

Struktury logiczne #6 (i cykliczne #7)

W jaki sposób sterować programem. Rozwiązaniem mogą być struktury logiczne i cykliczne.

- a. Instrukcja warunkowa if
- b. Instrukcja warunkowa switch
- c. Pętla while
- d. Pętla for
- e. Piszemy pierwszy program (w tym bootcampie, a może i w Twoim życiu).

Podejmowanie decyzji (wybór)

Program bardzo często zachowują się inaczej w zależności od danych które posiadają i celu jaki mają. Program musi podjąć decyzję co zrobić w określonej sytuacji.

Myślące istoty działają w ten sam sposób.

Program 'Sprzedawca' => Program ma ustalić czy sklep jest otwarte czy zamknięte. Pod uwagę powinien więc wziąć dwie rzeczy. Dzień tygodnia i dodatkowe sprawdzenie czy to nie jest niedziela handlowa. Sprawdzenie czy to godziny otwarcia. [dane i kolejność decyzji] To będzie zadanie domowe dla Ciebie.

Podejmowanie decyzji (wybór)

Zobaczmy inny przykład w formie programu 'kontrola drogowa'. Algorytm (uproszczony) wyglądałby tak:

1. Policja kamufluje się przy drodze przy której prędkość maksymalna wynosi ... [dane]
2. Policja mierzy prędkość samochodu (dane).
3. Policjant **podejmuje decyzje** czy zatrzymać pojazd (czy jedzie szybciej niż może). Jeśli przekroczył, zatrzymuje i przechodzi do punktu 4, jeśli nie, mówi w duchu 'no trudne' i wraca do punktu 2.
4. Policjant **podejmuje decyzje** o kwocie mandatu w oparciu o przedziały w taryfikatorze. Ustala wielkość przekroczenia prędkości (działanie matematyczne) i porównuje ją z przedziałami taryfikatora (dane muszą być dostępne).
5. Podobnie **podejmuje decyzję** dotyczącą punktów karnych..
6. Po przydzieleniu punktów karnych, policjant **podejmuje decyzje** czy zatrzymać prawo jazdy. Sprawdza czy przekroczona została liczba dopuszczalnych punktów karnych (maksymalną liczbę punktów przechowuje w swojej pamięci). Jeśli została przekroczona liczba punktów to zabiera prawo jazdy, jeśli nie, to mówi, 'do zobaczenia, następnym razem!' i **kończy** kontrolę i wraca do punktu 2
7. Po zabraniu prawa jazdy **podejmuje decyzje** czy wzywać holownik czy ...

Instrukcja warunkowa if

Podstawowe struktur w językach programowania to instrukcja warunkowa if

If, czyli tłumacząc na polski 'jeśli'

Program sprawdzający wiek:

```
pytanie = 'Ile masz lat chłopczyku?'
```

```
odpowiedz = [wartość podana przez kupującego]
```

```
If (odpowiedz > 18) 'Możesz kupić papierochy, ale odradzam. Może banana?'
```

If - przykład z warunkiem prawdziwym

Podstawowe struktur w językach programowania to instrukcja warunkowa IF

If, czyli tłumacząc na polski 'jeśli'

Program

`pytanie` = 'Ila masz lat chłopczyku?'

`odpowiedz` = [wartość podana przez kupującego]

`If` (odpowiedz > 18) 'Możesz kupić papierochy, ale odradzam. Może banana?'

Program w działaniu

`odpowiedz` = 22

`If` (22 > 18) 'Możesz kupić papierochy, ale odradzam. Może banana?'

// 22 > 18 => TAK -wynikiem tego porównania jest wartość `true`

//TO CO W (PO) IF WYKONA SIĘ.

If - przykład z warunkiem nieprawdziwym

Podstawowe struktur w językach programowania to instrukcja warunkowa IF

If, czyli tłumacząc na polski 'jeśli'

Program

`pytanie` = 'Ila masz lat chłopczyku?'

`odpowiedz` = [wartość podana przez kupującego]

`If` (odpowiedz > 18) 'Możesz kupić papierochy, ale odradzam. Może banana?'

Program w działaniu

`odpowiedz` = 16

`If` (16 > 18) 'Możesz kupić papierochy, ale odradzam. Może banana?'

