

POLITECHNIKA CZĘSTOCHOWSKA

TESTOWANIE OPROGRAMOWANIA

Laboratorium 2

Autor:

Piotr FILEK
Mateusz STALA
II grupa IO

Prowadzący:

dr inż. Grzegorz MICHALSKI

15 października 2014

1 Cel laboratorium

Celem laboratorium było zaprojektowanie przypadków testowych dla programu, który miałby za zadanie testować działanie programu sprawdzającego typ trójkąta (równoboczny, równoramienny, różboczony) oraz jego odporność na błędy. Drugim zadaniem było napisanie programu, który by przechodził wszystkie wcześniej zaprojektowane przypadki testowe.

2 Przebieg laboratorium

2.1 Działanie programu

```
./a.out  
Podaj bok a: 3  
Podaj bok b: 3  
Podaj bok c: 3
```

2.2 Przypadki testowe

2.2.1 Samodzielnie zaprojektowane

1. Czy boki tworzą trójkąt (2,2,2)
2. Czy boki nie tworzą trójkąta (1,2,3)
3. Czy program obsługuje liczby rzeczywiste (5.2, 5.2, 5.2)
4. Czy program odrzuca liczby ujemne (-31, -1, -2)
5. Czy program odrzuca zero (0, 5, 3)
6. Czy program odrzuca nieprawidłowe wejście (a, d, c)
7. Czy program rozpoznaje (5,5,5) jako trójkąt równoboczny
8. Czy program rozpoznaje (5,5,6) jako trójkąt równoramienny
9. Czy program rozpoznaje (4,5,6) jako trójkąt różnoramienny
10. Czy program nie rozpoznaje trójkątu równobocznego (5,5,5) jako równoramiennego

2.2.2 Propozycje testów od prowadzącego

1. Czy program czeka na wejście w przypadku wciśnięcia Enter
2. Czy program nie wykrywa (0,0,0) jako równobocznego
3. Czy program nie wykrywa (0,0,1) jako równoramiennego
4. Permutacje wszelkich wprowadzanych danych

2.3 Kod Programu

Autorzy: Piotr Filek, Mateusz Stala

```
1 #include <iostream>
2 #include "math.h"
3
4 using namespace std;
5
6 bool podajTrojkat(double &a, double &b, double &c){
7     cout << "Podaj bok a: ";
8     cin >> a;
9     if(!cin)
10     {
11         cout << "\nBłąd, nieprawidłowe wejście.\n";
12         return false;
13     }
14     cout << "Podaj bok b: ";
15     cin >> b;
16     if(!cin)
17     {
18         cout << "\nBłąd, nieprawidłowe wejście.\n";
19         return false;
20     }
21     cout << "Podaj bok c: ";
22     cin >> c;
23     if(!cin)
24     {
25         cout << "\nBłąd, nieprawidłowe wejście.\n";
26         return false;
27     }
28     return true;
29 }
30
31 bool czyTworzaTrojkat(double a, double b, double c)
32 {
33     if( (a + b > c) && (b + c > a) && (c + a > b) )
34         return true;
35     else
36         return false;
37 }
38
39 bool czyDodatnie(double a, double b, double c)
40 {
41     if( (a>0) && (b>0) && (c>0) )
42         return true;
43     else
44         return false;
45 }
46 void jakiTrojkat(double a, double b, double c)
47 {
48     if ( a == b && b == c )
49         cout << "Trójkąt równoboczny\n";
50     else if ( a == b || b == c || a == c )
51         cout << "Trójkąt równoramienny\n";
52     else
53         cout << "Trójkąt różnoboczny\n";
54 }
```

```

55 }
56
57 int main(int argc, char *argv[])
58 {
59     double a=0,b=0,c=0;
60     if(!podajTrojkat(a,b,c))
61         return 0;
62     else
63     {
64         if(!czyDodatnie(a, b, c))
65         { cout << "Wszystkie boki muszą być większe od zera!\n"; return
0;}
66         if(!czyTworzaTrojkat(a,b,c))
67         {
68             cout << "Boki nie tworzą trójkąta\n";
69             return 0;
70         }
71         jakiTrojkat(a,b,c);
72         return 0;
73     }
74 }

```

trojkat.cpp

3 Wnioski

W przeciwieństwie do pierwszych zajęć, przypadki testowane były planowane przed napisaniem programu - zasymulowanie efektu testowania programu, którego nie pisaliśmy/nie posiadamy kodu źródłowego. Pomimo prowadzącego zajęcia sugerującego istnienie co najmniej 15 przypadków testowych, nie udało się nam ich tyle odnaleźć. Prowadzący na koniec zajęć przeczytał zaproponowane przez siebie przypadki testowe, z których nie wszystkie znajdowały się na naszej liście. Utwierdza to w przekonaniu, że programista zawsze potrzebuje testera, gdyż bez tego nie wszystkie rzeczy zostaną przetestowane, a te krótko zaprojektowaliśmy - zawsze zakończą się sukcesem.