Struktury cykliczne #7

W jaki sposób sterować programem. Można za pomocą struktur cyklicznych, czyli pętli.

a. Pętla while
b. Pętla for
c. Zapętlamy program

Wielokrotne wykonanie zadania

```
while (warunek) {
    instrukcja1
    instrukcja2
    ...
}
```

kolejne iteracje (wykonania zawartości pętli) dopóki warunek będzie spełniony (nawet w nieskończoność - lepiej uważać ;))

Nieskończone wykonanie pętli

```
while (true) {
    console.log(“ooo ooo, chyba zawiesiłem komputer”)
}
```

Może obędzie się bez formatu ;)

Zmiana warunku

```
while (sprawdź czy jest true) {
    Rób coś
    Dokonaj zmian w warunku
}

i = 0
while ( i < 10) {
    console.log(i)
    i = i + 1
}
```

Program portfel

```
portfel = 2000
cenaJednejSztuki = 21
liczbaKupionychProduktow = 0

while (portfel >= cenaJednejSztuki) {
    Od portfel odejmij cenaJednejSztuki
    Do liczbaKupionychProduktow dodaj 1
}
console.log(liczbaKupionychProduktow)

//kolejne iteracje dopóki nie zostanie spełniony warunek
```

Kolejna porcja operatorów

-- Dekrementacja

++ Inkrementacja

= Operator przypisania

+= ; *= Operator przypisania połączony z działaniem na wartości

```
while (portfel >= cenaJednejSztuki) {
    portfel -= cenaJednejSztuki
    liczbaKupionychProduktow++
}
```

Pętla for

```
for (licznik; warunek; iterator) {
    instrukcja1
    instrukcja2
    ...
}
```

Licznik przeniesiony do pętli, podobnie jak iterator

Pętla for a pętla while

```
[PĘTLA FOR - SCHEMAT]
for (inicjalizacja licznika; warunek; iterator)
{
    instrukcja1
    instrukcja2
    ...
}
```

```
[PĘTLA WHILE - SCHEMAT]
Inicjalizacja licznika
while (warunek) {
    instrukcja1
    instrukcja2
    …
    Aktualizacja licznika
}
```

Pętla for i pętla while porównanie

```
i = 0
while ( i < 10) {
    console.log(i)
    i = i + 1
}
```

```
for ( i = 0; i < 10; i = i + 1 ) {
    console.log(i)
}
```