// 16 > 18 => NIE - wynikiem tego porównania jest wartość `false`

// TO CO W (PO) IF NIE WYKONA SIĘ.

If else - a co jeśli nie?

Do tej pory jeśli warunek nie był prawdziwy nic nie robiliśmy dalej, ale możemy to zmienić rozbudowując naszą instrukcję warunkową o słowo else.

Wariantem instrukcji if jest instrukcja **if else**

```
If (wiek > 18) 'Możesz kupić papierosy (ale odradzam)'  
else 'O ty gów***, może jeszcze piwko co, ty *****'
```

If else

```
if(warunek) {  
    //zrób wszystko co jest tutaj jeśli warunek jest prawdą np. 2>1  
}  
else {  
    //zrób wszystko co jest tutaj jeśli warunek jest fałszem np 2>2  
}
```


Zagnieżdżanie instrukcji w instrukcji

Wariantem instrukcji if jest if else na przykładzie programu sklepowego 'Sprzedać czy nie?'

```
If ([klient twierdzi, że ma co najmniej 18 lat])
```

```
    // ok, to piszemy program dalej
```

```
else
```

```
    'Przeprosić i powiedzieć, by przyszedł jak skończy 18 lat'
```

Zagnieżdżanie instrukcji w instrukcji

Wariantem instrukcji if jest if else na przykładzie programu sklepowego 'Sprzedać czy nie?'

```
If ([klient twierdzi, że ma co najmniej 18 lat])  
  If ([wierzymy mu?])  
    "Proszę, o to Twoje piwo"  
  else  
    //piszemy program dalej  
else  
  'Przeprosić i powiedzieć, by przyszedł jak skończy 18 lat'
```

Zagnieżdżanie instrukcji w instrukcji

Wariantem instrukcji if jest if else na przykładzie programu sklepowego 'Sprzedać czy nie?'

```
If ([klient twierdzi, że ma co najmniej 18 lat])
```

```
    If ([wierzemy mu?])
```

```
        "Proszę, o to Twoje piwo"
```

```
    else
```

```
        If ([czy ma dowód?])
```

```
            // co wtedy?
```

```
        else
```

```
            // co wtedy?
```

```
else
```

```
    'Przeprosić i powiedzieć, by przyszedł jak skończy 18 lat'
```

Zagnieżdżanie instrukcji w instrukcji

Wariantem instrukcji if jest if else na przykładzie programu sklepowego 'Sprzedać czy nie?'

```
If ([klient twierdzi, że ma co najmniej 18 lat])
```

```
    If ([wierzemy mu?])
```

```
        "Proszę, o to Twoje piwo"
```

```
    else
```

```
        If ([czy ma dowód?])
```

```
            'Sprzedaj i powiedz żarcik, że młodo wygląda i też tak byś chciał'
```

```
        else
```

```
            "Bez dowodu nie mogę Ci sprzedać, wróć z dowodem, a ja poczekam"
```

```
    else
```

```
        'Przeprosić i powiedzieć, by przyszedł jak skończy 18 lat'
```

Warunek logiczny - false lub true

```
if(warunek) {  
  //zrób wszystko co jest tutaj  
}
```

Przykładowe operatory

$X > Y$ true gdy np. $5 > 2$ false gdy np. $2 > 3$

$X \geq Y$ true gdy np. $3 \geq 3$ false gdy np. $2 \leq 1.99999999$

$X == Y$ porównanie a więc gdy X równe Y np. $2 == 2$ to true, ale $3 == 2$ to false

$X != Y$ tru zostanie zwrócone gdy X będzie różny niż Y np. $3 != 2$ zwróci true, ale $1 != 1$ już zwróci false

Switch

Program 'wyświetlacz', który będzie wyświetlała różne komunikaty w zależności od dnia tygodnia.

```
switch (dzienTygodnia)
  case 'poniedziałek'
    'Nowy tydzień, nowe możliwości'
  case 'piątek'
    'Piątunio!!!!'
  case 'sobota'
  case 'niedziela'
    'Weekendzie trwaj!'
  default
    'Jesteś poza czasem, gratulacje, zawiesiłeś matrix'
```