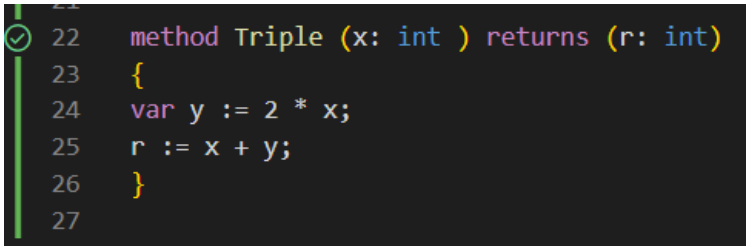
Student: Stănescu Andreea-Francesca, gr. 1, sg. 2

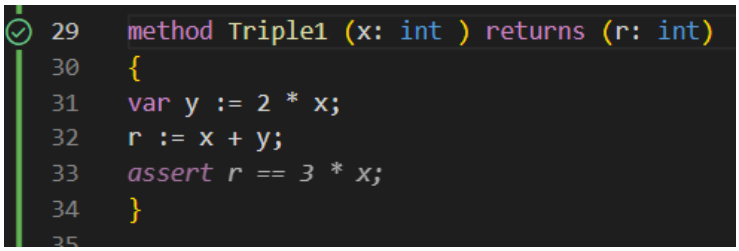
Laborator 12

1.1



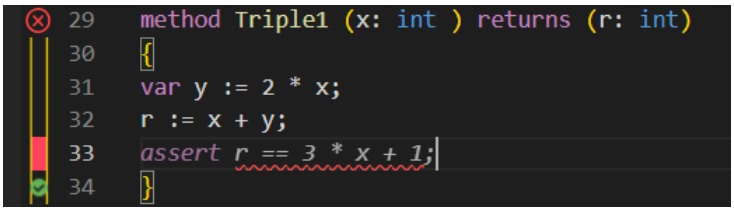
- x = 10 => y = 2 * 10 deci **y = 20,** iar r = x + y ⇔ r = 10 + 20 ⇔ **r = 30**
  Deci metoda asigneaza valoarea 30 variabilei r
- var t := Triple(18)
  x = 18 => y = 2 * 18 deci **y = 36,** iar r = x + y ⇔ r = 18 + 36 ⇔ **r = 54**
  Deci, in final, t are valoarea 54

1.2



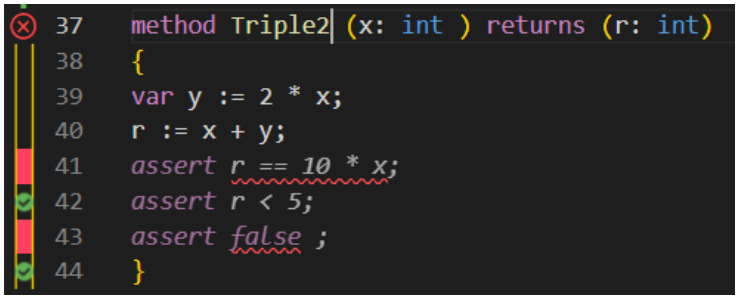
- Aceasta metoda va returna mereu triplul variabilei x, adica 3 * x. Se retine dublul lui x, 2 * x, in variabila y, iar apoi in variabila r adunam la dublul lui x, inca un x, deci obtinem 3 * x, astfel instructiunea *assert r == 3 * x* va fi mereu adevarata.

-



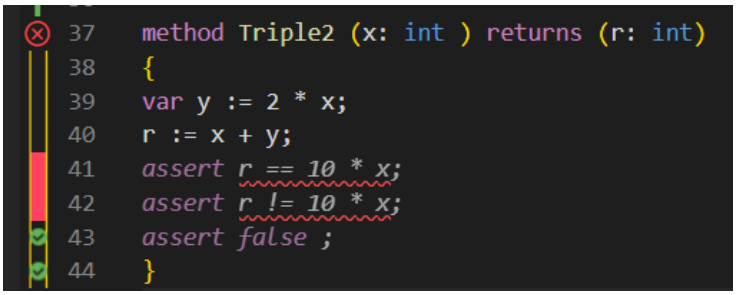
Schimbam asertiunea ca in imaginea de mai sus. Pentru ca aceasta asertiune sa fie adevarata, r ar trebui sa fie egal cu 3 * x + 1. Cu toate acestea, din calculele anterioare, stim ca r este de fapt 3 * x. Deci, 3 * x nu va fi niciodata egal cu 3 * x + 1 pentru nici o valoare intreaga x.

