.

Ad esempio, una versione del software 1.3.1 indica che il software si trova nella sua prima versione major, con tre funzionalità nuove e una patch implementata. Quando uno sviluppatore corregge un bug in quella versione, il prossimo rilascio si chiama versione 1.3.2.

Conflitti in git

Conflitto in git

In Git, un conflitto o conflict può presentarsi quando due o più cambiamenti concorrenti sono stati applicati sulla stessa linea di codice, o quando una persona sta lavorando su un file che è stato eliminato.

Cosa devo tenere?

I conflitti tra due cambiamenti sullo stesso codice, avvengono quando si esegue un merge di due branch. In questo caso bisogna scegliere manualmente quale dei due cambiamenti, vogliamo tenere nel codice.

Esempio di conflitto

Per esempio se due programmatori stanno modificando lo stesso file e linea su due branch diversi, Git mostrerà un errore di conflitto quando si proverà a unire (merge) i due branch.

Come si presenta un conflitto

Utilizzando il vostro text edito preferito, aprite il file contenente il conflitto (Per noi fileConflitto.js)

Git automaticamente inserisce nel file dei caratteri per mostrarvi dove il conflitto sia presente.

L'inizio del conflitto è dato da:

```
<<<<<< HEAD
```

I due cambiamenti concorrenti sono separati da:

