

Ministerul Educației și Cercetării al Republicii Moldova  
IP Colegiul “Iulia Hasdeu” din Cahul

Asistență pentru managementul proiectelor software

# **Tema 4: Crearea, schimbarea și unirea Branch-urilor**

Apareci Aurica  
Babaianu Savelie  
Grupa: AAW 2042

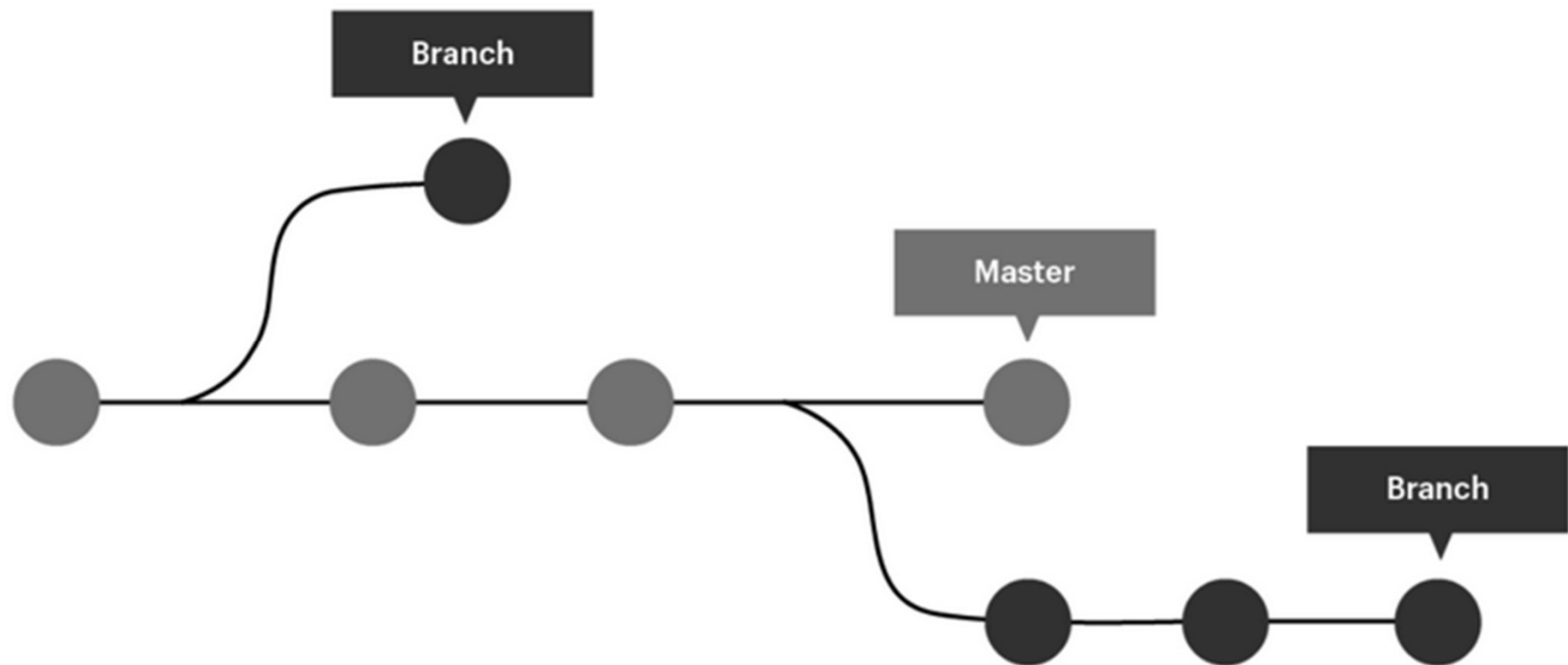


# **Noțiunea de BRANCH**

# Ce este un BRANCH ?

**Branch-urile** sunt linii de dezvoltare paralele care permit dezvoltatorilor să lucreze la diferite caracteristici ale proiectului în paralel, fără să interfere unul cu celălalt. De asemenea, ele sunt utile pentru organizarea și gestionarea dezvoltării într-un mod structurat prin separarea funcționalităților.

Crearea unui branch nou se face cu comanda ``git branch nume_branch``, unde ``nume_branch`` este numele noului branch. Branch-ul nou creat va fi o copie a branch-ului curent sau a branch-ului specificat în acel moment. Pentru a te schimba la un branch existent, poți utiliza comanda ``git checkout nume_branch``.



1

## Crearea și schimbarea branch-urilor

`git branch`

`git checkout new_branch`

`git checkout -b new_branch`

`git branch -d new_branch`

Rulează comanda `git branch -d <branch>`, pentru a șterge branch-ul curent. Dacă branch-ul poate fi șters în siguranță (adică nu conține modificări necomise), Git va șterge branch-ul și îți va afișa un mesaj de confirmare.

