

1. Tworzenie programu
2. Powtarzanie poleceń
  - 2.1. Pętla prosta
  - 2.2. Pętla w pętli
3. Sytuacje warunkowe
4. Zmienne i obliczenia
5. Stosowanie procedur



## Warto powtórzyć

1. Czym jest program komputerowy?
2. Czym jest język programowania?
3. Na czym polega przypisanie wartości zmiennej w programie komputerowym?
4. Czym jest podprogram?
5. Kiedy mamy do czynienia z sytuacją warunkową?
6. Na czym polega iteracja?


## 1. Tworzenie programu

Tworzenie programu w środowisku Baltie polega na ułożeniu ciągu poleceń (instrukcji języka programowania), które komputer powinien wykonać. W tradycyjnym programowaniu polecenia zapisywane są tekstowo, natomiast w Baltie instrukcje tworzące kod programu reprezentowane są przez elementy graficzne wybierane z **panelu poleceń** (rys. 1.) i **przedmioty** wybierane z **banków przedmiotów**.

Wszystkie polecenia tworzące program umieszczamy w obszarze roboczym, a następnie uruchamiamy program. Polecenia programu wykonywane są w kolejności wierszy od lewej do prawej. Raz utworzony program można uruchomić wiele razy. Można go też modyfikować.








### Aby utworzyć program, należy:



- wybrać tryb pracy – **Tryby/Programowanie**, np. **Nowicjusz**,
- umieścić na scenie kod programu, składający się z elementów (poleceń) wybieranych z panelu poleceń i przedmiotów (danych) wybieranych z banku przedmiotów,
- uruchomić program, naciskając przycisk .

Efekt wykonania programu zobaczymy w oddzielnym oknie.



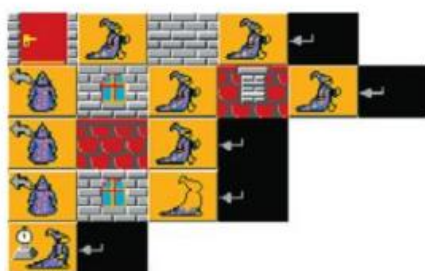
**Rys. 1.** Okno programu Baltie otwarte w trybie **Programowanie/Nowicjusz**. Na scenie jest widoczny krótki kod programu

Pierwsze trzy polecenia w panelu poleceń dotyczą zmiany położenia czarodzieja Baltiego. Konkretnie polecenie wybieramy zależnie od czynności, jaką Baltie ma wykonać, np.: **Idź** , **W lewo zwrot** , **W prawo zwrot** , **Niewidzialny** , **Czekaj** 

Liczbę kroków, jaką ma przebyć czarodziej, określamy, umieszczając przed poleceniem **Idź** przedmiot z odpowiednią cyfrą (cyframi), np. **3**  (trzy kroki), **1 2**  (dwanaście kroków).

### Uwaga !


Wszystkie polecenia opisane są w **Pomocy** do programu Baltie.




**Rys. 2.** Program do ćwiczenia 1.



**Rys. 3.** Wynik działania programu po modyfikacji – ćwiczenie 1.

Polecenie **Czekaj**  powoduje oczekiwanie czarodzieja na naciśnięcie dowolnego klawisza lub przycisku myszy. Jeśli chcemy określić czas czekania, należy po poleceniu wstawić liczbę określającą czas w milisekundach (1 sekunda to 1000 milisekund).

Należy dbać o czytelność programu i nie umieszczać wszystkich poleceń w jednym wierszu.

Na rysunku 2. polecenia umieszczono w pięciu wierszach – trzy z nich zaczynają się od polecenia zmiany kierunku ruchu Baltiego w lewo. Każdy wiersz kończy się poleceniem **Koniec wiersza** .



### Ćwiczenie 1. Piszemy program w Baltie



1. W obszarze roboczym umieść polecenia pokazane na rysunku 2. i uruchom program.
2. Zmodyfikuj program, aby otrzymać dom, jaki pokazano na rysunku 3. Postaraj się zastosować jak najmniej poleceń.
3. Zapisz program w pliku pod nazwą **Domek**.



## 2. Powtarzanie poleceń

Zamierzamy utworzyć program, w którym powstanie droga składająca się z trzynastu jednakowych przedmiotów. Aby wyświetlić jeden przedmiot, Baltie musi go wyczarować i przejść jedno pole. Czy aby wyświetlić trzynaście takich samych przedmiotów, musimy trzynaście razy umieścić te same polecenia?