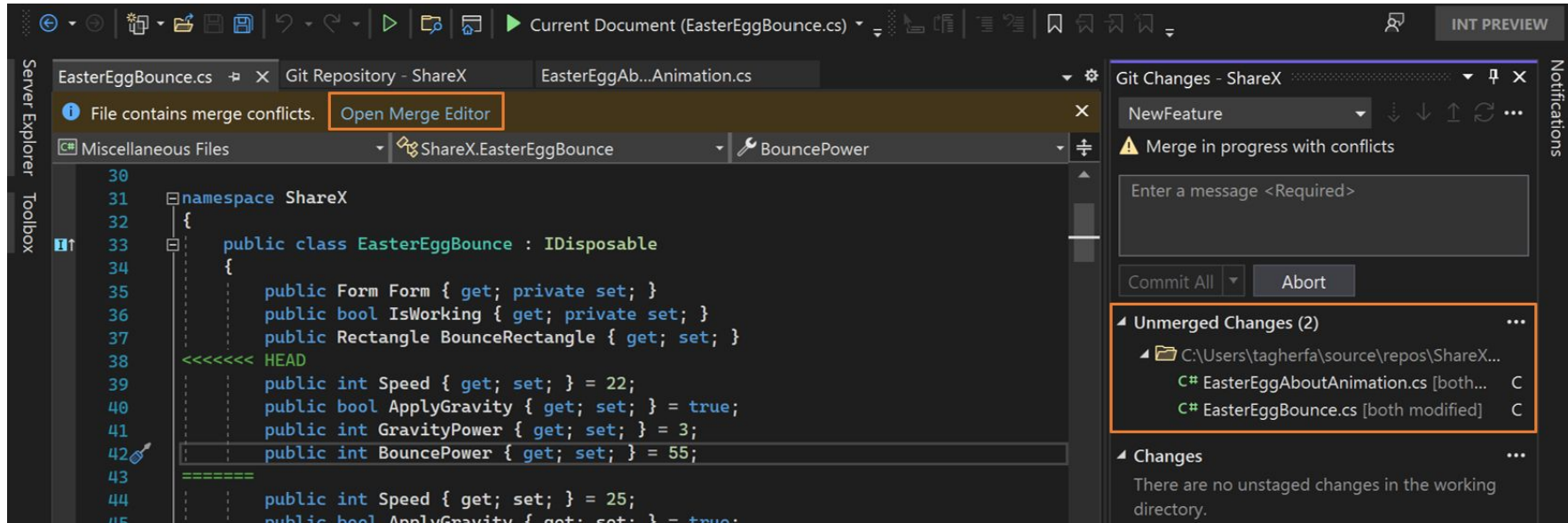
```
=====
```

seguito da :

```
>>>>>> nome-branch
```

Come risolvere un conflitto in Visual Studio Code

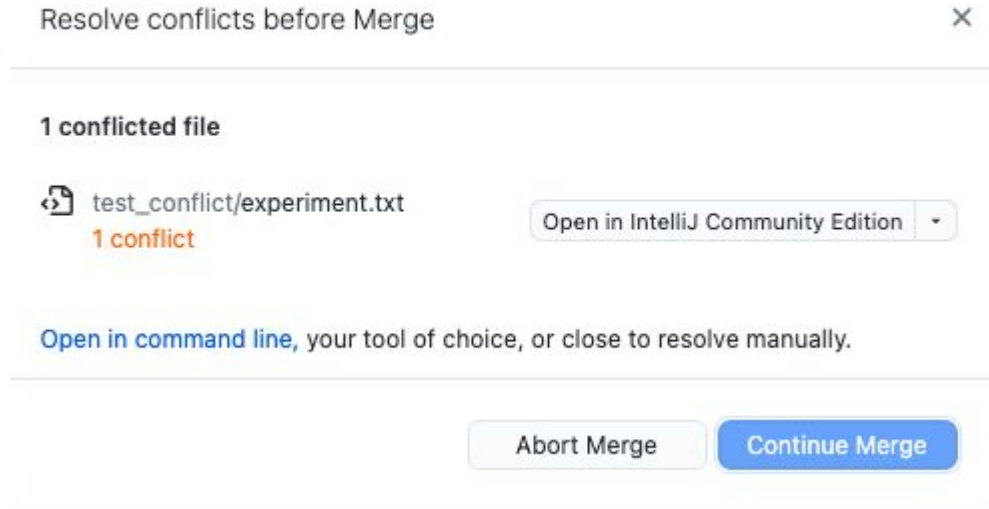
Per avviare la risoluzione dei conflitti, fare doppio clic su un file. In alternativa, se si dispone di un file con conflitti aperti nell'editor, è