The background is a dark navy blue. On the left, there is a faint, light blue Git logo consisting of a diamond shape with a branching diagram inside. To the right of the logo, the word "git" is written in a large, lowercase, light blue sans-serif font, serving as a watermark. The main text is centered and reads "Unirea branch-urilor" in a bold, white, sans-serif font, with "(merge)" in a smaller, white, sans-serif font directly below it.

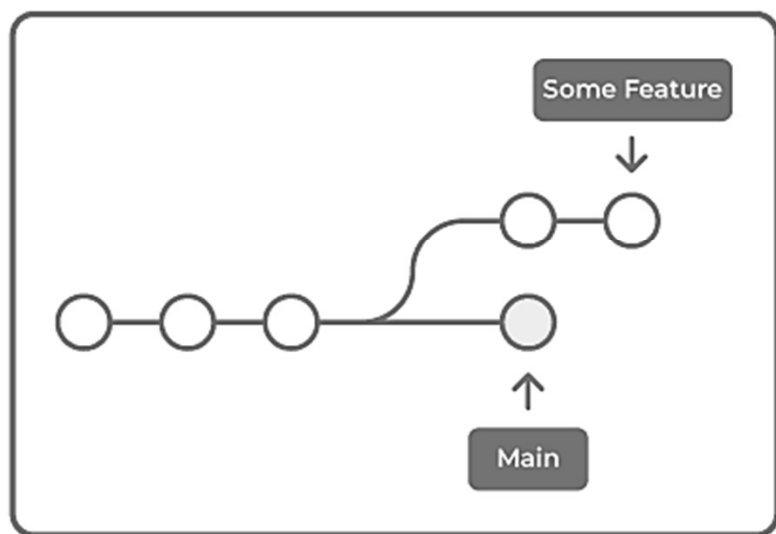
# **Unirea branch-urilor**

**(merge)**

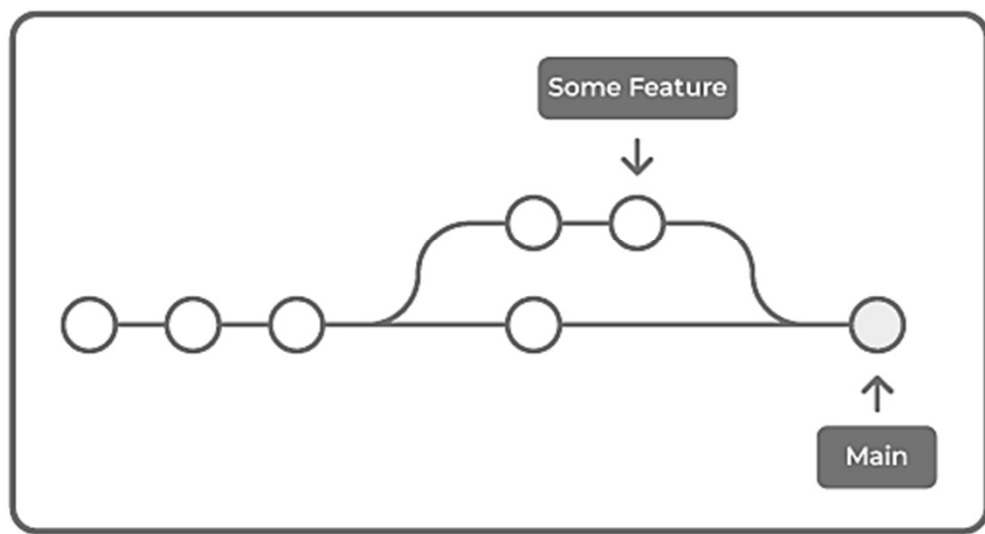
## Ce este **git merge** ?

Unirea branch-urilor (**merge**) în Git este procesul prin care modificările dintr-un branch sunt integrate în alt branch. Acest proces este utilizat pentru a combina dezvoltarea realizată pe branchuri separate, permițând astfel adăugarea noilor funcționalități sau corectarea problemelor în branchurile principale sau în alte branch-uri. Comanda utilizată pentru a realiza un merge este `git merge`, urmată de numele branch-ului din care dorești să faci merge. Merge-ul poate fi făcut într-un branch curent sau în alt branch specificat.

Before Merging



After a 3-way Merge





2

## Unirea branch-urilor

```
git merge new_branch
```

*new\_branch* reprezintă branch-ul din care dorești să faci merge în branch-ul curent

**git merge** comandă efectuează următoarele:

1

Identifică commit-ul comun cel mai recent dintre branch-ul curent și branch-ul specifica

2

Creează un nou commit care unifică schimbările din branch-ul specificat în branch-ul curent.

3

Actualizează curent pentru a reflecta noul commit de merge. referința branch-ului



În esență, git merge încorporează schimbările dintr-un branch în altul, astfel încât să poți lucra cu cea mai recentă versiune a codului. Dacă nu există conflicte, merge-ul se va efectua fără probleme.