### 2.1. Pętla prosta

W programie Baltie powtarzające się polecenia umieszczamy w nawiasach klamrowych, stosując element **Blok poleceń** . Przed nawiasem otwierającym umieszczamy liczbę powtórzeń. Jeśli zamiast liczby wstawimy element **Nieskończoność** , polecenia będą powtarzane, dopóki nie zamkniemy okna z efektem wykonania programu.

Na rysunkach 4a i 4b przedstawiono dwie wersje programu, w którym Baltie buduje szosę i ustawia na niej samochody. W wersji widocznej na rysunku 4a zastosowano dwie pętle. W pierwszej pętli rysowana jest szosa, natomiast w drugiej – samochody. W wersji przedstawionej na rysunku 4b trzy wiersze poleceń z dwiema pętlami z rysunku 4a zastąpiono jednym wierszem z jedną pętlą. Wynik działania programu w obu przypadkach jest taki sam (rys. 5).



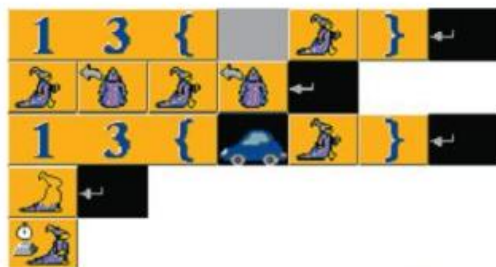
#### Ćwiczenie 2. Tworzymy program z zastosowaniem pętli

1. Utwórz program przedstawiony na rysunku 4a i uruchom go. Następnie na tej samej scenie umieść rozwiązanie II (rys. 4b). Nie usuwaj rozwiązania I, tylko przed każdym wierszem z poleceniami pierwszego programu umieść polecenie

#### Komentarz wierszowy

2. Uruchom drugi program i sprawdź, czy działa tak samo.
3. Zapisz program w pliku pod nazwą *Droga*.

**Wskazówka:** Wstawienie do programu na początku wiersza **Komentarza wierszowego** powoduje, że Baltie pominie wszystkie elementy od tego elementu do elementu kończącego wiersz. Elementy pomijane w programie są przekreślone.



Rys. 4a. Program do sceny pokazanej na rysunku 5. – rozwiązanie I



Rys. 4b. Program do sceny pokazanej na rysunku 5. – rozwiązanie II



Rys. 5. Wynik działania programów z rysunków 4a i 4b



### Ćwiczenie 3. Modyfikujemy program wykorzystujący pętlę

1. Zmodyfikuj program utworzony w ćwiczeniu 2. (rozwiązanie II), tak aby czarodziej zbudował ścieżkę przedstawioną na rysunku 6.
2. Zapisz plik pod tą samą nazwą.



Rys. 6. Wynik działania programu – ćwiczenie 3.

## 2.2. Pętla w pętli

Zamierzamy zbudować mur o długości dwunastu przedmiotów i wysokości czterech przedmiotów, którego górna część jest symetrycznym odbiciem dolnej części. Jak można usprawnić tworzenie programu, którego efektem działania będzie taki mur?

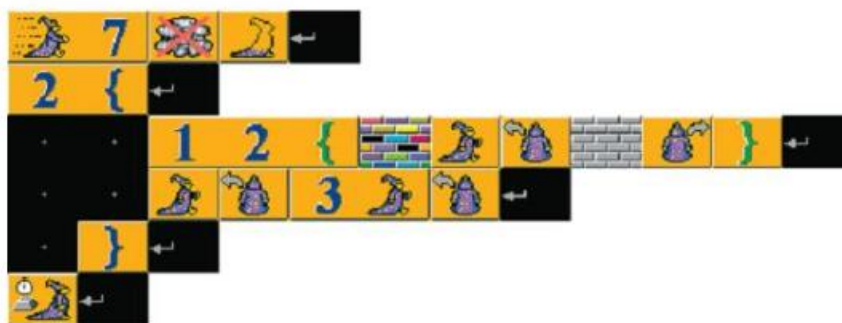
Do rozwiązania tego zadania można zastosować pętlę zagnieżdżoną, czyli pętlę w pętli. Zadanie można rozwiązać na kilka sposobów. Pokażemy jeden z nich. Aby postawić pierwszą i drugą warstwę muru, wykorzystamy rozwiązanie II z ćwiczenia 2. (rys. 4b).