possibile selezionare **Apri editor di merge**.



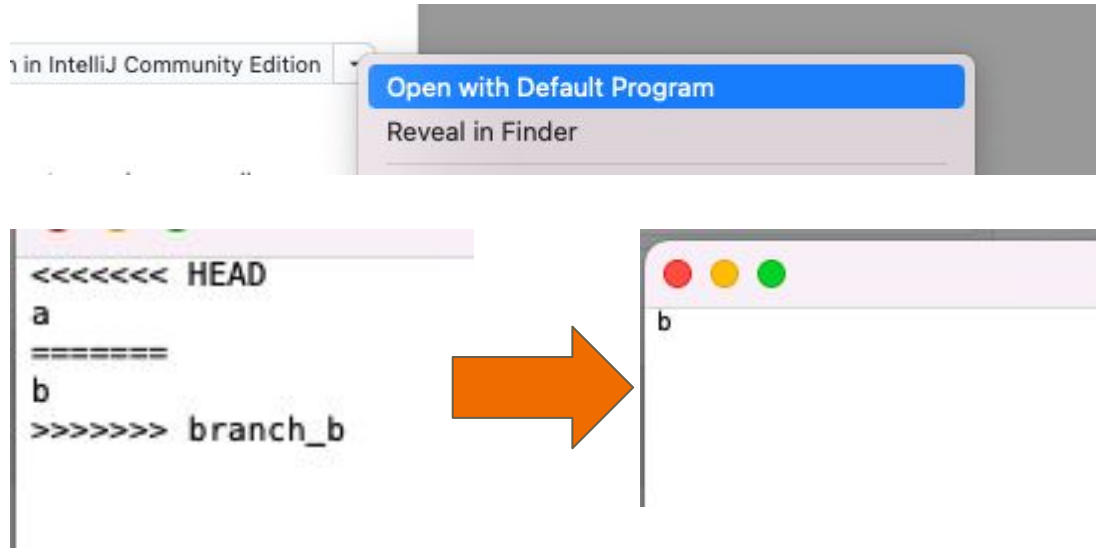
Esercizio! simulare un conflitto in Sourcetree

Per esempio: creiamo un branch in cui mettiamo un file, su questo branch costruiamo altri due branch che cambiano il file creato nel branch padre scrivendo due cose diverse sulla stessa riga.

