#(Instruckcja poc_lab4)

Zadanie 1

Sprawdź działanie obsługi wyjątków/błędów (exception handling) w języku C++.

- Utwórz klasę abstrakcyjną `Except` która ma metodę wirtualną do drukowania informacji o wyjątku (np. `PrintInfo()`)
- Utwórz klasę konkretną np. `Except1` która będzie posiadała implementację metody `PrintInfo()`
- W funkcji main umieść instrukcje służące do przechwytywania wyjątków:

```
try {
    ....
}
catch ( Except& e) {
    e.PrintInfo();
}
```

- W sekcji try wywołaj jakąś funkcję która rzuca wyjątek `Except1`:

```
throw Except1;
```

- Spróbuj zrobić to samo dla nowej klasy `Except2` która będzie drukowała informacje w której linii kodu został rzucony wyjątek (użyj zmiennej preprocesora `__LINE__`)
- Sprawdź działanie sekcji `catch(...)` do przechwytywania wszystkich wyjątków. Dopisz ją poniżej już istniejącej sekcji `catch` i spróbuj rzucić wyjątek który nie jest typu `Except`.

Zadanie 2

Wykorzystanie prostych wzorców (templatów)

- Napisz wzorce (template'y) funkcji `mymin` i `mymax` liczące odpowiednio minimum i maksimum z dwóch argumentów. Sprawdź działanie tych funkcji dla różnych typów np. `int`, `double`.
- Co jest konieczne aby można było wykorzystać powyższe funkcje również do klasy np. `Wektor2D`?
- Przerób klasę `Wector2D` tak aby był to wzorzec klasy sparametryzowany typem składowych wektora:

```
template< class T >
Wector2D
{
    . . .
};
```

- Sprawdź działanie takiej klasy.
- Przerób klasę `Wector2D` tak aby była ona również sparametryzowana ilością składowych:

```
template< class T, int N >
Wector
{
    . . .
};
```

- Sprawdź działanie takiej klasy.