

## Baltie

### Zadanie 1. Budowanie

W trybie **Budowanie** wybuduj domek jak na rysunku. Przedmioty do wybudowania domku weź z banku 0.



### Zadanie 2. Czarowanie – sterowanie i powtarzanie



W trybie **Czarowanie** z pomocą czarodzieja po lewej stronie domku wybudowanego w zadaniu 1. posadź drzewko, a po prawej ustaw płotek.



**Wskazówki:** Do sterowania Baltiem służą przyciski:



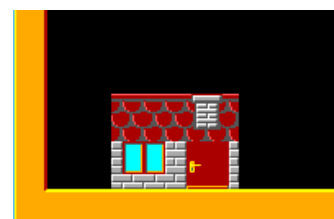
. Czarodziej zawsze wyczarowuje przedmiot przed sobą.

Każdy ostatnio wyczarowany przedmiot pojawia się po prawej stronie przycisku **Wyczaruj z wyborem** jako przycisk **Wyczaruj ostatni**. Jeśli czarodziej ma wyczarować ten sam przedmiot kilka razy, wystarczy kliknąć przycisk **Wyczaruj ostatni**   ,

### Zadanie 3. Programowanie – budowanie domku


Wybierz tryb **Programowanie/Nowicjusz**. Utwórz program, w którym Baltie wybuduje dom jak na rysunku.

**Wskazówki:** Polecenia (m.in. poruszanie się Baltiego) wybieramy z panelu poleceń, a przedmioty z banków przedmiotów.



Pierwszy wiersz programu:

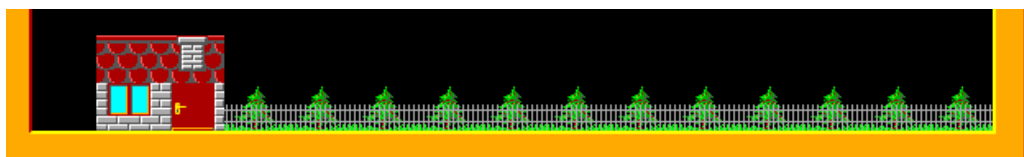


Umieść na końcu programu polecenia **Czekaj**  , aby okienko z programem nie zamknęło się natychmiast po jego wykonaniu.

Aby uruchomić program należy nacisnąć przycisk **Start** .

### Zadanie 4. Programowanie – powtarzanie poleceń

Uzupełnij program z zadania 3. tak, aby Baltie postawił po prawej stronie domu (do końca sceny) płotek z choinkami.



**Wskazówki:** Polecenia, które mają się powtarzać ujmujemy w nawias  , umieszczając

przed nawiasem liczbę powtórzeń, np.



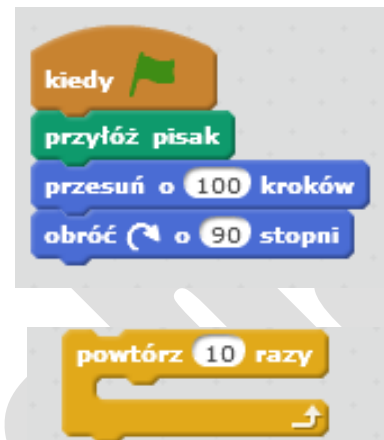
## Scratch

### Zadanie 1. Powtarzanie poleceń

Napisz program rysujący sześciokąt o boku 100 kroków.

#### Wskazówki:

Na początku skryptu umieść element **kiedy** (z grupy **Zdarzenia**), a potem odpowiednie polecenia:



Zastosuj polecenie **powtórz** (z grupy **Kontrola**):

W polu tekstowym polecenia domyślnie umieszczona jest liczba 10; należy wpisać odpowiednią liczbę powtórzeń.

### Zadanie 2. Stosowanie zmiennych i instrukcji warunkowej

Napisz program, który umożliwi wprowadzenie dwóch różnych liczb z klawiatury i wyprowadzenie większej z nich.

#### Wskazówki:

Zadeklaruj dwie zmienne: *liczba1*, *liczba2*.

