

1. Stwórz nowy projekt Wyjątki.

Spróbuj wykonać następujący kod:

```
int liczba1;  
liczba1=30 / 0;
```

2. Jaki błąd wyświetlił kompilator? Czy da się go oszukać? (podpowiedź użyj 3 zmiennych)

3. Operacje na zmiennych (np. `liczba1 = liczba2 / liczba3`) umieść w bloku `try`.

4. Stwórz blok `catch` i dodaj tam komunikat np. "Nie lubię dzielić przez zero".

5. Stwórz blok `finally` i umieść w nim komunikat "FINALLY".

6. Uruchom program dwukrotnie, modyfikując zmienną 3 na liczbę inną niż 0 i 0. Widzisz różnicę między `catch` i `finally`?

7. Edytuj blok `catch` w następujący sposób: `catch (Exception e)`, a wyświetlany komunikat zamień na `e.Message`.

8. Wywołaj program dzieląc przez zero. Sprawdź co jeszcze można wywołać ze zmiennej `e` oprócz polecenia `Message`.

9. Stwórz tablicę `int[] table = new int [2]`, a następnie w bloku `try` umieść kod: `table[25] = 2`. Jak widać wychodzimy poza zakres. Możemy stworzyć kilka wyjątków tzw. selektywna obsługa wyjątków.

10. Stwórz 2 bloki `catch`

```
catch(IndexOutOfRangeException e)  
catch()
```

W obu umieść różne komunikaty.

Uruchom program, zwróć uwagę który kod się wywołał.

11. Ciekawą opcją dla programisty, jest wywołanie wyjątku - konstrukcja jest prosta. W bloku `try` umieść następujący kod `throw new IndexOutOfRangeException();` następnie użyj bloku `catch` aby przechwycić wyjątek.

12. Sprawdź w Internecie czego dotyczą następujące wyjątki:

StackOverflowException

NullReferenceException

FileNotFoundException

AccessViolationException

IndexOutOfRangeException

Sprawdź jak dużo różnych wyjątków [https://msdn.microsoft.com/pl-pl/library/system.exception\(v=vs.110\).aspx](https://msdn.microsoft.com/pl-pl/library/system.exception(v=vs.110).aspx)

13. Stwórz własną klasę wyjątku. Teoretycznie wystarczy stworzyć klasę dziedziczącą po `System.Exception`, stwórz publiczną klasę `NaszWyjatekException`.

Przydałoby się zdefiniować przynajmniej jeden użyteczny konstruktor, a także doimplementować domyślny, stwórz konstruktor

```
NaszWyjatekException(string message) : base(message)
```

oraz:

```
NaszWyjatekException(String message, Exception  
innerException)
```

```
    : base(message, innerException)
```

14. Stwórz konsolowy kalkulator (minimalne operacje to: dodawanie, odejmowanie, mnożenie, dzielenie, pierwiastek kwadratowy). Zadbaj o właściwą obsługę wyjątków.

15. Stwórz klasę `Osoba` z polami `imię` i `nrPesel`. Następnie zaplanuj program z obsługą wyjątków, który będzie od użytkownika pobierał imię i numer PESEL, a jako wyjątek wyrzucał informację kiedy PESEL nie jest poprawny.

O zasadach PESEL możesz poczytać tu <https://pl.wikipedia.org/wiki/PESEL>

Jeśli potrzebujesz generatora numerów PESEL, skorzystaj z

<http://www.bogus.ovh.org/generatory/all.html> lub wyszukaj inny w sieci.