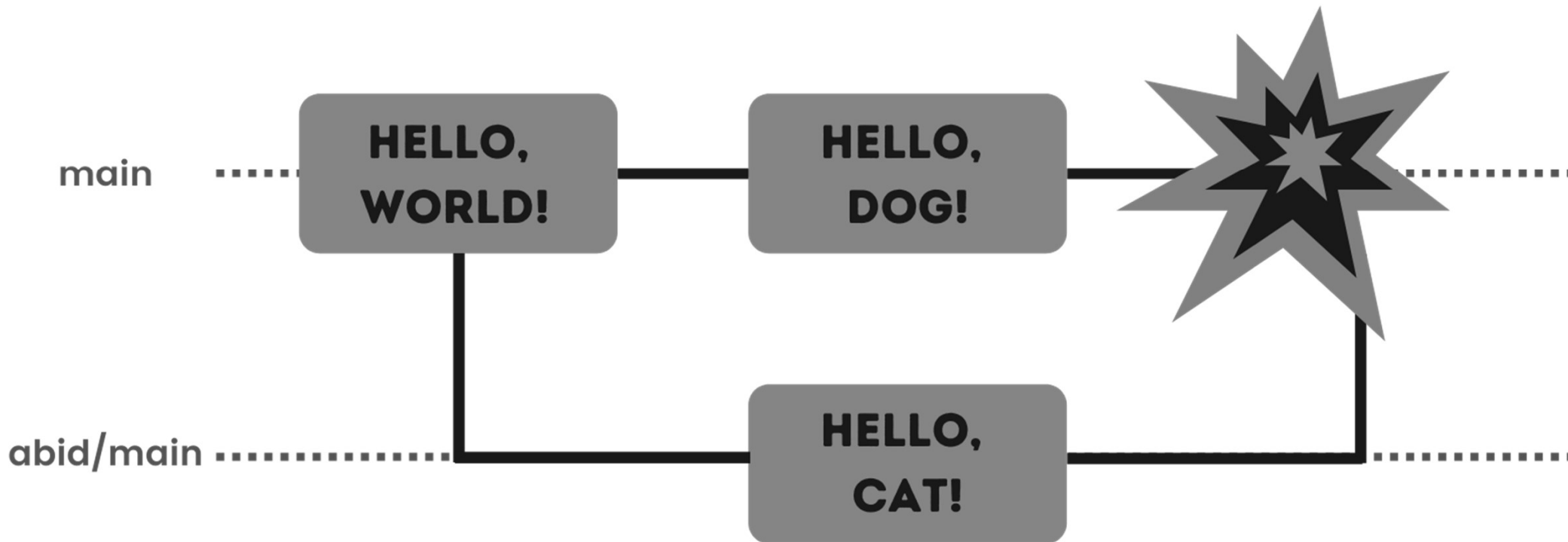


# **Rezolvarea conflictelor de unire (merge conflicts)**

## Ce înseamnă **merge conflicts**?

Conflictul de **unire (merge conflict)** apare atunci când două branch-uri sau două versiuni diferite ale aceluiași fișier au modificări incompatibile. Acest lucru se întâmplă în timpul procesului de unire (**merge**) atunci când Git nu poate decide automat cum să combine aceste modificări.

Pentru a rezolva conflictul de unire, dezvoltatorul trebuie să intervină manual și să decidă cum să combine aceste modificări incompatibile. Procesul implică analiza fișierelor cu conflicte, selectarea modificărilor corecte și apoi realizarea unui nou commit pentru a finaliza unirea.



## Exemplu **merge conflicts**

1 Ne imaginăm că două branch-uri separate, "branch-A" și "branch-B", au avut modificări în același fișier "fișier.txt". Când încerci să faci merge a branch-ului "branch-A" în "branch-B" sau invers, Git detectează un conflict de unire în "fișier.txt"

2 În acest moment, fișierul "fișier.txt" conține marcaje speciale care indică zonele conflictuale, cum ar fi "<<< HEAD" pentru branch-ul curent ("branch-B") și "<<< branch-A" pentru branch-ul "branch-A".

3 Dezvoltatorul trebuie să editeze manual "fișier.txt" și să aleagă care modificări să păstreze și care să elimine. Aceste decizii trebuie luate în funcție de cerințele proiectului.

## Exemplu **merge conflicts**

4

După ce modificările sunt făcute și conflictul este rezolvat, se adaugă fișierul la staging area cu ``git add fișier.txt`` și se face commit cu un mesaj care indică rezolvarea conflictului. Conflictul de unire a fost rezolvat cu succes și procesul de merge poate continua.

5

Rezolvarea conflictelor de unire este o parte importantă a gestionării versiunilor cu Git și necesită atenție și comunicare între membrii echipei pentru a asigura că modificările sunt integrate corect.