



# Baltie 3

---

*Podręcznik do nauki programowania  
dla klas IV - VI szkoły podstawowej*

Tadeusz Sołtys, Bohumír Soukup

# Nawiasy

## Nawiasy dla wyrażeń

Nawiasów okrągłych używa się:

- do zaznaczenia części wyrażenia, która ma być obliczona wcześniej,
- przy wydzieleniu złożonych parametrów funkcji,
- gdy jest to konieczne do dokładnego określenia warunku pętli lub warunku polecenia warunkowego,
- uproszczenia programu.

Przykłady stosowania nawiasów okrągłych:

- wyświetl na scenie wynik następującego wyrażenia liczbowego:



← 5 × ( 2 + 6 ) × 4

- na polu o współrzędnych 2,2 umieść średnią arytmetyczną liczb 12, 13, i 8:



2 2 ← ( 12 + 13 + 8 ) / 3

## Nawiasy dla indeksowania

Nawiasów kwadratowych używa się, gdy chcemy mieć dostęp do wybranego elementu lub elementów na podstawie indeksu (indeksów).

Przykłady stosowania nawiasów kwadratowych:

- wypisz drugi element ciągu znaków „Abraka dabrak” (wynikiem jest znak „a”):



Abraka dabrak [ 2 ]

- wypisz elementy od drugiego do piątego ciągu znaków „Abraka dabrak” (wynikiem jest ciąg znaków „brak”):



Abraka dabrak [ 2 .. 5 ]

# Instrukcje warunkowe proste (IF)

Instrukcji warunkowej (IF) używamy, gdy chcemy, by program wykonał określone polecenie lub blok poleceń tylko wtedy, gdy spełniony zostanie pewien warunek.

Tworzenie instrukcji warunkowej:

- wstawiamy element IF (jeśli):



- definiujemy warunek, np.:



- określamy co ma zrobić Baltie, gdy warunek zostanie spełniony, np.:



Przykłady instrukcji warunkowych:

- jeśli na polu przed Bałtciem jest prezent (przedmiot 25), wyczaruj przed Bałtciem rower (przedmiot 30):
  - pełny zapis (tak, jak w języku C):
  - skrócony zapis (jeżeli sytuacja nam pozwoli, to będziemy używać skróconych zapisów):
- jeśli naciśnięto lewy przycisk myszy, obróć Bałticia w lewo:
- jeśli na pozycji Bałticia znajduje się przedmiot 14, odtwórz dźwięk i czekaj na naciśnięcie klawisza lub przycisku myszy:
- jeśli spełniony jest warunek, że liczba 5 jest większa od liczby 3, to 5 razy wykonaj blok poleceń {wyczaruj przed Bałtciem przedmiot 61 i idź o jedno pole do przodu}:
  - pełny zapis:
  - zapis uproszczony (bez nawiasów warunkowych):
  - jeszcze krótszy zapis (pętla jest w tym przypadku traktowana jak jedno polecenie, dlatego może nie być ujęta w nawiasy klamrowe, ale spacja musi oddzielać 3 od 5, w przeciwnym razie sprawdzany byłby warunek 5>35):
  - najkrótszy zapis (3 nie łączy się z literalem 5, dlatego nie potrzeba spacji oddzielającej):

## Instrukcje warunkowe kompletne (IF ... ELSE)

Instrukcji warunkowej kompletnej (IF ... ELSE) używamy, gdy chcemy, by program wykonał inne polecenie lub blok poleceń wtedy, gdy postawiony warunek zostanie spełniony, a inne wtedy, gdy warunek spełniony nie będzie.

Tworzenie instrukcji warunkowej kompletnej:

- wstawiamy element IF (jeśli):
- definiujemy warunek, np.:
- określamy co ma zrobić Bałtcie, gdy warunek zostanie spełniony, np.:
- wstawiamy element ELSE (w przeciwnym razie):
- i mówimy Bałtciu co ma zrobić, gdy postawiony warunek nie zostanie spełniony, np.:

Przykłady instrukcji warunkowych kompletnych:

- jeśli na polu przed Baltiem znajduje się dowolny przedmiot, idź o jedno pole do przodu; w przeciwnym razie wyczaruj przed Baltiem przedmiot 1122 i także idź o jedno pole do przodu:

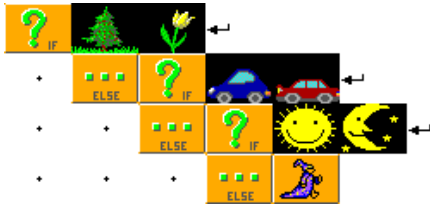


- jeśli na pozycji Baltiego znajduje się prezent (przedmiot 25), to wyczaruj na tej pozycji auto (przedmiot 46) i idź o jedno pole do przodu; w przeciwnym wypadku tylko idź:



- jeśli na pozycji przed Baltiem jest drzewo, to wyczaruj kwiatek; w przeciwnym razie, jeśli na pozycji przed Baltiem jest niebieski samochód, to czaruj czerwony samochód; w przeciwnym razie, jeśli na pozycji przed Baltiem jest słońce, to czaruj księżyc; w przeciwnym razie (czyli we wszystkich pozostałych przypadkach), idź:

- pełny zapis wielokrotnego (zagnieżdżonego) IF (z wcięciami):



- zapis uproszczony (takiego sposobu będziemy używać):






## Stałe i zmienne w programie

### Stałe

**Stała** to element z nazwą, którego wartość jest taka sama przez cały czas działania programu.

Wyróżnia się trzy typy stałych i oznacza kolorami, podobnie jak w przypadku literałów:

- **stałe całkowite** (jasnoniebieskie) ,
- **stałe rzeczywiste** (zielone) ,
- **stałe tekstowe** (żółte) .

W każdym z trzech typów stałych wyróżnić można:

- **stałe nazwane** - tworzone i nazywane przez użytkownika, np.:



- **stałe systemowe**, np.:



- **wielokrotne** - używane do pracy ze współrzędnymi, np.:



Przykłady użycia stałych w programie:

- 15 razy (szerokość sceny w polach) wykonaj blok poleceń:






- wyczaruj kaczkę (przedmiot 12110) na polu o współrzędnych X,Y:









## Zmienne

**Zmienna** to element z nazwą, którego wartość może się zmieniać w czasie działania programu. To takie pudełko z nazwą, w którym można przechować zawartość, którą następnie, w dowolnej chwili, można użyć w programie. Zawsze trzeba rozróżniać między samą zmienną (tzn. nazwanym pudełkiem) i wartością zmiennej (tzn. zawartością pudełka). Napotykając słowo „zmienna” w poniższym tekście, należy zwracać uwagę na to, co ono oznacza - czy pudełko, czy jego zawartość, a może jedno i drugie. W Baltie zmienne przedstawiane są jako szufladki lub koszyczki.



Wyróżnia się trzy typy zmiennych:

- **zmienne całkowite** (jasnoniebieskie) ,
- **zmienne rzeczywiste** (zielone) ,
- **zmienne tekstowe** (żółte) .

W każdym typie wyróżnia się kilka rodzajów zmiennych:

- **zmienne obrazowe** - ich nazwa nie może zostać zmieniona, np.: 
- **zmienne nazwane** - tworzone i nazywane przez użytkownika, np.: 
- **zmienne z numerem** (indeksowana zmienna = jednowymiarowa tablica), np.:  lub 
- **tablice**, np.: 
- **zmienne wielokrotne** - używane do pracy ze współzrędnymi, np.: 

Ze względu na zasięg zmienne dzieli się na globalne (szufladki) i lokalne (koszyczki).

- **zmienne globalne (publiczne)**, czyli takie, które są widoczne dostępne ze wszystkich miejsc programu. Zmienna globalna może mieć nadaną wartość początkową bezpośrednio po uruchomieniu programu. Zmienne globalne przyjmują postać szufladki, np.: 
- **zmienne lokalne (prywatne)** - są widoczne i dostępne tylko dla pomocnika, którego definicja zawiera tę zmienną. Ani inni pomocnicy, ani program główny nie mają do tej zmiennej dostępu. Zmienne lokalne mają postać koszyczków: 

**Nazwa zmiennej** może być dowolnym ciągiem znaków, przy czym:

- nie może zawierać polskich znaków diakrytycznych takich, jak: ą, ć, ę, ł itp.,

- musi być niepowtarzalna w konkretnym banku zmiennych (w jednym banku nie może być innej zmiennej o takiej samej nazwie).

**Wartość zmiennej** jest w zależności od typu liczbą lub ciągiem znaków. W czasie wykonywania programu możliwe są następujące operacje na zmiennych:

- przypisanie konkretnej wartości, np.:



- zwiększenie wartości, np.:



- zmniejszenie wartości, np.:



Przykłady użycia zmiennych:

- A razy idź o jedno pole:



- przesuń Baltiego na pole o współrzędnych X,Y:



- 8 razy powtarzaj blok poleceń: P razy {wyczaruj drzewo (przedmiot 12) i idź o jedno pole}, obróć Baltiego w lewo oraz zwiększ wartość zmiennej P o jeden (początkową wartość P ustawiono na 1):



## Pętle WHILE, DO-WHILE oraz FOR

### Pętle WHILE i DO-WHILE

**Pętli WHILE (DO-WHILE)** używamy, gdy chcemy, by program powtarzał wskazane polecenie lub blok poleceń dopóty, dopóki spełniony jest pewien warunek.