1.3

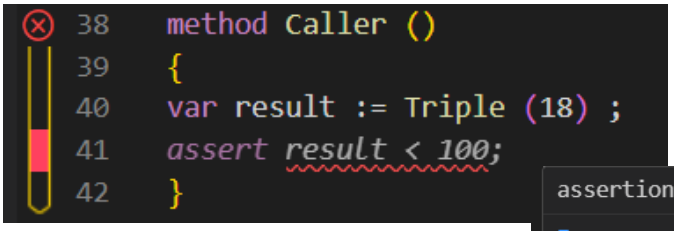


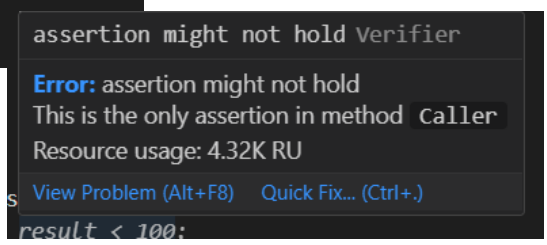
- in var y se va retine 2 * x, adica dublul lui x, iar in val r se retine x + 2 * x, adica 3 * x.
- asertiunea *assert r == 10 * x;* verifica daca r este egal cu 10 * x. Stim deja ca r = 3 * x, deci aceasta asertiune va esua, deoarece 3 * x nu este egal cu 10 * x pentru nicio valoare a lui x, cu exceptia cazului in care x = 0.
- asertiunea *assert r < 5;* verifica daca r este mai mic decat 5. Stim ca r = 3 * x. Asertiunea depinde de valoarea lui x (chiar daca x e o val mica). Daca x este diferit de 0, r va fi un multiplu de 3, deci nu garanteaza ca r < 5.
- asertiunea *assert false;* va esua intotdeauna, deoarece assert false este o conditie care nu poate fi niciodata adevarata.

- Pentru a face ca compilatorul sa se planga de primele doua asertiuni, dar nu de a treia, putem modifica a doua asertiune astfel incat sa fie contradictorie cu prima, dar compatibila cu *assert false*.



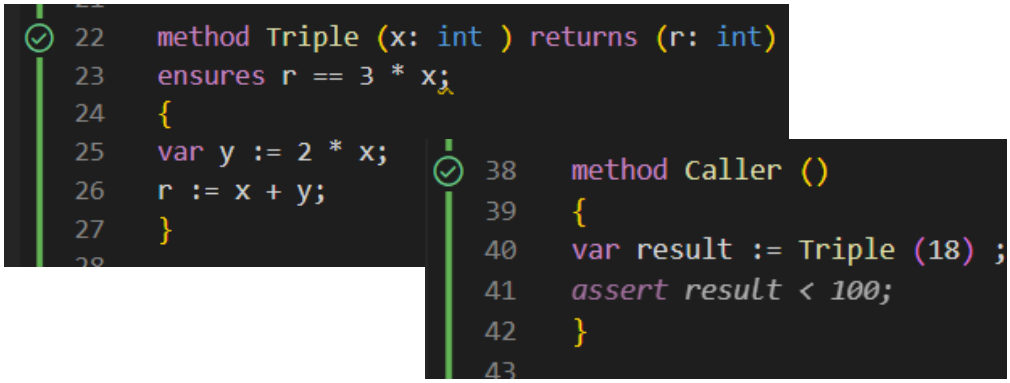
1.5



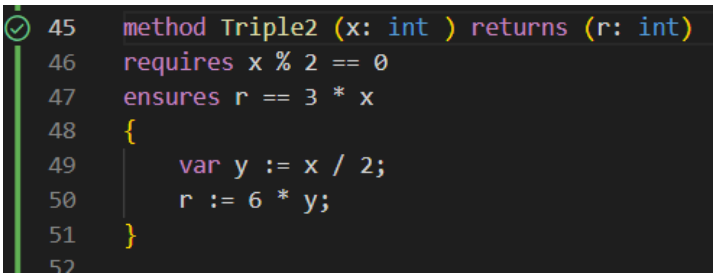


- Mesajul de eroare "asertion might not hold" indică faptul că verificatorul Dafny nu a putut demonstra că aserțiunea este adevărată. Asertiunea din metoda Caller, assert result < 100; verifica daca rezultatul returnat de metoda Triple (apelata cu argumentul 18) este mai mic decat

100. Deoarece asertiunea a generat o eroare, inseamna ca Dafny nu a putut garanta ca valoarea returnata de metoda Triple este intr-adevar mai mica decat 100, chiar daca ar trebui sa fie conform semanticii.

- 

Observam ca adaugand o clauza ensures, ne asiguram ca conditia specificata , r == 3 * x, este intotdeauna respectata la finalul executiei metodei si ca verificatorul Dafny poate valida aceasta conditie, eliminand astfel eroarea.



Preconditia (requires) specifica conditiile necesare pentru a apela o metoda, in timp ce postconditia (ensures) garanteaza ce trebuie sa fie adevarat la sfarsitul executiei acelei metode. Preconditia asigura intrarile corecte, in timp ce postconditia verifica rezultatul corectitudinii metodei.