Al merge la situazione sui due branch dovrebbe crearsi questa situazione:



Aggiustando manualmente il file incriminato con la modifica che si vuole tenere il git gui client ci dirà che il conflitto è stato risolto



Altri tipi di conflitti?

Ricordo che a volte ci sono conflitti strani e si ha fretta, come nel caso di questa lezione.

L'alternativa a sistemare i problemi è ri-clonare il progetto da zero.

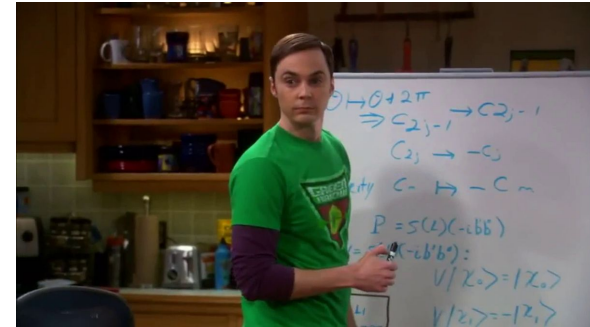
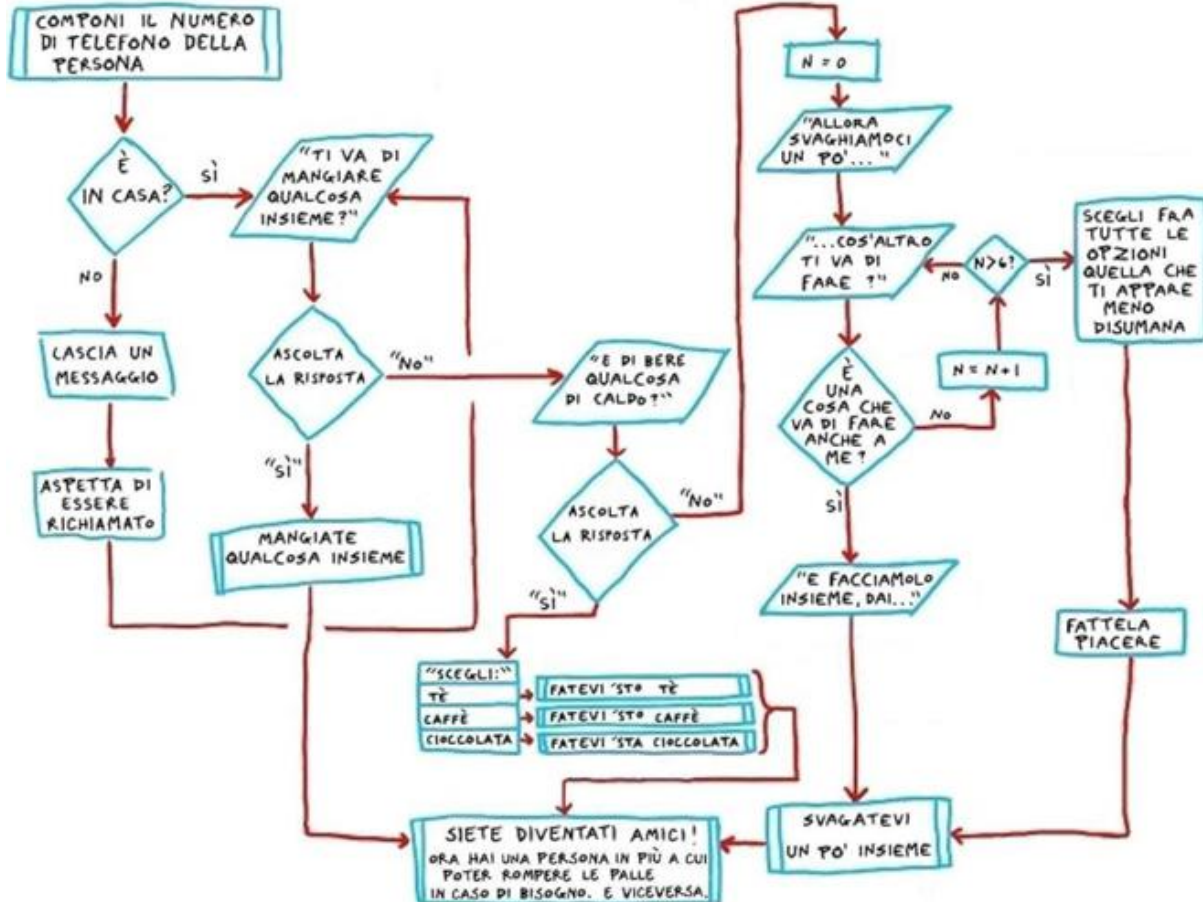
Questa opzione non è attuabile però in situazione lavorative quando bisogna unire due branch che sono da troppo tempo indipendenti.