### Ćwiczenie 4. Piszemy program z pętlą prostą

1. Napisz program, w którym Baltie zbuduje mur o długości dwunastu przedmiotów i wysokości dwóch przedmiotów. W każdej warstwie muru Baltie ma umieścić elementy w innym kolorze.
2. Zapisz program w pliku pod nazwą *Mur*.

Aby zbudować kolejną część muru, zastosujemy pętlę zagnieżdżoną – Baltie wykona dwukrotnie polecenie budowania muru o długości dwunastu przedmiotów i wysokości dwóch przedmiotów (rys. 7.).



Rys. 7. Program realizujący pętlę zagnieżdżoną – ćwiczenie 5.

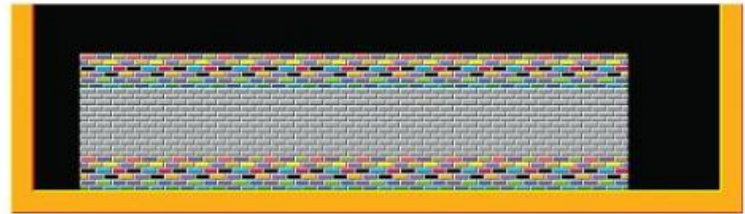


### Ćwiczenie 5. Piszemy program z pętlą zagnieżdżoną

1. Zmodyfikuj program utworzony w ćwiczeniu 4. zgodnie z rysunkiem 7. Zwróć uwagę na inny kolor nawiasów pętli wewnętrznej. Rozróżnienie kolorystyczne zostało wprowadzone po to, aby wyraźnie były widoczne pętli wewnętrzna i zewnętrzna. Sprawdź wynik działania programu (rys. 8.).
2. Zmodyfikuj program utworzony w punkcie 1., tak aby kolorowe cegielki były w środku muru, a jednolite – na dole i u góry.
3. Zapisz plik pod tą samą nazwą.



**Wskazówka:** Aby zwiększyć czytelność programu, zwłaszcza w przypadku pętli zagnieżdżonych, staraj się stosować wcięcia wierszy (rys. 7.).



Rys. 8. Wynik działania programu – ćwiczenie 5. (punkt 1.)

### 3. Sytuacje warunkowe

Zamierzamy utworzyć program, w którym Baltie umieści na scenie kolorową ścieżkę, a potem, idąc po niej, posadzi kwiatki, ale tylko na przedmiotach w kolorze zielonym. Jak zaprogramować możliwość decydowania, kiedy ma zostać posadzony kwiatek?



W programie Baltie polecenie realizujące sytuację warunkową ma postać:

**If** *warunek* **{** *lista\_poleczeń1* **}** **else** **{** *lista\_poleczeń2* **}**.

Jeśli *warunek* jest spełniony, to realizowane są polecenia *lista\_poleczeń1*, w przeciwnym przypadku realizowane są polecenia *lista\_poleczeń2*.

W wersji uproszczonej instrukcja ma postać: **If** *warunek* **{** *lista\_poleczeń1* **}** *kolejne polecenia programu*.

Jeśli *warunek* nie jest spełniony, realizowane są *kolejne polecenia programu*.

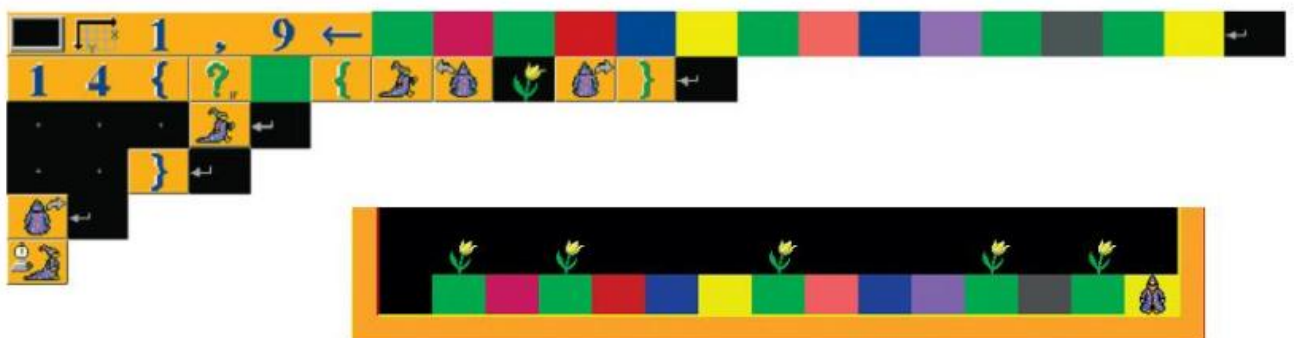
W środowisku Baltie jest kilka możliwości określania warunków. Można na przykład użyć przedmiotu z banku przedmiotów – warunek będzie spełniony, gdy w oknie działania programu wskazany w warunku przedmiot znajdzie się przed Baltiem.