Aby zadeklarować użycie w programie nowej zmiennej, należy wybrać element **Utwórz zmienną** (z grupy **Dane**) – otworzy się okno dialogowe, w którym wpisujemy nazwę zmiennej. Utworzone zmienne będą widoczne w panelu poleceń.

Zmiennej o danej nazwie możemy przypisać wartość, stosując polecenie

**ustaw** *liczba1* na **0**. W polu za słowem **na** możemy podać konkretną wartość lub

umieścić element **odpowiedź** (z grupy **Czujniki**), umożliwiający wprowadzanie danych z klawiatury.



Przed poleceniem **ustaw** należy dodać polecenie **zapytaj**.

Wyrażenia logiczne i arytmetyczne wybieramy z grupy **Wyrażenia**.

W zadaniu zakładamy, że wprowadzane liczby są różne. Aby sprawdzić, która liczba jest większa i wyprowadzić właściwy wynik, stosujemy instrukcję warunkową.

### Zadanie 3. Stosowanie zmiennych, instrukcji warunkowej i instrukcji powtarzania

Zmodyfikuj program z zadania 2 tak, aby można było sprawdzić w ten sposób dziesięć par różnych liczb. Zastosuj polecenie powtarzania. Zastanów się, które polecenia powinny być umieszczone wewnątrz elementu **powtórz**.

## C++

### Podstawowa struktura programu w języku C++

```
#include <iostream>
using namespace std;
...           // deklaracje stałych
...           // deklaracje funkcji
int main ()
{
    return 0; // część wykonawcza programu
}
```

#### Zadanie 1. Wyprowadzanie danych

Napisz program wyprowadzający na ekran napis: „jestem na konferencji”. Zapisz program pod nazwą *zadanie1*.

**Wskazówki:** Prosty program w języku C++ ma postać:

```
Zadanie1.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Jestem na konferencji";
8      return 0;
9  }
```

W języku C++ `cout` jest obiektem reprezentującym standardowe wyjście.

#### Zadanie 2. Deklarowanie zmiennych, wprowadzanie danych, wykonywanie obliczeń i wyprowadzanie wyników na ekran

Napisz program umożliwiający wprowadzenie dwóch liczb całkowitych *a* i *b*, obliczenie ich sumy (*suma*) i wyprowadzenie wyniku (*suma*) na ekran monitora. Skompiluj i uruchom program. Zapisz program pod nazwą *zadanie2*.

**Wskazówki:**

W pokazanym fragmencie programu zadeklarowano trzy zmienne typu całkowitego:

```
Zadanie2.cpp
1  #include <iostream>
2
3  using namespace std;
4  int a,b,suma;
5  int main()
6  {
```

Przykład wprowadzania danych: `cin >> a >> b;`

W języku C++ `cin` jest obiektem reprezentującym standardowe wejście.

W instrukcji przypisania używamy znaku „=” np. `iloczyn=a*b;`

### Zadanie 3. Dodawanie opisów

Do programu z zadania 2. dodaj opisy „wprowadz pierwsza liczbe”, „wprowadz druga liczbe”, „suma wynosi”.

**Wskazówki:** Umieszczenie `endl` lub znaków `"\n"` na końcu polecenia `cout` zaczyna nowy wiersz, np.:

```
cout << "wprowadz pierwsza liczbe:" << "\n";
```

### Zadanie 4. Stosowanie instrukcji warunkowej

Napisz program, który umożliwi wprowadzenie z klawiatury dwóch różnych liczb całkowitych (*liczba1*, *liczba2*) i wyprowadzenie na ekran większej z nich.

**Wskazówki:** Instrukcja warunkowa w języku C++ ma postać:

```
if (wyrażenie)
    instrukcja1;
else instrukcja2;
```

lub

```
if (wyrażenie) instrukcja1;
```

Na przykład:

```
13     if(liczba1>liczba2)
14         cout << "wieksza jest: " << liczba1;
15     else cout << "wieksza jest: " << liczba2;
16
```

### Zadanie 5. Stosowanie instrukcji iteracyjnej-for

Zmodyfikuj program z zadania 4, aby można było sprawdzić dziesięć par różnych liczb.