Wyróżnia się dwa rodzaje pętli:

- **WHILE** – spełnienie warunku sprawdzane jest przed wykonaniem polecenia (bloku poleceń):



- **DO-WHILE** - spełnienie warunku sprawdzane jest po wykonaniu polecenia (bloku poleceń):



Przykład użycia pętli WHILE: wykonuj blok poleceń (ścianę nad drogą Baltiego) dopóty, dopóki przed Baltiem nie ma przedmiotu:



### Pętli FOR

**Pętli FOR** używa się, gdy liczba powtórzeń jest zależna od wartości tzw. **zmiennej sterującej**.

Tworzenie pętli FOR:

- wstawiamy element FOR:



- wstawiamy zmienną sterującą, np.:



- określamy wartość początkową, np.:



- określamy wartość końcową, np.:



- podajemy krok, czyli o jaką wartość ma się zmieniać zawartość zmiennej sterującej (domyślnie o 1), np.:



- i mówimy Baltiemu jakie polecenie lub blok poleceń ma wykonać, np.:



Przykłady pętli FOR:

- dla każdej liczby z zakresu od 1 do 9 wyczaruj kwiatka (przedmiot 9) na podanych współrzędnych (jeżeli krok jest równy 1, to może zostać pominięty):



- dla każdej liczby z zakresu od 9 do 1 wyczaruj kwiatka (przedmiot 9) na podanych współrzędnych (krok -1):



## Przerwij (BREAK)



Elementu **BREAK (przerwij powtarzanie)** używa się do bezpośredniego przerywania powtarzania.

Za elementem BREAK może stać liczba zagnieżdżonych powtórzeń, które chcemy przerwać, np.:



Jeśli użyjemy elementu BREAK z elementem nieskończoność, przerwiemy wszystkie powtórzenia.

Przykład zastosowania polecenia BREAK: powtarzaj w nieskończoność czytanie klawisza; przerwij, gdy naciśnięto klawisz End:



## Procedury z parametrami

**Procedury z parametrami** to takie, które wykorzystują podczas pracy zmienne. W procedurach można wykorzystywać wszystkie rodzaje zmiennych, ale specjalnie dla procedur (pomocników) przeznaczone są zmienne lokalne (koszyki).

Zmienne lokalne charakteryzują się tym, że dostęp do nich ma tylko procedura, do której zostały przypisane, natomiast nie ma do nich dostępu ani program główny, ani żadna inna procedura. Poza tym zmiennych lokalnych używa się analogicznie jak zmiennych globalnych.

Procedury z parametrem lub parametrami są bardziej uniwersalne od procedur bez parametrów. Na dodatek liczba parametrów jest w zasadzie nieograniczona.

Na przykład w procedurze bez parametru, której zadanie polega na wyczarowaniu granic ogródka, konieczne jest określenie wielkości wspomnianego ogródka już na etapie programowania, np.:



Natomiast w przypadku procedury z parametrem, wielkość ogródka można określić w chwili wywołania procedury, np.:



## Sposoby podawania parametrów

Istnieją dwa sposoby przekazania parametrów procedurze:

- przez wartość,
- przez odniesienie.

**Przekazanie przez wartość** ogranicza się do przekazania parametrowi procedury jedynie wartości i realizowany jest za pośrednictwem elementu przydział, np.:



**Przekazanie przez odniesienie** przekazuje wszystkie cechy parametru i realizowane jest w następujący sposób:



## Praca z multimediami

Wykorzystanie multimediiów w środowisku Baltie ułatwia **Tablica plików i obszarów**.

