Esercizi – Corso integrativo

# Esercizio 1 – Stack

Lo scopo dell’esercizio e’ di scrivere del codice che implementa uno Stack e di utilizzare Clousot per provarne la correttezza (parziale).

L’interfaccia della classe Stack e’:

public class Stack

{

private object[] arr;

private int nextFree;

public Stack(int len) { // … }

public bool IsEmpty { get { // … } }

public void Push(object x) { // … }

public object Pop() { // … }

}

## Semantica informale

Un oggetto di tipo stack contiene due campi: un array di oggetti che contiene i valori dello stack, ed un puntatore alla prossima posizione libera nello stack.

Inoltre, contiene un construttore, una proprieta’ e due metodi.

Il construttore inizializza un oggetto di tipo Stack, allocando un array di dimensione len.

IsEmpty ritorna vero se e solo se lo Stack e’ vuoto.

Push aggiunge un elemento in cima allo stack. Nel caso lo stack sia pieno rialloca gli elementi dell’array, utilizzando un array di dimensione almeno doppia.

Pop estrae l’elemento in cima allo stack e lo restituisce al chiamante.

## Parte 1.

* Terminare l’implementazione della classe Stack sopra, ed assicurarsi che compili senza errori
* Lanciare Clousot sull’implementazione
* Quali warning riporta lo static checker? Perche’? Giustificare tutti i warning (e.g. bug nel codice, imprecisione dell’analisi (perche’?) , mancanza di contratti …)

## Parte 2.

* Aggiungere i contratti e ricompilare
  + I contratti sono precondizioni, postcondizioni, ed invarianti di classe
* Lanciare Clousot sull’implementazione
* Clousot riporta zero warnings?
* In caso contrario,
  + spiegare il perche’
  + aggiungere altri contratti
  + o modificare il codice per arrivare a zero warnings

# Esercizio 2

public string TrimSuffix(string s, string suffix)

{

Contract.Requires(s != null);

Contract.Requires(suffix != null);

string res = s;

while (res.EndsWith(suffix))

{

int len = res.Length - suffix.Length;

res = res.Substring(0, len);

}

return res;

}

* Qual e’ la variante del ciclo?
  + (Una variante *v* e’ una funzione positiva che decresce strettamente ad ogni iterazione)
* Modificare il codice aggiungendo un’asserzione che menzioni il fatto che la variante di ciclo decresce strettamente
* Clousot puo’ provare l’asserzione? Perche’? Qual e’ il bug nel codice?

# Clousot

Puo’ essere scaricato

<http://msdn.microsoft.com/en-us/devlabs/dd491992.aspx>

# Email:

[logozzo@microsoft.com](mailto:logozzo@microsoft.com)