**Wskazówki:** Instrukcja iteracyjna w języku C++ ma postać:

```
for (wyrażenie_początkowe; warunek; wyrażenie_pętli) instrukcja;
```

Fragment programu z zastosowaniem instrukcji `for`:

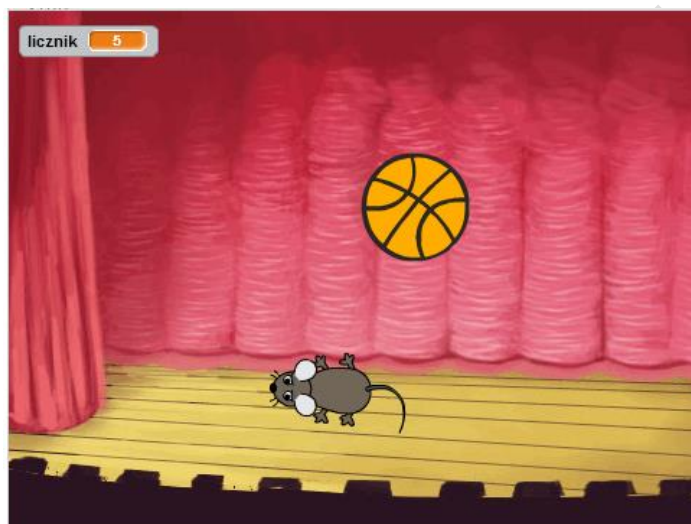
```
9     for (i=0;i<10;i++)
10     {
11         cout << "Podaj pierwsza liczbe: ";
12         cin >> liczba1;
13         cout << "Podaj druga liczbe: ";
14         cin >> liczba2;
15         if(liczba1>liczba2)
16             cout << "wieksza jest: " << liczba1 << "\n";
17         else cout << "wieksza jest: " << liczba2 << "\n";
18     }
```

## Zadanie dla zainteresowanych

### Gra z dwoma duszkami

Utwórz grę, w której udział wezmą dwa duszki (wzór na rysunku). Jeden z duszków ma poruszać się losowo po całej scenie. Drugim duszkiem będziemy sterować, naciskając klawisze strzałek. Jeśli duszki dotkną się otrzymujemy 1 punkt. Gra ma kończyć się, gdy otrzymamy 5 punktów. Na koniec powinien wyświetlić się komunikat „Wygrałeś!”.

Uwaga: Dla każdego duszka układamy oddzielny skrypt.



### Zmiana postaci i dodawanie tła

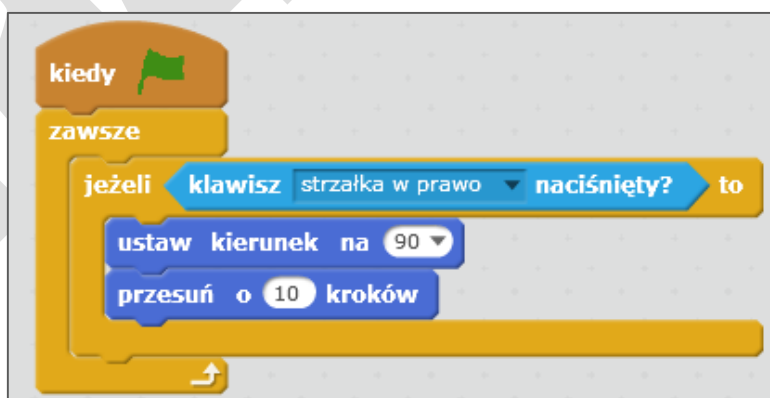
Nowego duszka lub nowe tło można narysować samodzielnie lub wybrać gotowe.

Usuń duszka-kotka ze sceny. Postać nowego duszka (np. myszkę) wybierz, klikając

Nowy duszek:  . Nowe tło – klikając:  .

### Sterowanie duszkiem-(instrukcja **jeżeli**)

- Na rysunku pokazany jest fragment skryptu z zastosowaniem polecenia **jeżeli** – polecenia po słowie **to** będą wykonane, gdy warunek będzie spełniony.



- Umieść wszystkie polecenia **jeżeli** (do sterowania duszkiem w pozostałych kierunkach) wewnątrz polecenia **zawsze**.  
**Uwaga:** Pamiętaj o możliwości duplikowania poleceń (kliknij prawym klawiszem myszy dane polecenie lub grupę poleceń i z menu kontekstowego wybierz polecenie **duplikuj**).

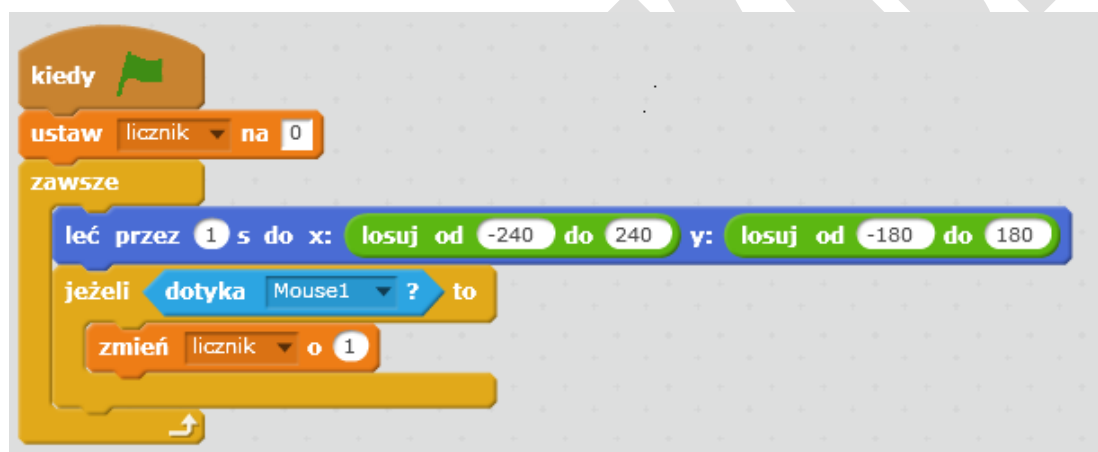
### Dodawanie drugiego duszka, losowe przemieszczanie duszka po scenie

- Dodaj drugiego duszka (np. piłkę). Duszek powinien losowo przemieszczać się po scenie.
- Aby duszek przemieszczał się w losowy sposób po scenie, użyj polecenia:

, ale zamiast konkretnych wartości x, y podaj losowe liczby , gdzie zamiast 1 i 10 podaj odpowiednie wartości współrzędnych.

### Zliczanie punktów zależnie od warunku (zastosowanie instrukcji jeżeli)

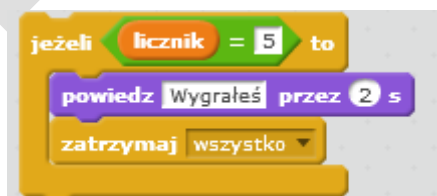
- Aby zliczać punkty, gdy duszki dotkną się, należy do polecenia warunkowego wstawić warunek: i wybrać danego duszka.
- Aby punkty w grze były zliczane od zera, należy wyzerować licznik: .
- Następnie należy dodać zliczanie punktów: , co odpowiada poleceniu:  $licznik := licznik + 1$ .



Tak powinien wyglądać program dla duszka-piłki.

### Dodawanie warunków zakończenia programu (zastosowanie instrukcji jeżeli)

- W pętli **zawsze** umieść jeszcze instrukcję warunkową, która w przypadku zdobycia wymaganej liczby punktów wyświetla napis „Wygrałeś” i kończy program.



Omawiane środowiska programowania są dostępne bezpłatnie w Internecie:

- *Baltie 3* <http://www.baltie.com/pl/>
- *Scratch 2.0* <https://scratch.mit.edu/scratch2download/>
- *Kompilator Dev-C++* <https://sourceforge.net/projects/orwelldevcpp/>