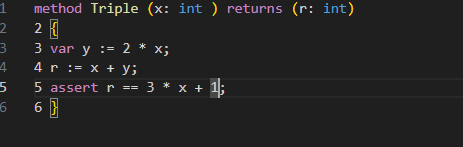
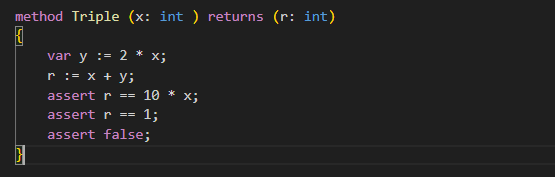
**Laborator 12**

**1.2**



Cu aserțiunea modificată `r == 3 \* x + 1`, metoda va eșua la aserțiune deoarece `r` (care este `3 \* x`) nu este niciodată egal cu `3 \* x + 1` pentru niciun număr întreg `x`.

**1.3**

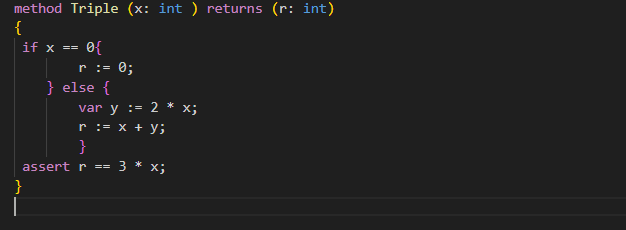


Verificatorul va raporta o eroare la Linia 5: `assert r == 10 \* x;`.

Verificatorul va raporta, de asemenea, o eroare la Linia 6: `assert r < 5;`.

Verificatorul va verifica apoi Linia 7: `assert false;`, care va eșua întotdeauna. Totuși, dacă verificatorul eșuează deja la afirmațiile anterioare, este posibil să nu continue până la această afirmație.

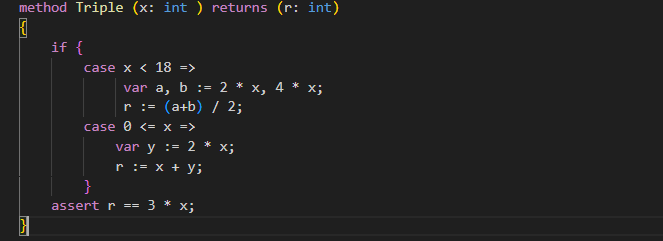
**1.4**



Pentru x = 0, r este 0, iar afirmația este adevărată.

Pentru orice x diferit de zero, metoda calculează r ca 3 \* x, iar afirmația este adevărată.

Comportamentul metodei este determinist deoarece, pentru orice valoare de intrare a lui x, rezultatul lui r este întotdeauna determinat în mod unic de logica din instrucțiunea if-else.

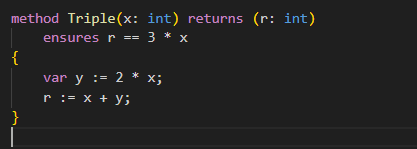


Pentru x < 18, metoda calculează r ca (2 \* x + 4 \* x) / 2, care se simplifică la 3 \* x.

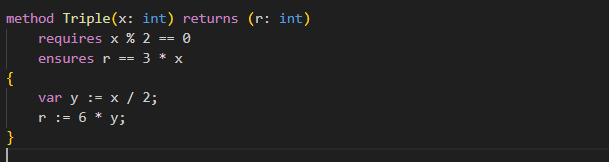
Pentru x >= 18, metoda calculează r ca x + 2 \* x, care este de asemenea 3 \* x.

Comportamentul metodei este determinist deoarece, pentru orice valoare de intrare a lui x, rezultatul lui r este întotdeauna determinat în mod unic de logica din instrucțiunea if.

**1.5**

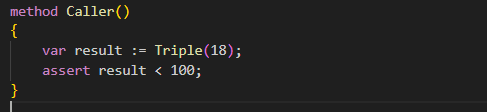


Verificatorul va reuși.Postcondiția `ensures r == 3 \* x` specifică faptul că după executarea metodei, valoarea returnată `r` ar trebui să fie de trei ori valoarea de intrare `x`. În această metodă, `r` este calculată de fapt ca `x + (2 \* x)`, care se simplifică la `3 \* x`. Prin urmare, postcondiția este satisfăcută.

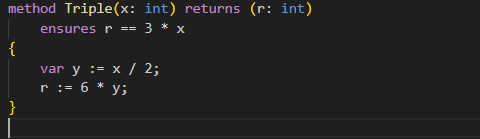


- Precondiție (requires x % 2 == 0): Această precondiție specifică faptul că intrarea x trebuie să fie un număr par. Asigură că metoda este apelată cu argumente valide.

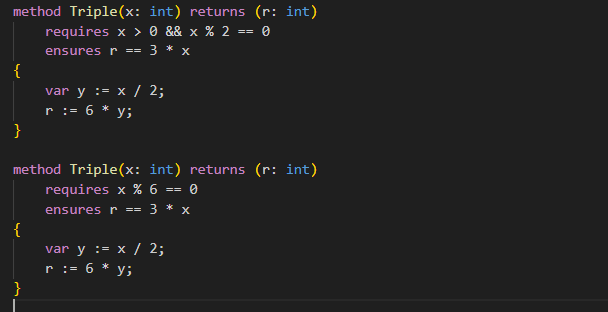
- Postcondiție (ensures r == 3 \* x): Această postcondiție specifică faptul că după executarea metodei, valoarea returnată r ar trebui să fie de trei ori valoarea de intrare x. În această metodă, r este calculată ca 6 \* (x / 2), care se simplifică la 3 \* x. Prin urmare, postcondiția este satisfăcută.



Această metodă apelează metoda Triple cu inputul x = 18. Conform celui de-al doilea exemplu al metodei Triple, precondiția require x % 2 == 0 nu este încălcată deoarece 18 % 2 == 0. Prin urmare, metoda Triple este executată cu succes și returnează o valoare de 54, satisfăcând postcondiția ensure r == 3 \* x. Afirmația assert result < 100 este de asemenea adevărată deoarece 54 < 100.



Eroarea de la verificatory.Fără precondiția `x % 2 == 0`, verificatorul va genera o eroare legată de o posibilă împărțire la zero. Deoarece împărțim `x` la 2 pentru a calcula `y`, operația de împărțire nu este sigură pentru toate valorile lui `x`.



Alternative pentru postconditia x%2 ==0.