W programie pokazanym na rysunku 9. zastosowano uproszczoną wersję instrukcji warunkowej. Jest ona wykonywana w pętli.



#### Uwaga

Elementy **if** i **else** są dostępne w panelu poleceń po wybraniu w trybie **Programowanie poziome Zaawansowany**.



Rys. 9. Przykład zastosowania uproszczonej instrukcji warunkowej – ćwiczenie 6.





### Ćwiczenie 6. Stosujemy instrukcję warunkową w wersji uproszczonej z warunkiem prostym

1. Umieść w obszarze roboczym polecenia pokazane na rysunku 9. Uruchom program. Odpowiedz na pytania: *Jakie polecenie jest wykonywane, gdy warunek jest spełniony, a jakie, gdy nie jest spełniony? Ile razy będzie wykonana instrukcja warunkowa?*
2. Zmodyfikuj program tak, aby Baltie dodatkowo zamienił przedmioty w kolorze granatowym i żółtym na choinki.
3. Zapisz program w pliku pod nazwą *Łąka*.

**Wskazówka:** W pierwszym wierszu programu jest wczytywana łąka. Do tego celu zastosowano polecenia (dostępne po wybraniu trybu **Programowanie/Zaawansowany**):

**Ekran** , **Współrzędne pola** , **Przydziel**  (przydzielamy blok przedmiotów do ekranu, począwszy od miejsca określonego podanymi współrzędnymi). Liczba przed przecinkiem określa kolumnę (liczoną od lewej, od 0 do 14), a liczba po przecinku – wiersz (liczony od góry, od 0 do 9). Współrzędne pola w lewym górnym rogu sceny to (0,0), a w prawym dolnym rogu – (14,9).


Warunek może być również złożony. Pomiedzy przedmiotami określającymi wa-

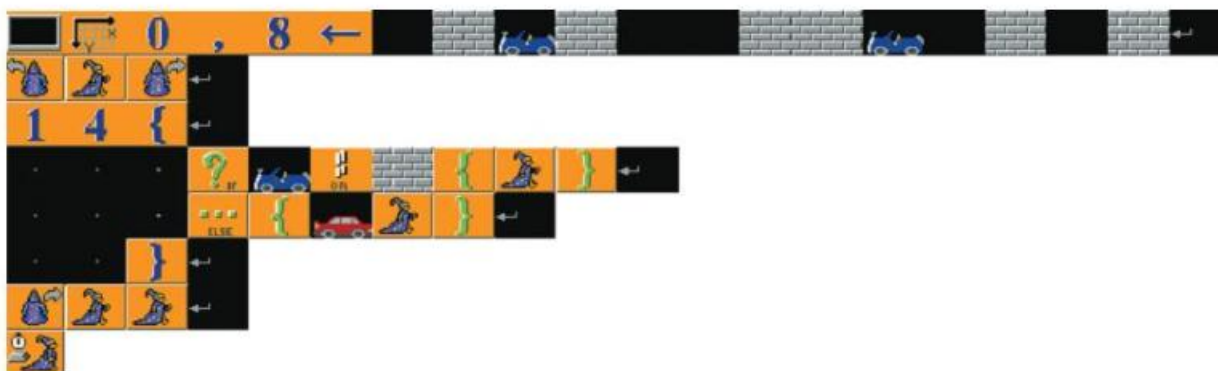
runek umieszczamy wtedy spójnik logiczny **i (and)**  albo **lub (or)** .

Więcej informacji na temat stosowania warunków można znaleźć w **Pomocy**.



### Ćwiczenie 7. Stosujemy instrukcję warunkową w wersji pełnej z warunkiem złożonym


1. Umieść w obszarze roboczym polecenia pokazane na rysunku 10. Uruchom program.
2. Omów, co dzieje się w poszczególnych wierszach programu. Odpowiedz na pytanie: *Jakie polecenie jest wykonywane, gdy warunek jest spełniony, a jakie, gdy nie jest spełniony?*
3. Zmodyfikuj program tak, aby Baltie zamiast zamieniać przedmioty koloru czarnego na czerwone samochody, uzupełniał mur (umieszczając przedmiot )
4. Zapisz plik pod nazwą *Warunek*.








