**Sprawozdanie - konstrukcje sterujące**

Konstrukcje sterujące pozwalają użytkownikowi na wybranie które instrukcje mają zostać wykonane w programie w zależności od wartości wpisanego w warunek zdania logicznego(ZL). Konstrukcja if(ZL){instrukcja1} else{instrukcja2} działa w sposób następujący: jeżeli ZL == true to wykonaj instrukcję1, w przeciwnym przypadku instrukcje2. Konstrukcja if(ZL){instrukcja1} else if(ZL2){instrukcja2} … else{} działa w analogiczny sposób. Konstrukcja switch(zmienna) wykonuje dane instrukcje w zależności od wartości danej zmiennej, zapis: switch(zmienna){case x: instrukcja1; break; case y: instrukcja2; break; default: instrukcja3; break;}. Kompilator będzie pomijał instrukcje aż do momentu natrafienia na „case” z wartością równą zmiennej, następnie będzie wykonywać instrukcje aż do napotkania komendy break;. W przypadku pominięcia wszystkich „case-ów” zostaną wykonane instrukcje pod „default”.

Badanie przy jakiej wartości zmienne pojedynczej i podwójnej precyzji tracą swoją dokładność:

**double**   **float**

dla a = 10^-i c = 1 b = 2, i = {5,6,7,8,9,10,11,12,13,14,15}

Δ = b^2 – 4\*a\*c = 4 – 4\*a\*1

|  |  |  |
| --- | --- | --- |
| i=5 -> Δ = 3.999960000000000 | i = 5 -> Δ = 3.999959945678711 | |
| i=6 -> Δ = 3.999996000000000 | i = 6 -> Δ = 3.999995946884155 | |
| i=7 -> Δ = 3.999999600000000 | i = 7 -> Δ = 3.999999523162842 | |
| i=8 -> Δ = 3.999999960000000 | i = 8 -> Δ = 4.000000000000000 | |
| i=9 -> Δ = 3.999999996000000 | i = 9 -> Δ = 4.000000000000000 | |
| i=10 -> Δ = 3.999999999600000 | i = 10 -> Δ = 4.000000000000000 | |
| i=11 -> Δ = 3.999999999960000 | i = 11 -> Δ = 4.000000000000000 | |
| i=12 -> Δ = 3.999999999996000 | i = 12 -> Δ = 4.000000000000000 | |
| i=13 -> Δ = 3.999999999999600 | i = 13 -> Δ = 4.000000000000000 | |
| i=14 -> Δ = 3.999999999999960 | i = 14 -> Δ = 4.000000000000000 | |
| i=15 -> Δ = 3.999999999999996 | i = 15 -> Δ = 4.000000000000000 | |
| i=16 -> Δ = 4.000000000000000  Δ równa się 4 wtedy i tylko wtedy gdy a == 0. Z powyższych danych można wywnioskować iż double traci swoją dokładność gdy nadamy mu wartość mniejszą niż 10e-16, dla typu float ta wartość wynosi 10e-8. Zatem wartość zdefiniowanej zmiennej TOLERANCJA należy uzależnić od powyższych wyników. Jeżeli jakikolwiek wprowadzona wartość będzie mniejsza od tolerancji:  if(fabs(a) < TOLERANCJA || fabs(b) < TOLERANCJA || fabs(c) < TOLERANCJA)  należy przerwać program: exit(-1); lub poprosić użytkownika o ponowne wprowadzenie danych.  Dzięki temu jesteśmy pewni o dokładności delty i możemy zastosować 3 warunki w celu obliczenia pierwiastków równania: (Δ > 0), (Δ == 0), (Δ < 0). | |