In quel caso bisogna studiare bene una soluzione e dedicarci quindi il giusto tempo.



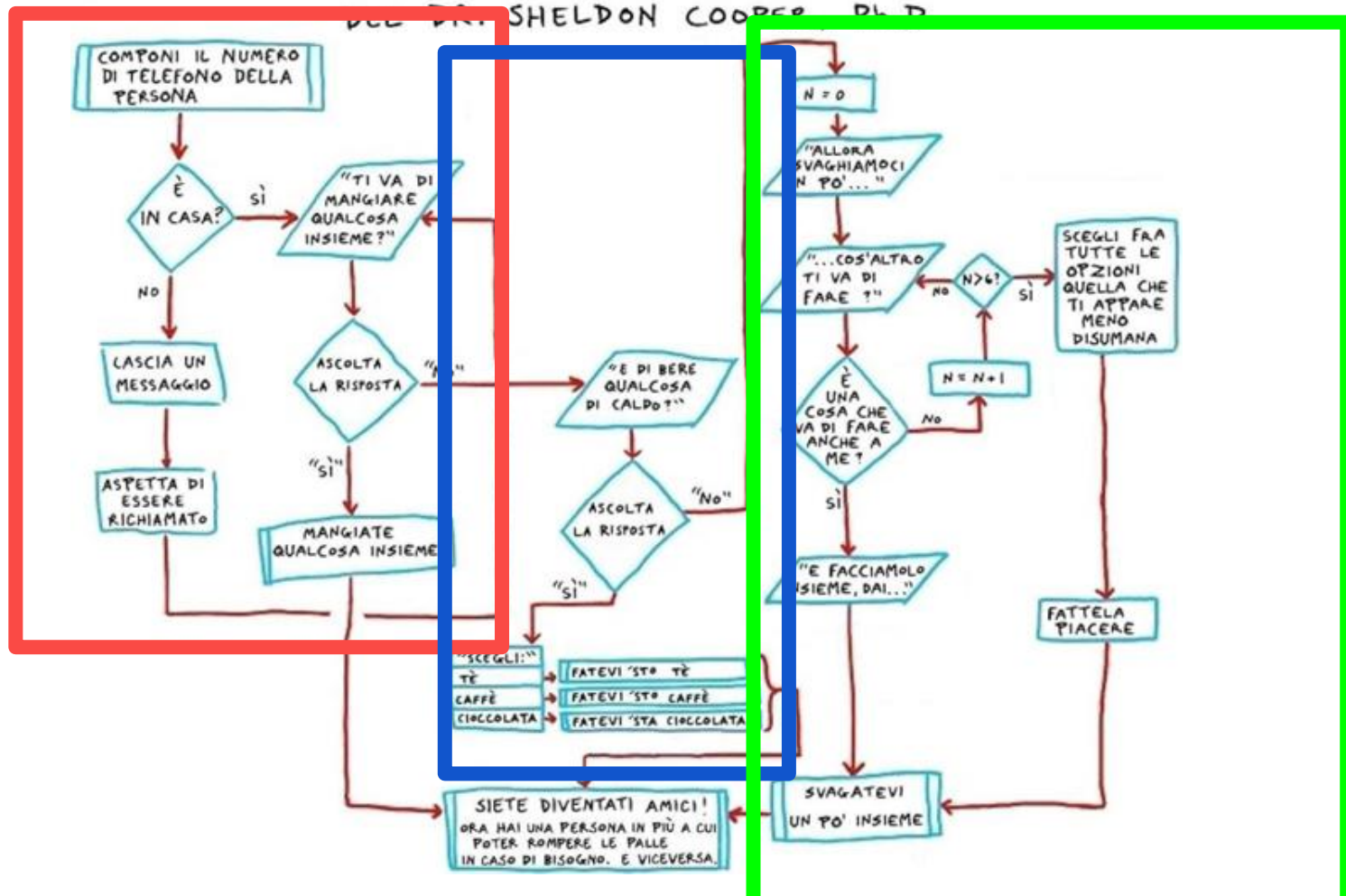
L'ALGORITMO DELL'AMICIZIA

DEL DR. SHELDON COOPER, Ph.D



L'ALGORITMO DELL'AMICIZIA

DEL DR. SHELDON COOPER, Ph.D.