Rys. 10. Przykład zastosowania instrukcji warunkowej pełnej z warunkiem złożonym – ćwiczenie 7.



**Aby umieścić w sposób losowy przedmiot na scenie, stosujemy polecenie**

**Liczba losowa:** 




Na przykład:  **6** generuje liczbę losową z liczb: 0, 1, 2, 3, 4 i 5, a polecenie:   **1 5**   **1 0** wyczarowuje żabę w przypadkowym miejscu sceny. Możemy też określać dowolny zakres liczb, spośród których będą losowane liczby całkowite.

Na przykład:  **1**  **6** generuje liczbę spośród: 1, 2, 3, 4, 5 i 6.



### Ćwiczenie 8. Stosujemy instrukcję warunkową w wersji uproszczonej i pełnej

1. Utwórz program, w którym w dolnym wierszu będą wyświetlane losowo liście . Jeśli czarodziej, idąc w prawą stronę, widzi przed sobą liść, zamienia go w żabkę i idzie dalej.
2. Zmodyfikuj program tak, aby w przeciwnym wypadku (brak liścia przed Bałtiem) czarodziej wyczarowywał choinkę.
3. Zapisz program w pliku pod nazwą *Liście*.

**Wskazówki:** Aby wylosować więcej niż jeden liść, umieść polecenie **Liczba losowa** w bloku poleceń i powtarzaj np. 7 razy:



 **7** {    **1**  **1 4**  **9** }




Zauważ, że jeśli zostanie wylosowana ta sama wartość  $x$ , to przedmiot zostanie wyświetlony w tym samym miejscu.

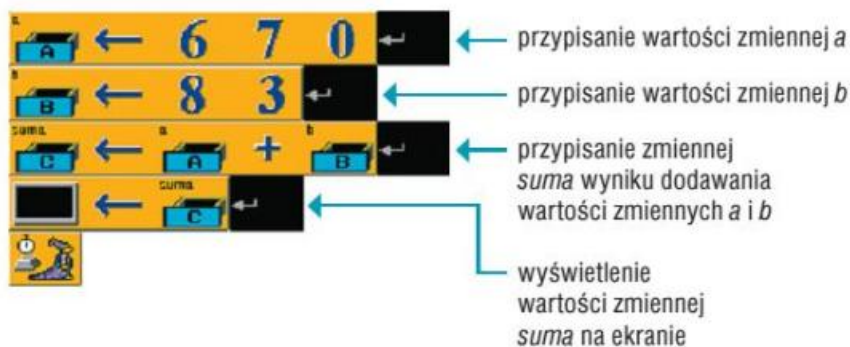
## 4. Zmienne i obliczenia

{ Zamierzamy obliczyć sumę dwóch liczb. W jaki sposób można w środowisku Baltie utworzyć program umożliwiający wprowadzanie danych z klawiatury i wykonanie na nich obliczeń? }



**Aby użyć w programie nowej zmiennej**, możemy wybrać element  (**Nowa zmienna**) z panelu poleceń w trybie **Programowanie/Zaawansowany**. Otworzy się okno dialogowe **Nowa ikona**, w którym wpisujemy nazwę zmiennej, np.  $a$ . Na ekranie pojawi się element  (rys. 11.).

Zmiennej o danej nazwie możemy przypisać wartość, stosując polecenie (**Przydziel**), po którym podajemy konkretną liczbę (rys. 11.) lub element (**Klawiatura**) umożliwiający wprowadzanie danych z klawiatury   . Nazwa nowej zmiennej (np.  $a$ ) widoczna jest w lewym górnym rogu elementu z rysunkiem szuflady oznaczonej literą **A**. W dolnej części okna można śledzić wykonanie programu (po kliknięciu przycisku **Pokaż zmienne**).



