

1. Stwórz nowy projekt. Spróbuj wykonać następujący kod:

```
int liczba1;  
liczba1=2 / 0;
```

Jaki błąd wyświetlił kompilator? Czy da się go oszukać? Zamień kod na:

```
int liczba1;  
int liczba2 = 2;  
int liczba3 = 0;  
liczba1 = liczba2 / liczba3;
```

2. Operacje na zmiennych (np. `liczba1 = liczba2 / liczba3`) umieść w bloku try. Stwórz blok catch i dodaj tam komunikat np. "Nie dziel przez zero!".

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

6. Uruchom program dwukrotnie, modyfikując zmienną3 na liczbę inną niż 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. Przetestuj pozostałe właściwości: `Data`, `HelpLink`, `Source`, `TargetSite`.

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

```
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.

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

Przydałoby się zdefiniować przynajmniej jeden użyteczny konstruktor poza domyślnym:

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

Przetestuj wywoływanie własnych wyjątków.

14. Stwórz nowy projekt. Dodaj w nim następujący kod:

```
int n = int.MaxValue;
```

```
n = n + 1;
```

```
Console.WriteLine(n);
```

```
Console.ReadKey();
```

Przeanalizuj kod, skompiluj i uruchom aplikację. Czemu tak się dzieje?

Dodaj powyższy kod w block checked i unchecked.

15. Przeanalizuj dwa poniższe kody jak bezpiecznie odczytywać zawartość plików tekstowych:

<https://github.com/Piotrek85/WyjatkiOdczytPliku>

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

- dzielenie przez zero

- wpisanie błędnego łańcucha na wejściu (kropki, przecinki, litery)

- wyjście poza zakres

- liczenie pierwiastka kwadratowego z liczby ujemnej

a) Przy okazji sprawdź czy znaczenie ma użycie typu `double`, `float` czy `decimal`?

b) przeanalizuj różnice w instrukcjach wejścia `Console.ReadLine`, `Console.Read`, `Console.ReadKey`

c) czy ma znaczenie użycie konwertowanie, parsowania? a może lepiej użyć TryParse?

1. 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.