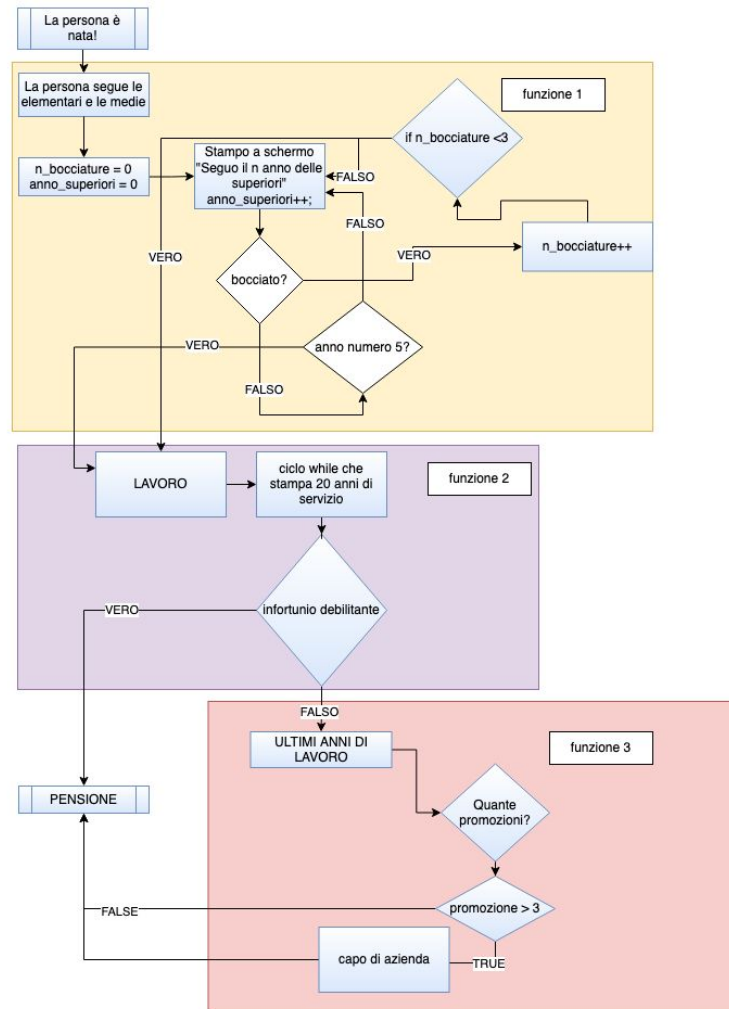


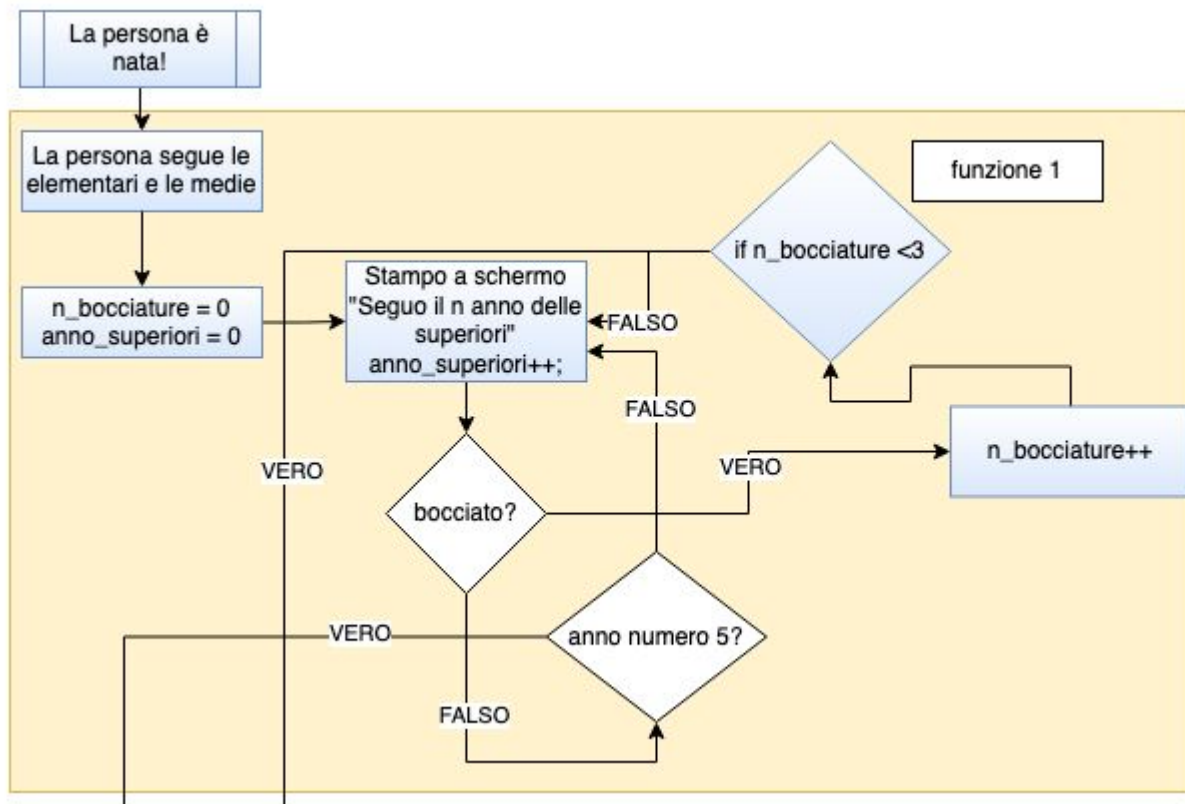
Esercizio su parte git

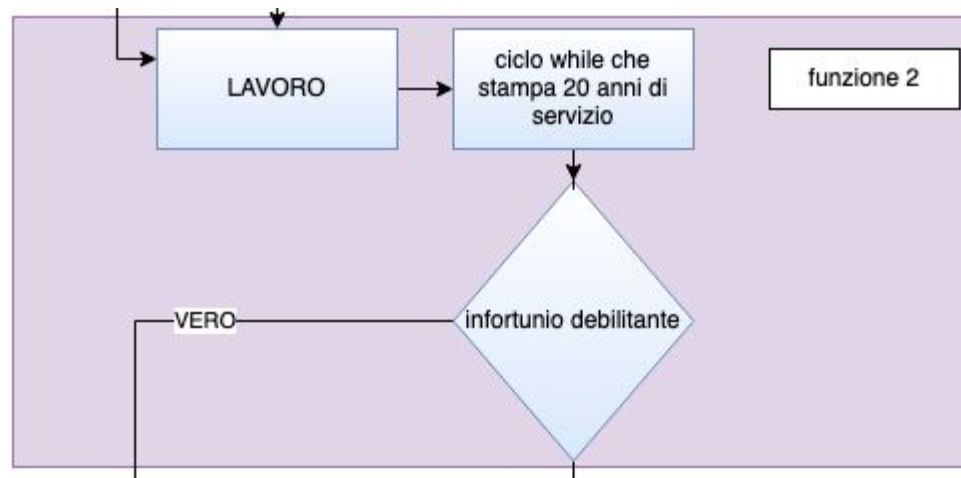
Ulteriore esercizio sull'uso dei branch

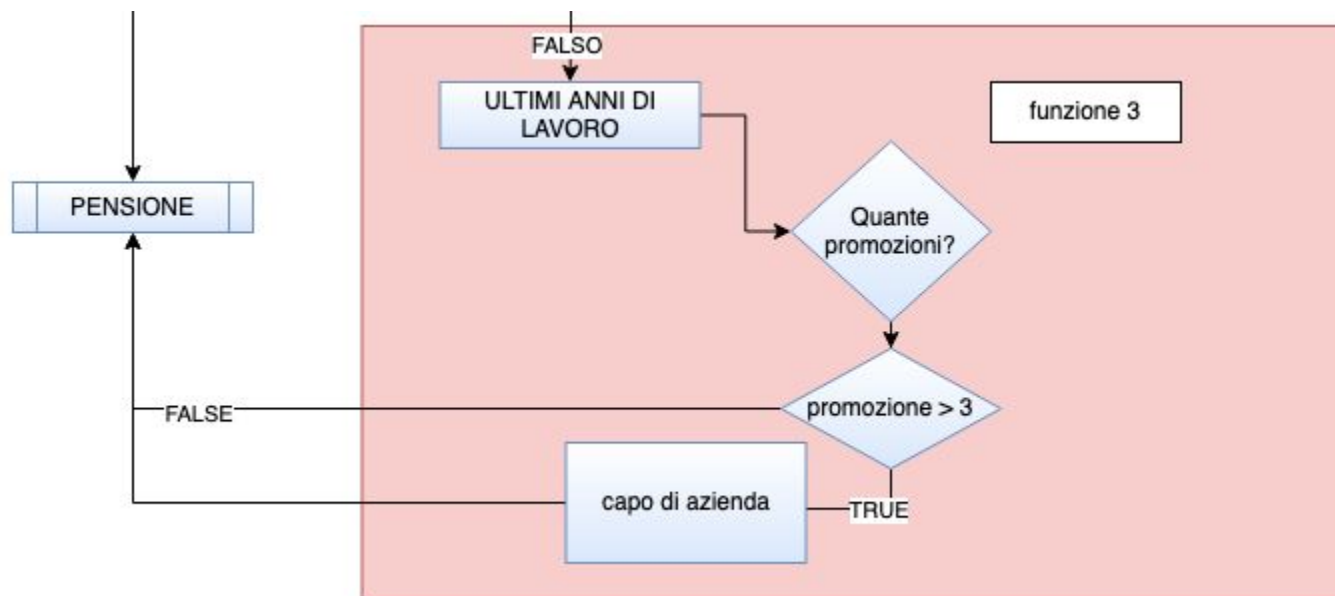
Algoritmo della vita











Benvenuti nel negozio di animali virtuali! In questo esercizio, dovrai creare un programma per gestire un negozio di animali virtuali. Ogni animale ha un nome e una lista di giocattoli preferiti. I giocatori possono interagire con gli animali offrendo loro dei giocattoli. Dovrai implementare tre funzioni che eseguano diverse operazioni sul negozio di animali.

