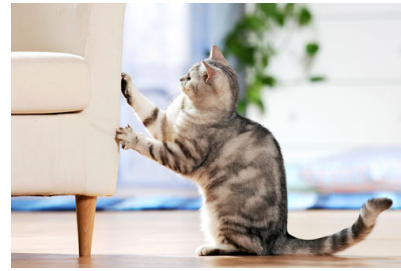


## Nie będę drapać mebli!



```
#pętla #while
#zwiększanie_o_1
```

Niesforne pociechy karało się dawniej nakazując im wielokrotne przepisanie tego samego umoralniającego zdania, na przykład: *Nie będę podkradał czekolady z kredensu.*

W czasach, gdy komputery są wszechobecne, sprytny psotnik mógłby posłużyć się do tego *instrukcją pętli*, która umożliwia wielokrotne powtarzanie tej samej operacji. W języku C++ istnieje kilka rodzajów pętli, a najprostszą z nich jest pętla `while` (wymawiaj: *łajl*). Składnia tej instrukcji jest następująca:

```
while(warunek)
{
    treść pętli
}
```

Jeśli *treść pętli* ogranicza się do pojedynczej instrukcji (być może złożonej), wtedy nawiasy klamrowe `{}` można pominąć.

Jako *warunek* wstawiamy wyrażenie (zdanie) logiczne. Oto przykład niekończącej się pętli:

```
while(true)
    cout << "Motyla noga, chyba się zapętliłem!\n";
```

Zazwyczaj jednak będziemy używać pętli działającej określoną ilość razy. W tym celu przyda się nam zmienna pełniąca rolę licznika (ang. *counter*, wymawiaj: *kałnter*) obiegów pętli (zwiększającego o jeden po każdym obiegu). Tak więc gdzieś na początku programu umieścimy deklarację zmiennej `counter` o wartości początkowej 0:

```
int counter{ };
```

Nawiasy klamrowe `{ }` po nazwie w deklaracji służą do *inicjalizacji* zmiennej, czyli nadania jej początkowej wartości. Zazwyczaj sama deklaracja nie gwarantuje, że zmienna będzie miała jakąś określoną wartość – w tym przypadku może to być przypadkowa liczba całkowita. Puste nawiasy oznaczają nadanie wartości równej zero.

W treści pętli wstawimy powiększanie licznika o jeden:

```
counter++;
```

Instrukcja powiększania o jeden jest często stosowana, stąd w języku C++ występuje specjalny operator służący do tego celu: `++`. Analogiczny operator `--` służy do zmniejszania wartości zmiennej.

Nazwy zmiennych można pisać po polsku (byle bez polskich znaków diakrytycznych **ą**, **ę** itp.), ale proponujemy trzymać się jednak języka angielskiego.

Ostatecznie nasz program (odpowiadający 10-krotnemu wypisaniu karnego tekstu przez naszego kota, który zniszczył skórzaną kanapę) powinien wyglądać mniej więcej tak:

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int counter{ };
    while(counter < 10)
    {
        cout << "Nie będę drapać mebli!\n";
        counter++;
    }
    return 0;
}
```

Po uruchomieniu zobaczymy taki wydruk:

```
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
Nie będę drapać mebli!
```

Po takiej karze nasz kotek na pewno będzie już grzeczny!