**Rys. 11.** Przykład stosowania zmiennych w obliczeniach – ćwiczenie 9.



**Rys. 12.** Wyświetlenie wyniku na ekranie – ćwiczenie 9.




### Ćwiczenie 9. Wykonujemy obliczenia z użyciem zmiennych

1. Przepisz polecenia programu z rysunku 11. i uruchom program.
2. Zmień program tak, aby dane można było wprowadzać z klawiatury po jego uruchomieniu.
3. Dodaj obliczenia różnicy i iloczynu liczb *a* i *b*. Wyniki wyświetlaj w różnych odstępach w siódmym wierszu.
4. Uzupełnij program tak, aby można było wykonać wszystkie obliczenia dla pięciu par liczb. Jaką instrukcję należy zastosować?
5. Przetestuj program dla wybranych danych.
6. Zapisz program w pliku pod nazwą *Obliczenia*.

**Wskazówka:** Aby określić miejsce na ekranie, w którym wyświetlane są wyniki obliczeń,

można użyć elementu **Współrzędne pola**, np.



Więcej elementów umożliwiających korzystanie ze zmiennych zostało umieszczonych w specjalnych bankach, dostępnych po kliknięciu elementu  na pasku narzędzi programu Baltie. Informacje o tym, jak je stosować, można znaleźć w **Pomocy** programu.

## 5. Stosowanie procedur

Zamierzamy utworzyć program, w którym Baltie wybuduje cztery domki – różne, ale z takimi samymi dachami. W jaki sposób można w środowisku Baltie opracować cząstkowy problem (wybudowanie dachu), a potem wielokrotnie wykorzystywać to rozwiązanie w programie?



W środowisku programowania Baltie można definiować procedury – tzw. **pomocników**. Omówimy przykład tworzenia procedury bez parametrów, pozostawiając zainteresowanym uczniom samodzielne zapoznanie się z definiowaniem procedur z parametrami (w tym celu można skorzystać z **Pomocy**).

Po zdefiniowaniu pomocnika możemy go wywoływać w wybranych miejscach programu, bez konieczności zapisywania tych samych operacji od nowa.





**Aby zdefiniować procedurę**, należy:

- kliknąć element **Tablica pomocników**  na pasku narzędzi okna programu Baltie,
- w oknie **Pomocnicy** (rys. 13a) kliknąć **Nowego pomocnika**  (pomocnik pojawi się w obszarze roboczym pod utworzonymi poleceniami),
- wprowadzić polecenia, które ma wykonywać pomocnik.

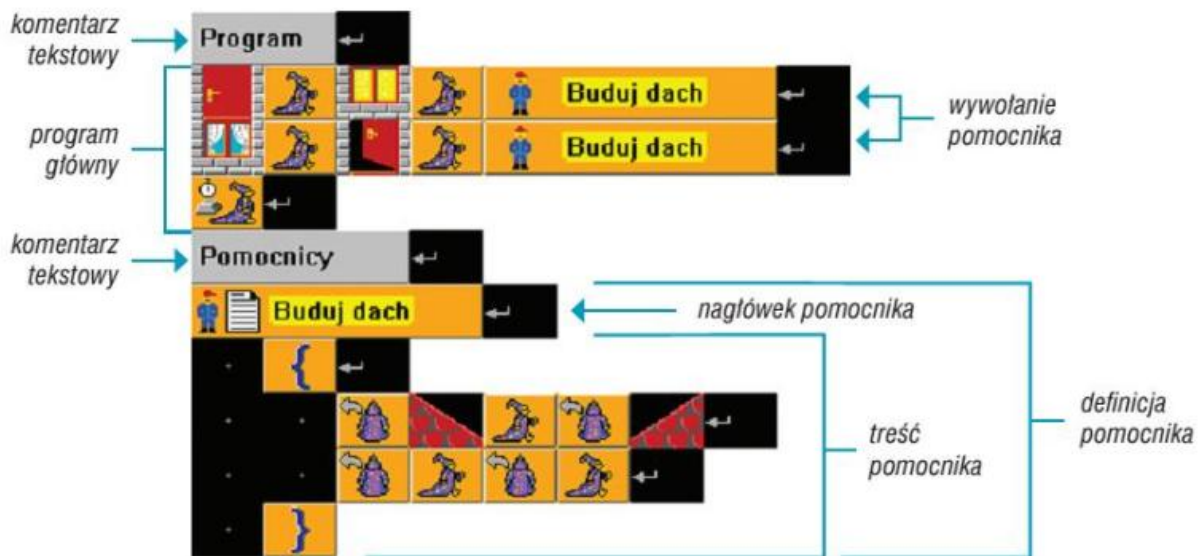
**Aby wywołać procedurę**, należy kliknąć zdefiniowanego pomocnika widocznego w **Tablicy pomocników** (rys. 13b) i umieścić go w odpowiednim miejscu programu (rys. 14.).