Istruzioni:

1. Funzione "crea_animale":
 - La funzione "crea_animale" riceve come argomenti il nome dell'animale e una lista di giocattoli preferiti.
 - Deve creare un dizionario rappresentante l'animale con le seguenti chiavi:
 - "nome": il nome dell'animale (stringa).
 - "giocattoli_preferiti": una lista dei giocattoli preferiti dell'animale (lista di stringhe).
 - La funzione deve restituire il dizionario dell'animale creato.
2. Funzione "offri_giocattolo":
 - La funzione "offri_giocattolo" riceve come argomenti il dizionario dell'animale e un giocattolo offerto.
 - Deve aggiungere il giocattolo offerto alla lista dei giocattoli preferiti dell'animale.
 - La funzione non restituisce nulla.
3. Funzione "stampa_animali":
 - La funzione "stampa_animali" riceve come argomento una lista di dizionari rappresentanti gli animali del negozio.
 - Deve stampare il nome di ogni animale e la lista dei suoi giocattoli preferiti.
 - Utilizza un loop per iterare attraverso la lista di animali e stampare le informazioni corrispondenti a ciascun animale.

Note aggiuntive:

- Puoi utilizzare la funzione `print()` per stampare i risultati.
- Puoi testare le funzioni creando alcuni animali, offrendo loro dei giocattoli e stampando le informazioni degli animali.
- Assicurati di gestire correttamente i dati e di utilizzare i concetti di dizionari, liste e loop per completare le funzioni richieste.

Hai deciso di creare un programma che aiuti i tuoi amici ad organizzare una festa memorabile. Il programma utilizzerà tre funzioni distinte per ottenere le preferenze degli invitati, pianificare le attività della serata e creare una playlist perfetta.

****Specifiche****

1. La funzione ``get_party_preferences()`` chiede agli invitati di specificare le loro preferenze per la festa. Ad esempio, "Che tipo di musica vuoi ascoltare?", "Ti piace ballare?" e "Cosa vuoi mangiare?". La funzione restituisce le risposte degli invitati sotto forma di un dizionario.
2. La funzione ``plan_party(preferences)`` utilizza le preferenze degli invitati per pianificare le attività della serata. Ad esempio, se la maggior parte degli invitati ama ballare, la funzione potrebbe organizzare una sfida di ballo o una gara di karaoke. La funzione restituisce un elenco di attività pianificate.
3. La funzione ``create_playlist(activities)`` utilizza le attività pianificate dalla funzione precedente per creare una playlist perfetta. La playlist deve includere canzoni adatte ad ogni attività e deve essere memorizzata in un file di testo.
4. Il programma utilizza le tre funzioni descritte sopra per ottenere le preferenze degli invitati, pianificare le attività della serata e creare la playlist perfetta.

****Esempio di output****

...
Che tipo di musica vuoi ascoltare? Rock
Ti piace ballare? Sì
Cosa vuoi mangiare? Pizza

La tua festa perfetta include:

- Una sfida di ballo con la canzone "Livin' on a Prayer" dei Bon Jovi
- Un gioco di birra pong con la canzone "All Summer Long" di Kid Rock
- Una gara di karaoke con la canzone "Sweet Child O' Mine" dei Guns N' Roses
- Un'immane pizza party con la canzone "That's Amore" di Dean Martin

La tua playlist personalizzata è stata creata con successo e salvata su file.

...

****Note****

Ricorda di personalizzare le attività in base alle preferenze degli invitati e di utilizzare canzoni adatte ad ogni attività. Puoi anche aggiungere funzionalità aggiuntive, come la possibilità di creare una lista di drink e cocktail, o di invitare un DJ per la serata. Assicurati solo di avere abbastanza pizza per tutti gli invitati!