Otwieramy ją przy pomocy przycisku **Tablica**:






Oprócz współrzędnych obszarów, Tablica umożliwia przechowywanie nazw plików:

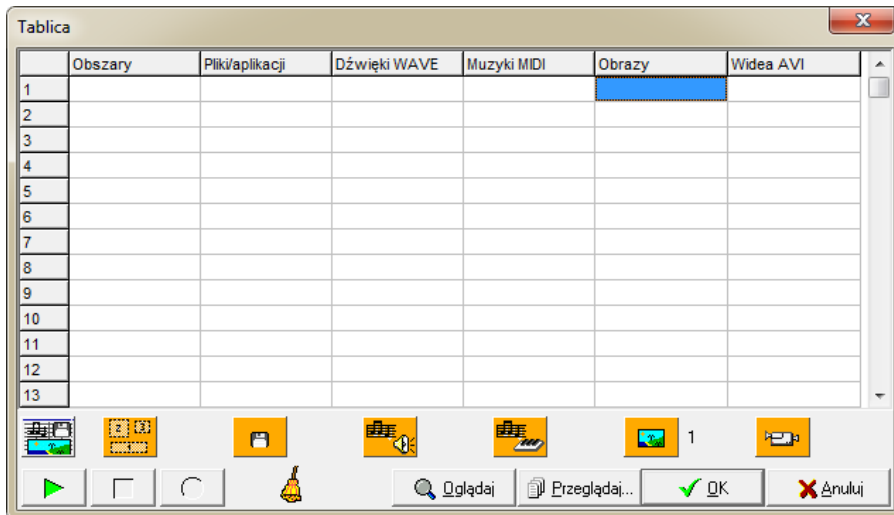
- dźwiękowych:





- muzycznych:  

- graficznych:  

- wideo:  

- i innych.


Przyciski poszczególnych typów plików znajdują się w dolnej części Tablicy plików i obszarów:

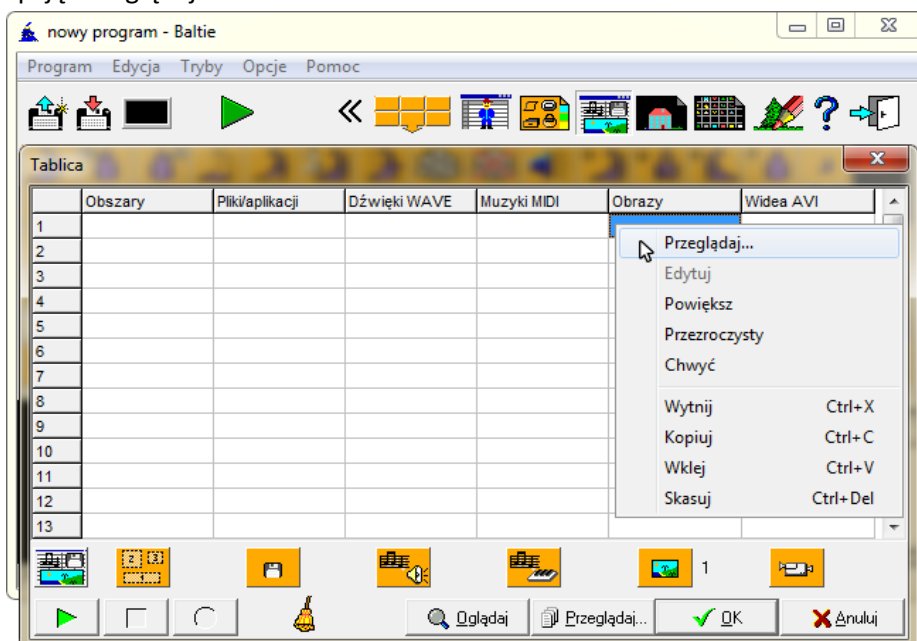


Aktywna (bieżąca) komórka tablicy zawsze jest zacieniowana. Aby zmienić komórkę aktywną, wystarczy na niej kliknąć.

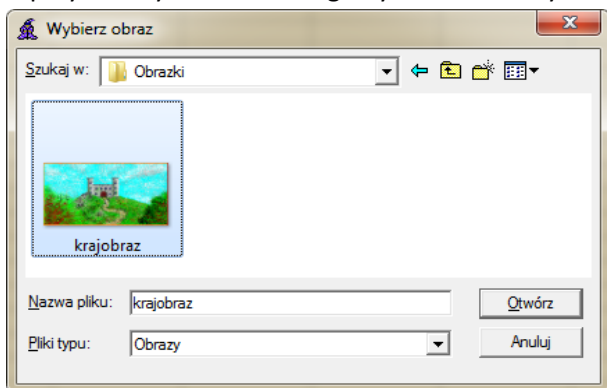
### **Dodawanie plików do tablicy plików i obszarów**

W celu dodania pliku do tablicy należy wybrać odpowiednią komórkę, a następnie:

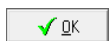
- kliknąć na przycisku Tablica:  