Rys. 13a. Okno Pomocnicy



Rys. 13b. Okno Pomocnicy ze zdefiniowanym pomocnikiem



Rys. 14. Stosowanie procedury w programie – ćwiczenie 10.

W programie głównym zbudujemy ściany każdego z domów, a budowanie dachu zlecimy pomocnikowi, definiując (pod programem głównym) odpowiednią procedurę (**Nowego pomocnika**) i wywołując pomocnika w odpowiednim miejscu programu głównego.

W programie Baltie możemy zdefiniować więcej niż jednego pomocnika, dlatego każdy z nich powinien być nazwany. Nazwę można wprowadzić np. w polu tekstowym



, które wybieramy w trybie **Programowanie/Zaawansowany**.



## Ćwiczenie 10. Definiujemy i wywołujemy procedurę

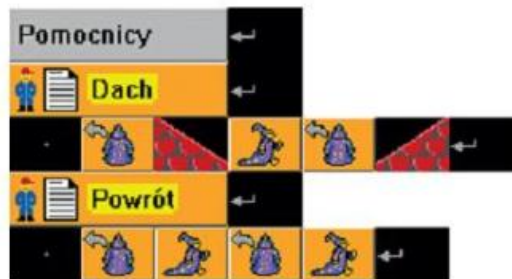
1. Zdefiniuj pomocnika o nazwie *Buduj dach*. Przepisz polecenia programu (rys. 14.), w którym Baltie buduje dwa parterowe domki (bez dachów) i dwukrotnie wywołaj pomocnika w programie. Uruchom program.
2. Uzupełnij program tak, aby Baltie wybudował dwa kolejne domki z takim samymi dachami.
3. Zapisz program pod nazwą *Pomocnicy*.



## Ćwiczenie 11. Stosujemy dwie procedury

1. Otwórz plik *Pomocnicy* zapisany w ćwiczeniu 10. Podziel pracę pomocnika *Buduj dach* na dwie części. W tym celu utwórz dwie procedury: w pierwszej (o nazwie *Dach*) Baltie powinien tylko budować dach, a w drugiej (o nazwie *Powrót*) – zejść z dachu (rys. 15a).
2. Procedurę *Powrót* wywołuj po wybudowaniu dachu na każdym z trzech pierwszych domków (rys. 15b).
3. Zapisz plik pod tą samą nazwą.

Rys. 15a. Definiowanie pomocników – ćwiczenie 11.



Rys. 15b. Wywoływanie procedur w programie – ćwiczenie 11.



Mając zdefiniowane dwie procedury, możemy zdefiniować kolejną, w której będziemy wywoływać procedury zdefiniowane wcześniej.



## Ćwiczenie 12. Wywołujemy procedurę w procedurze

1. Otwórz plik *Pomocnicy* zapisany w ćwiczeniu 11. W obszarze roboczym, pod definicjami pomocników *Dach* i *Powrót*, utwórz procedurę o nazwie *Dach z powrotem*, w której wywołasz dwie wcześniej zdefiniowane procedury (rys. 16a i 16b).
2. Zapisz plik pod tą samą nazwą.

Rys. 16a. Wywoływanie procedur w definicji nowej procedury – ćwiczenie 12.



Rys. 16b. Wywoływanie procedury w programie – ćwiczenie 12.








### Ćwiczenie 13.

1. Otwórz plik *Pomocnicy* zapisany w ćwiczeniu 12. Zmodyfikuj program, zmieniając położenie domków według własnego pomysłu.
2. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Pozycję Baltiego ustalamy, korzystając z elementu **Baltie** , za którym umieszczamy polecenie **Współrzędne pola**. Na rysunku 17. Określono współrzędne  $x = 8$ ,  $y = 9$ . Po każdym określeniu pozycji Baltiego należy wstawić element **Koniec wiersza**.



*Rys. 17. Przykładowe ustalenie pozycji Baltiego – ćwiczenie 13.*



### Warto zapamiętać

- W programie Baltie najpierw wprowadzamy wszystkie polecenia, tworząc w ten sposób program, a dopiero potem program jest wykonywany.
- W programie Baltie polecenia ukryte są pod elementami graficznymi. Istotna jest kolejność ich ułożenia w obszarze roboczym okna, podobnie jak w tradycyjnych językach programowania istotna jest kolejność wpisywanych instrukcji.
- Program Baltie umożliwia wykonywanie zadań, w których występują powtarzanie poleceń i sytuacje warunkowe. Można też tworzyć procedury i korzystać ze zmiennych.



### Pytania i polecenia

1. Wyjaśnij na przykładzie, w jaki sposób pisze się prosty program w środowisku Baltie.
2. Podaj przykład zastosowania pętli zagnieżdżonej w środowisku Baltie.
3. Omów na przykładzie sposób tworzenia w środowisku Baltie programu zawierającego sytuację warunkową.
4. Omów na przykładzie sposób definiowania procedury w środowisku Baltie.



### Zadania

1. Utwórz program, w którym Baltie zbuduje klomb kwiatów.
2. Utwórz program, w którym Baltie zbuduje zamek.
3. Utwórz program, który wyświetli na środku sceny napis POZDROWIENIA. Napis ma zostać obwiedziony ciekawie zaprojektowaną ramką.
4. Utwórz program, w którym Baltie zbuduje jednopiętrowy dom.
5. Zmodyfikuj program z zadania 4. tak, aby Baltie zbudował dom czteropiętrowy.
6. Utwórz program, w którym Baltie zbuduje na środku sceny boisko o wymiarach 5 przedmiotów na 4 przedmioty, a następnie posadzi dookoła niego trawę.
7. Przygotuj zadanie z wykorzystaniem powtarzania poleceń według własnego pomysłu.

- Przepisz polecenia programu (rys. 18.). Uzupełnij program tak, aby Baltie pozbiarał grzyby w lesie, czyli pozamieniał przedmioty z grzybami na przedmioty koloru czarnego (pierwszy przedmiot w banku przedmiotów). Następnie w miejsce przedmiotów koloru czarnego Baltie powinien posadzić choinki. Zapisz program w pliku pod nazwą *Las1*.



Rys. 18. Polecenia programu – zadanie 8.

- Zmodyfikuj program z zadania 8. tak, aby w dolnym wierszu Baltie najpierw wyczarował 13 choinek, a następnie zostały w nim wyświetlone losowo czerwone grzybki. Wyświetlanie grzybów powtarzaj 5 razy. Zapisz program w pliku pod nazwą *Las2*.
- Przepisz polecenia programu z rysunku 19. Uzupełnij program tak, aby Baltie pozbiarał śmieci w lesie, czyli pozamieniał przedmioty inne niż choinki na przedmioty koloru czarnego. Zapisz program w pliku pod nazwą *Śmieci*.



Rys. 19. Polecenia programu – zadanie 10.

- Utwórz program, który umożliwi obliczenie obwodu i pola prostokąta o bokach równych  $a$  i  $b$ , wprowadzanych z klawiatury. Zapisz program w pliku pod nazwą *Obwód i pole*.
- Zmodyfikuj program z zadania 11. tak, aby pole i obwód były liczone dla 10 par liczb  $a$  i  $b$ . Zapisz plik pod tą samą nazwą.
- Korzystając z ćwiczenia 11., utwórz program, w którym Baltie zbuduje trzy piętrowe domki z wykorzystaniem procedury budującej piętro (o nazwie *Buduj piętro*). Każdy domek ma mieć inny dach. Zapisz program w pliku pod nazwą *Trzy domki*.

### Dla zainteresowanych

- Przygotuj zadanie z wykorzystaniem pętli w pętli według własnego pomysłu.
- Utwórz program obliczający sumę  $n$  liczb wprowadzanych z klawiatury.  
**Wskazówka:** Skorzystaj ze schematu blokowego – rys. 4., temat 10.
- Utwórz program, w którym niewidzialny Baltie wypełni całą scenę losowo umieszczanymi czterema różnymi przedmiotami. Umieszczenie kolejnych przedmiotów powtarzaj 100 razy. Ustaw prędkość Baltiego na 7. Włącz **Czarowanie bez chmurki**. Zapisz program pod nazwą *Wygaszacz ekranu*.
- Przygotuj zadanie z zastosowaniem poleceń warunkowych według własnego pomysłu. Zastosuj również losowe umieszczanie przedmiotów.
- Napisz program, w którym Baltie zbuduje cztery garaże w czterech rogach sceny. Garaże mają mieć taką samą przednią ścianę, ale różne dachy. Wykorzystaj możliwość definiowania procedur i zmiany położenia Baltiego.