- wskazać odpowiednią komórkę i z menu kontekstowego (po kliknięciu prawym przyciskiem myszy) wybrać opcję Przeglądaj...:



- w przywołanym oknie dialogowym wskazać wybrany plik i kliknąć na przycisku Otwórz:



- kliknąć na przycisku OK:



## Umieszczanie pliku w programie

Aby umieścić w programie plik znajdujący się w Tablicy plików i obszarów należy:

- otworzyć Tablicę plików i obszarów:



- wskazać wybrany plik poprzez kliknięcie na komórce zawierającej jego lokalizację i nazwę,
- kliknąć w dolnej części okna Tablicy plików i obszarów na przycisku w odpowiedniej kolumnie, w naszym przypadku:



- umieścić przycisk wraz z numerem pliku w tym miejscu programu, gdzie ma zostać wykonana jego treść, np.:



W jednym programie możemy wykorzystać kilka typów plików.

## Praca z obrazkami

Baltie obsługuje kilka formatów plików graficznych:

- Bitmapy z rozszerzeniem .bmp. Obrazy tego formatu uznawane są przez prawie wszystkie edytory graficzne. Do ich utworzenia można użyć aplikacji Rysowanie. Jeśli 16 kolorów Baltiego to za mało, polecamy Windowsowy edytor graficzny Paint.
- Ikony z rozszerzeniem .ico. Te pliki używane są przez system do wyświetlania ikon. Ikony mogą być przezroczyste.
- Wektorowe pliki graficzne z rozszerzeniem .wmf (Windows Meta File). Są plikami poleceń do tworzonej grafiki i tekstu.
- Banki Baltiego z rozszerzeniem .b00 - .b99 lub .c00 - .c99. Można narysować te obrazy przy pomocy Baltiego lub użyć banków standardowych.

## Wczytanie obrazka z pliku na scenę

Aby wstawić obrazek zapisany w tym samym folderze co program, wystarczy użyć polecenia:



Jeśli plik jest przechowywany w innym folderze, należy podać ścieżkę dostępu do tego pliku, np.:



## Wczytanie obrazka umieszczonego w Tablicy plików i obszarów

Wykorzystanie obrazów z Tablicy plików i obszarów zwalnia użytkownika z podawania nazwy i lokalizacji pliku:

- proste wyświetlenie obrazka na ekranie:



- wyświetlenie obrazka z dostosowaniem do pełnego ekranu:



- wyświetlenie obrazka na ekranie na pozycji XY=5,5:



- wyświetlenie obrazka z dostosowaniem do obszaru X1Y1=5,5; X2Y2=8,8:



## Animacje komputerowe

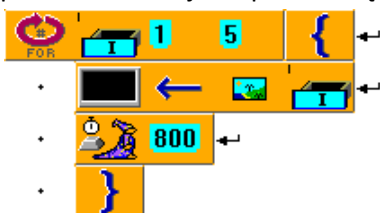
**Animację komputerową** określić można jako ruchomy obraz stworzony przy pomocy komputera, składający się z wyświetlanych w odpowiedniej kolejności i z odpowiednią szybkością obrazów (tzw. ramek), co wywołuje u odbiorcy wrażenie ruchu.

### Sekwencja obrazów o zmiennej zawartości

Najprostszym sposobem uzyskania animacji w Baltie jest wyświetlenie we wskazanej kolejności kilku obrazków z **Tablicy plików i obszarów**, np. przy pomocy pętli FOR:



Niestety w powyższym przykładzie obrazki będą wyświetlane zbyt szybko. Można to zmienić poprzez umieszczenie polecenia czekaj z odpowiednią liczbą milisekund, np.:



Aby uzyskać odtwarzanie animacji w pętli wystarczy umieścić na początku programu liczbę powtórzeń lub wartość nieskończoność, np.:



### Wrażenie ruchu

Wrażenie ruchu wywołać można przez symulowanie przemieszczania się obiektu lub zmiany jego wyglądu, przy czym w obu przypadkach odpowiednio dobrać należy czasy wyświetlania kolejnych ramek animacji.

Przemieszczanie się obiektu w środowisku Baltie można uzyskać poprzez cykliczne czyszczenie sceny i wyczarowywanie obiektu w innym miejscu, np. animację duszka wędrującego po scenie uzyskać można w następujący sposób:



Możliwa jest również animacja przedmiotu znajdującego się na określonej pozycji. W kilku bankach środowiska Baltie umieszczono sekwencje przedmiotów nadające się do wykorzystania w animacji, np. symulację otwierania drzwi uzyskać można w następujący sposób:

