

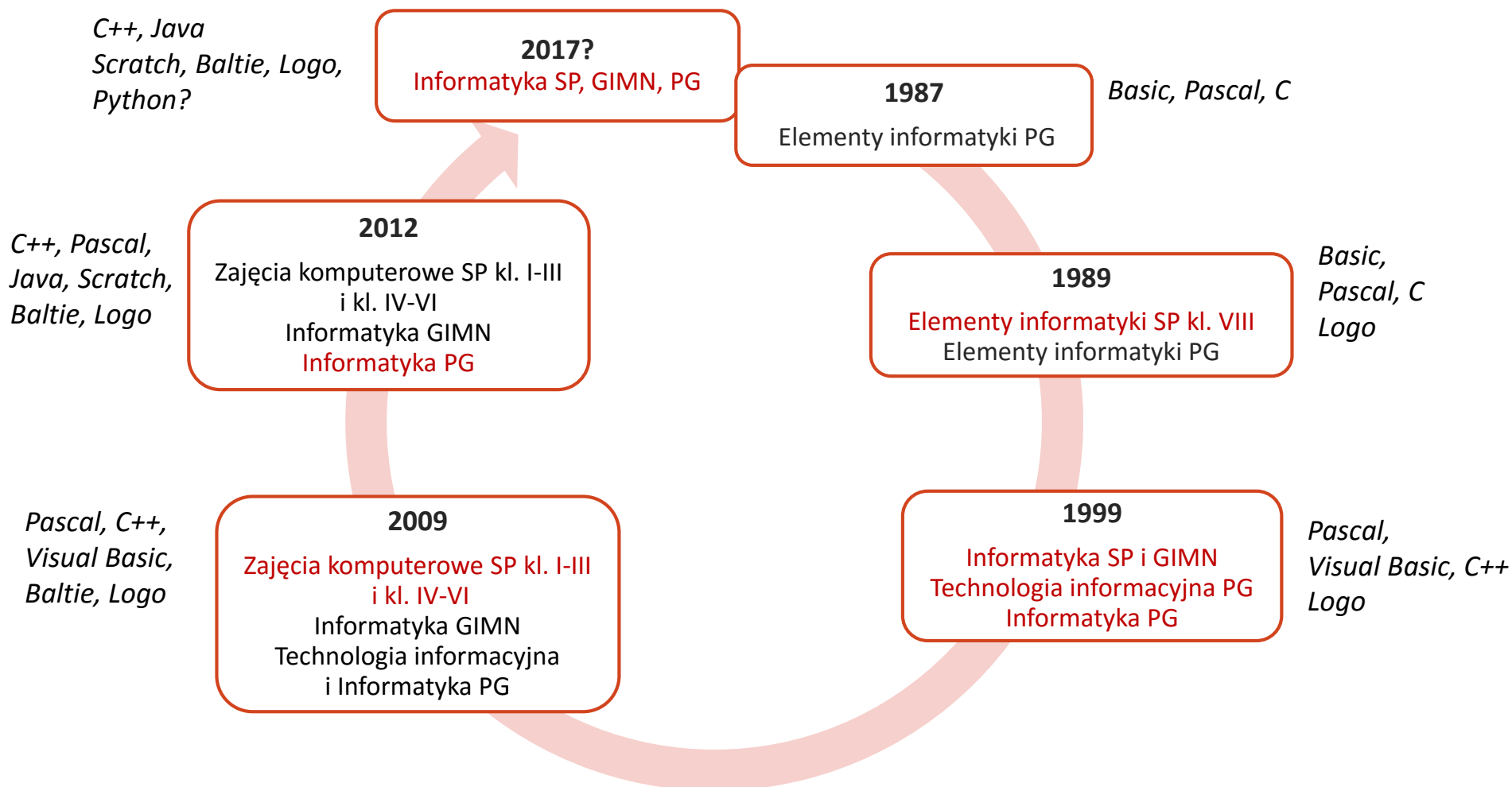
Programowanie – od pierwszoklasisty do maturzysty

Grażyna Koba

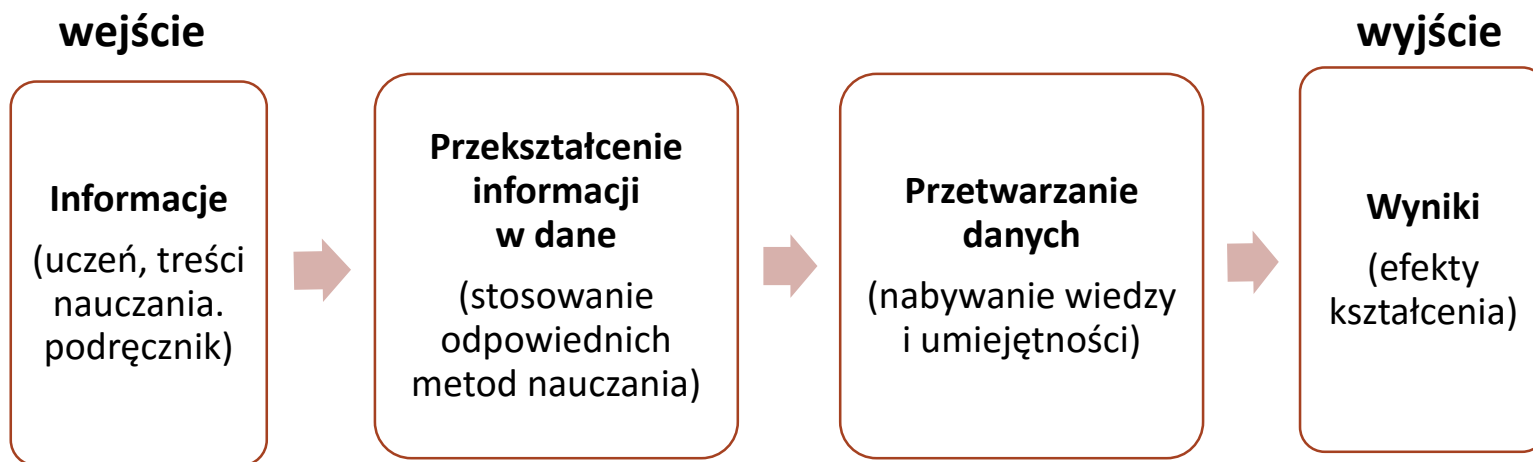


„Kąg trzydziestolecia” nauki programowania **MiGra**

PODRECZNIKI DO INFORMATYKI



Aby na wyjściu otrzymać dobre **wyniki** musimy nie tylko wprowadzić właściwe **informacje**, ale właściwie je **przetwarzać** (= stosować dobre metody nauczania).



Przykład z podstawy programowej

– informatyka PG (zakres rozszerzony):

*Uczeń opisuje i stosuje algorytm, np. wydawania reszty metodą **zachłanną**.*



Algorytm zachłanny – algorytm, który w celu wyznaczenia rozwiązania w każdym kroku dokonuje zachłannego, tj. najkorzystniejszego w danym kroku, rozwiązania częściowego.

Krok 1.

Największy nominał, nie większy niż R to 50 zł.

$69 \text{ div } 50 = 1$ wypłacamy: **50 zł.**

$R := 69 - 50 \cdot 1 = 19$ do wydania zostało 19 zł.

Krok 2.

Największy nominał, nie większy niż aktualne R to 10 zł.

$19 \text{ div } 10 = 1$ wypłacamy **10 zł.**

$R := 19 - 10 \cdot 1 = 9$ do wydania zostało 9 zł.

Krok 3.

Największy nominał, nie większy niż aktualne R to 5 zł.

$9 \text{ div } 5 = 1$ wypłacamy **5 zł.**

$R := 9 - 5 \cdot 1 = 4$ do wydania zostały 4 zł.

Krok 4.

Największy nominał, nie większy niż aktualne R to 2 zł.

$4 \text{ div } 2 = 2$ wypłacamy dwie monety **2 zł.**

$R := 4 - 2 \cdot 2 = 0$ wypłaciliśmy całą kwotę.

Reszta została wydana następująco: **$1 \cdot 50 \text{ zł} + 1 \cdot 10 \text{ zł} + 1 \cdot 5 \text{ zł} + 2 \cdot 2 \text{ zł} = 69 \text{ zł}.$**





Czego trzeba nauczyć?

Struktury programu

Stosowania zmiennych,
w tym – tablic

Definiowania funkcji

Instrukcji iteracyjnej **for**

```
#include <iostream>

using namespace std;

int nominaly[100];
int liczbaNominalow;

void wczytajNominaly()
{
    cout << "Podaj liczbe nominalow: ";
    cin >> liczbaNominalow;

    cout << "Podaj nominaly (w kolejnosci malejacej!): " << endl;
    for(int i = 0; i < liczbaNominalow; ++i)
    {
        int nominal;
        cin >> nominal;

        nominaly[i] = nominal;
    }
}
```

Czego trzeba nauczyć?



```
void znajdzReszte(int reszta)
{
    cout << "Wyplacone nominały: " << endl;
    int aktualnyNominal = 0;
    while( reszta > 0 && aktualnyNominal < liczbaNominalow )
    {
        while( aktualnyNominal < liczbaNominalow && reszta < nominały[aktualnyNominal] )
            ++aktualnyNominal;

        if( aktualnyNominal < liczbaNominalow && reszta >= nominały[aktualnyNominal] )
        {
            int liczbaBanknotow = reszta/nominały[aktualnyNominal];
            cout << liczbaBanknotow << "x" << nominały[aktualnyNominal] << "zł" << endl;
            reszta -= liczbaBanknotow*nominały[aktualnyNominal];
        }
    }

    if( reszta > 0 )
        cout << "Brak nominałów do wydania pozostałej kwoty: " << reszta << "zł" << endl;
}
```

Instrukcji
iteracyjnej **while**

Instrukcji
warunkowej **if**

Czego trzeba nauczyć?

Wywoływania funkcji

Wprowadzania danych

Wyprowadzania
wyników

```
int main()
{
    wczytajNominaly();

    int cena;
    int wpłata;
    cout << "Podaj cenę: ";
    cin >> cena;

    do
    {
        cout << "Podaj wpłatę: ";
        cin >> wpłata;
    } while( wpłata < cena );

    znajdźResztę(wpłata - cena);

    return 0;
}
```

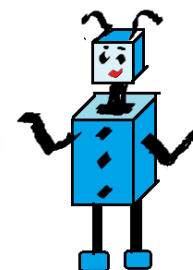


A przede wszystkim MYŚLENIA!



**Praca z komputerem to praca z programami komputerowymi.
Każdy program trzeba uruchomić.**

- zaczynamy od sterowania postacią na ekranie, czyli uczymy wykonywania poleceń według listy kroków



Projekt nowej podstawy programowej:

<https://men.gov.pl/strony/projekt-nowej-podstawy-programowej-ksztalcenia-informatycznego-2.ht>



Czarowanie

1. Obróć się w lewo
2. Idź krok do przodu
3. Obróć się w prawo
4. Idź 2 kroki
5. itd.



Programowanie

Utwórz program, w którym Baltye wyczaruje łączkę z czterema żółtymi kwiatkami i na niej wybuduje dom jak na rysunku.

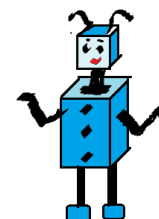




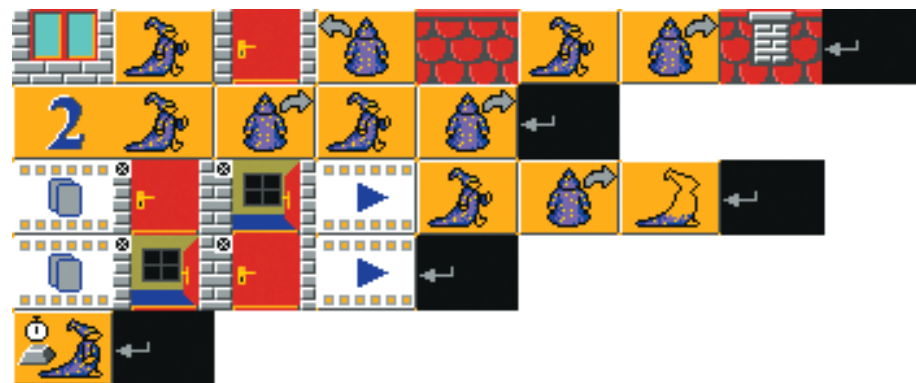
- systematyzujemy pojęcia i rozszerzamy wiedzę i umiejętności dotyczące programowania

W obowiązującej podstawie:

Uczeń: za pomocą ciągu poleceń tworzy proste motywy lub steruje obiektem na ekranie;



- nazywamy metody i definiujemy ważniejsze pojęcia, m.in.:
***program** jako ułożone w odpowiedniej kolejności polecenia dla komputera*
- uczniowie tworzą trudniejsze programy, m.in.:
stosują **złożone animacje**,
współrzędne ekranu



- uświadamiamy uczniom, że...**komputer wykonuje programy komputerowe**

- uczniowie poznają język tekstowy – interpretowany
- definiują i wywołują procedury (bez parametrów)

oto **PROSTOKĄT**

ukp "czerwony

ugp "2

powtórz 4 [np 50 pw 90 np 100 pw 90]

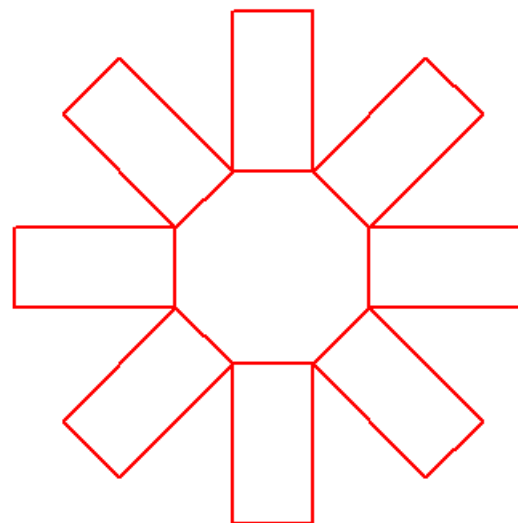
już



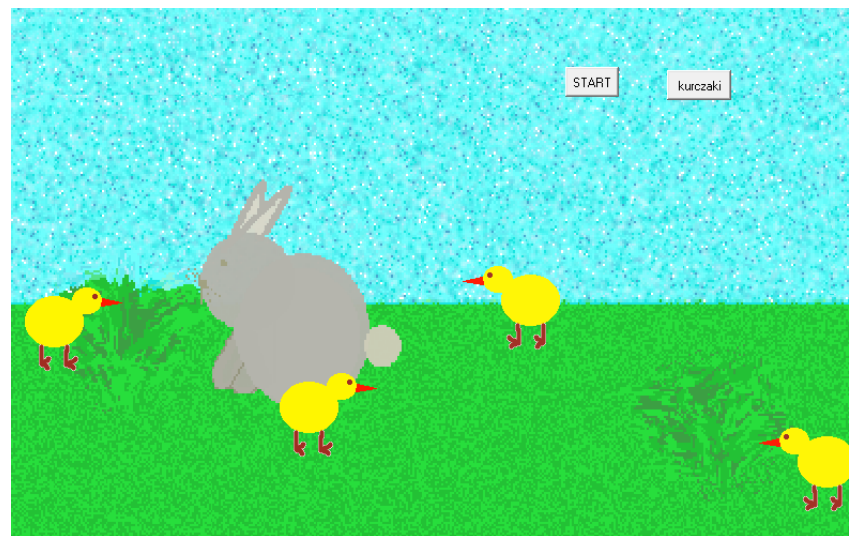
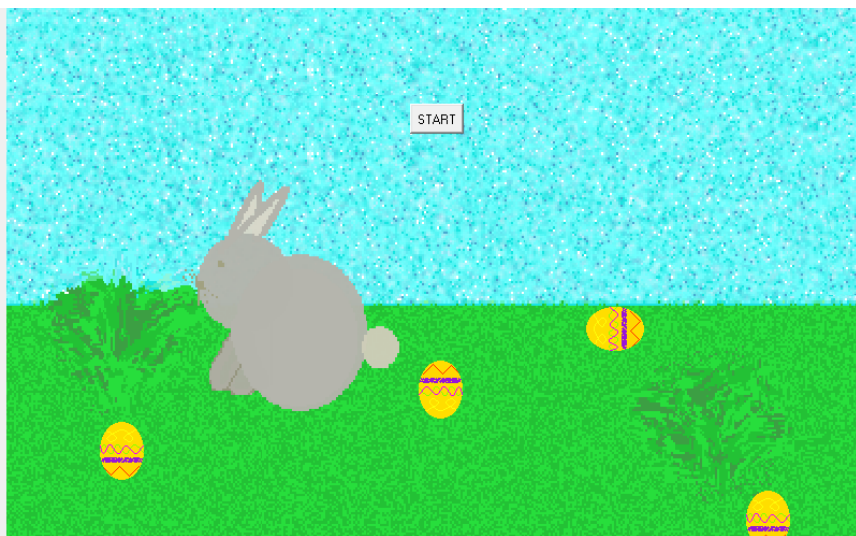
oto FIGURA

powtórz 8 [**PROSTOKĄT** np 50 lw 45]

już

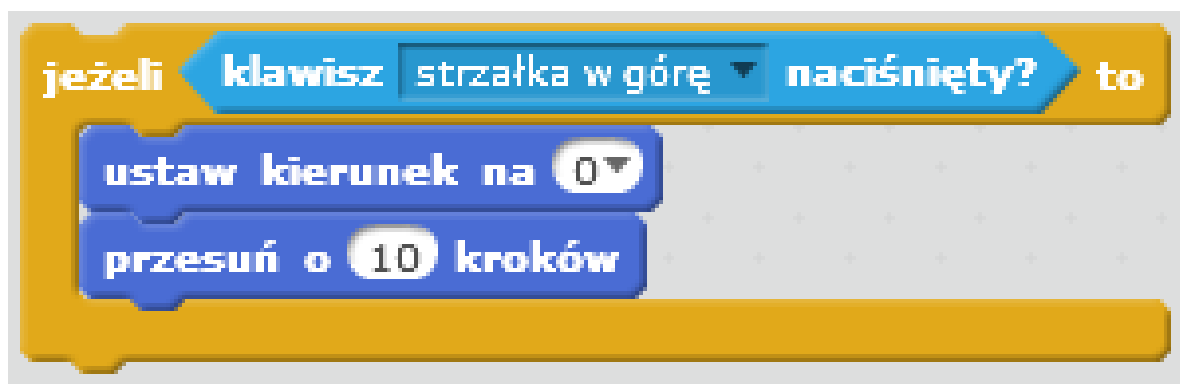
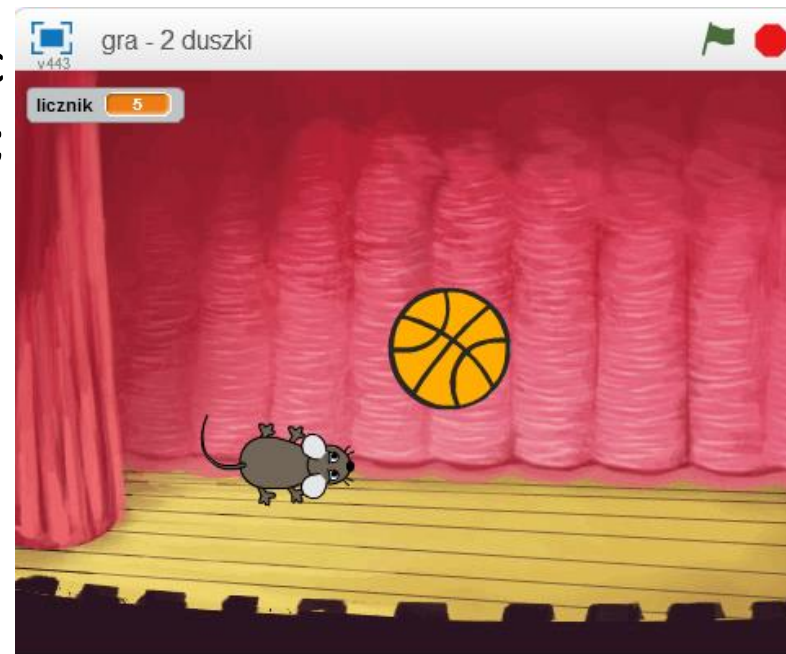


- uczniowie tworzą projekty, m.in.: stosują polecenie powtarzania, zmieniają postać, umieszczają przyciski na ekranie, czyli... obsługują **zdarzenia**



... tworzymy gry w Scratchu

- uczniowie tworzą proste gry, stosując instrukcje: warunkową i powtarzania; pojawia się też zmienna (licznik) i wyrażenie logiczne





W obowiązującej podstawie:

Uczeń:

- *stosuje podejście algorytmiczne w rozwiązywaniu problemów*
- *opisuje sposób znajdowania wybranego elementu w zbiorze nieuporządkowanym i uporządkowanym,*
- *opisuje algorytm porządkowania zbioru elementów*
- *wykonuje wybrane algorytmy za pomocą komputera*



- uczniowie m.in. stosują algorytm wyboru największego elementu do posortowania elementów w porządku malejącym



przed uporządkowaniem



po uporządkowaniu

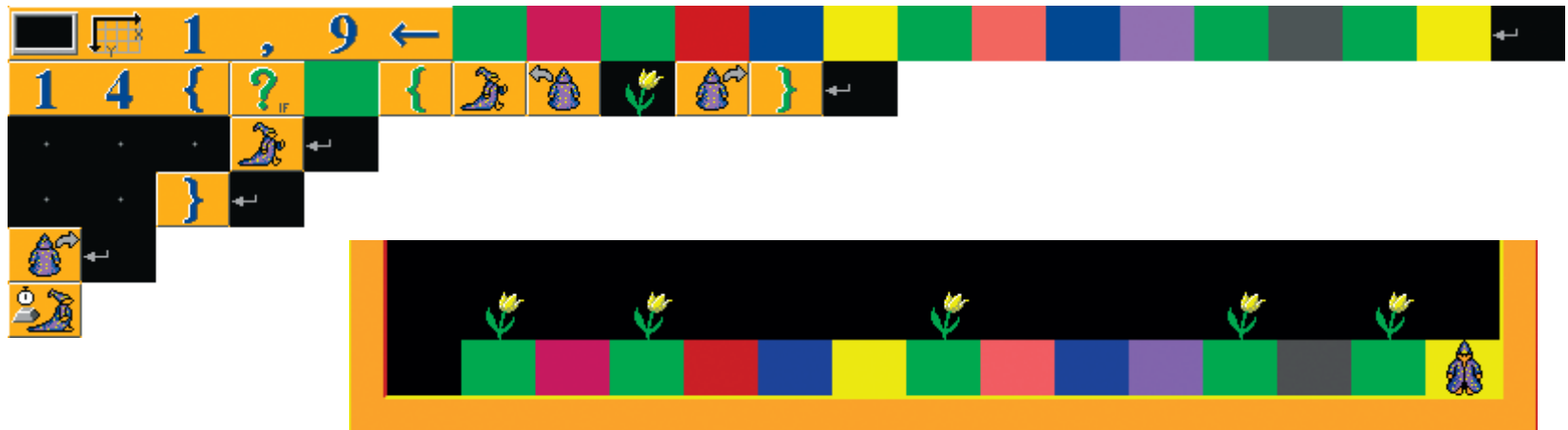


Przykład sortowania przez wybór

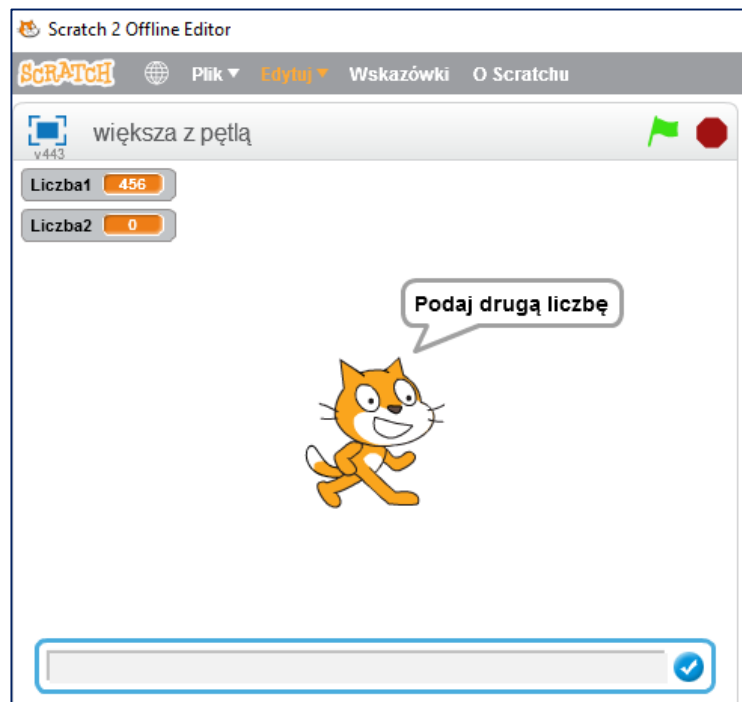
- uczniowie stosują instrukcje warunkową i iteracyjną – Baltie

Analiza rozwiązania

Odpowiedz na pytania: *Jakie polecenie jest wykonywane, gdy warunek jest spełniony, a jakie, gdy nie jest spełniony? Ile razy będzie wykonana instrukcja warunkowa?*

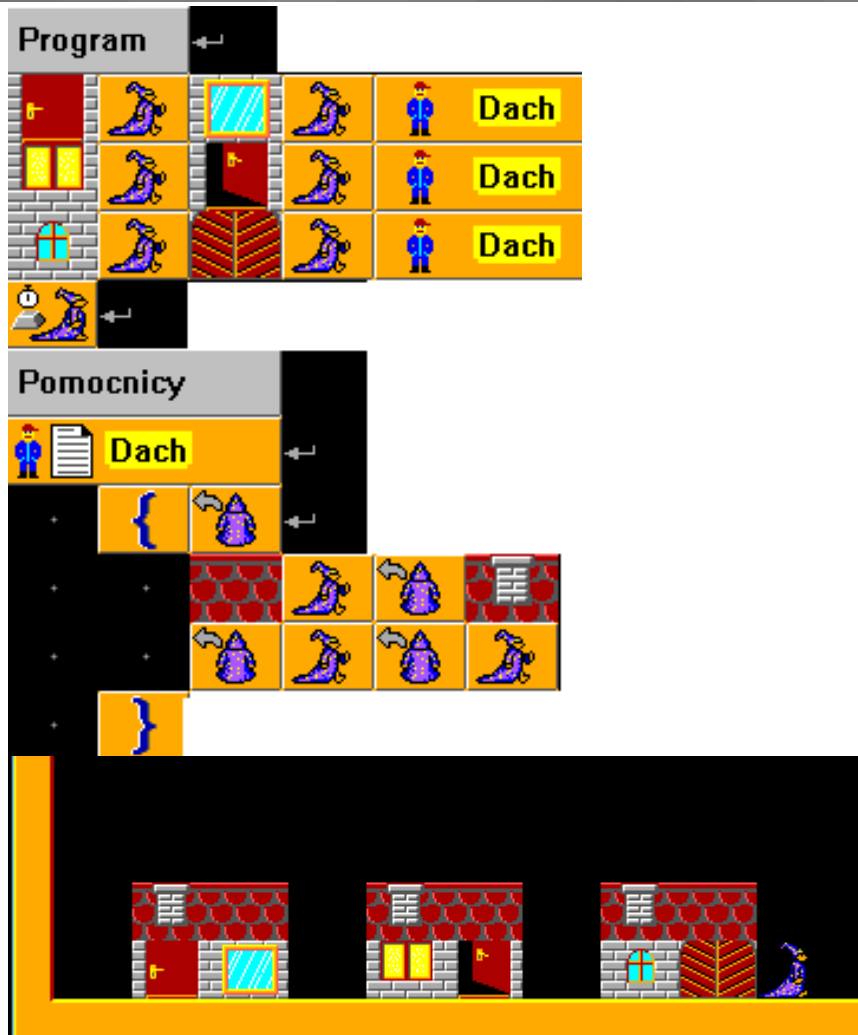


- uczniowie stosują zmienne, wykonują na nich obliczenia, wprowadzają dane z klawiatury, wyprowadzają wyniki na ekran, stosują instrukcje powtarzania i warunkową - Scratch



... stosujemy procedury, w tym z parametrami **MiGra**

PODRECZNIKI DO INFORMATYKI



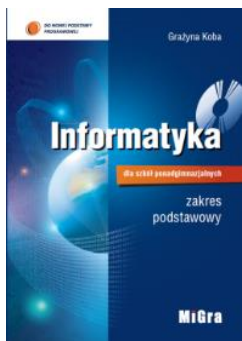
Zakres podstawowy



W obowiązującej podstawie:

Uczeń: [...] stosuje podejścia algorytmicznego:

- *prowadzi dyskusje nad sytuacjami problemowymi;*
- *formułuje specyfikacje dla wybranych sytuacji problemowych;*
- *projektuje rozwiązanie,*
- *realizuje rozwiązanie na komputerze za pomocą oprogramowania aplikacyjnego lub języka programowania;*
- *testuje otrzymane rozwiązanie,*



- uczniowie poznają podstawy wybranego języka wysokiego poziomu, np. C++



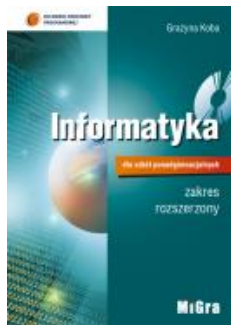
W języku C++ instrukcja warunkowa ma postać:

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



W języku C++ **instrukcja iteracyjna** ma postać:

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



... realizują wybrane algorytmy, stosując poznane instrukcje, np.:

- warunkowe,
- iteracyjne,

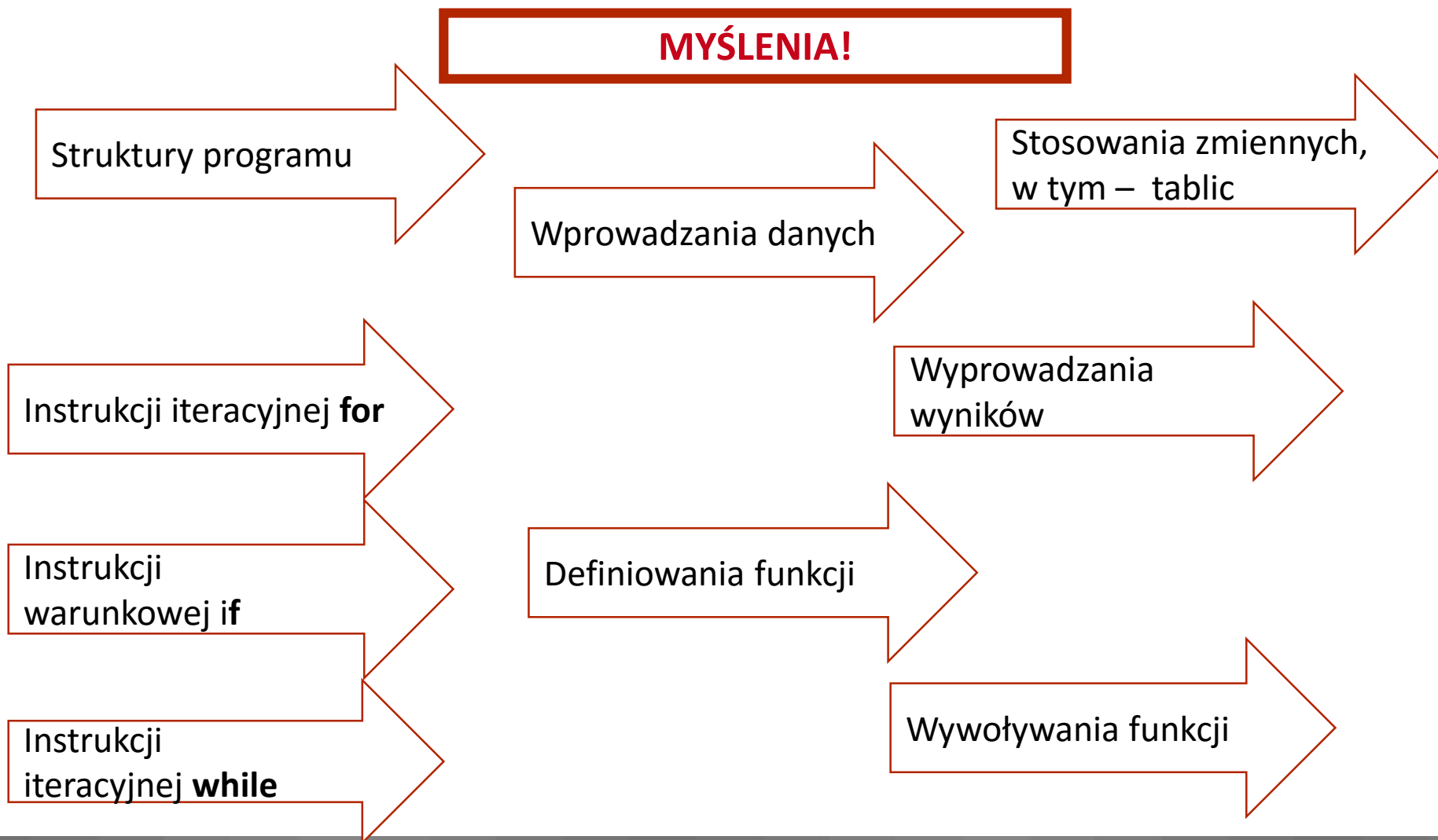
C++

```
#include <iostream>
using namespace std;
int main()
{
    float x, w;
    cin >> x;
    if (x > 0) w = x; else w = -x;
    cout << "Wartość bezwzględna liczby x = " << w << endl;
    return 0;
}
```

C++

```
#include <iostream>
using namespace std;
int main()
{
    int i, a, iloczyn, n;
    cin >> n;
    iloczyn=1;
    for (i=0; i<n; i++)
    {
        cin >> a;
        iloczyn*=a;
    }
    cout << iloczyn;
    return 0;
}
```

Czego mieliśmy nauczyć? Czy się udało?



- Rozumienie, że programy wykonuje komputer, a nie czarodziej czy kotek.
- Rozumienie, że postać instrukcji musi być zgodna z zasadami składni danego języka – dotyczy również wyboru odpowiedniego elementu graficznego.
- Rozumienie, że kolejność występowania instrukcji w programie (także prezentowanych przez elementy graficzne) powinna odpowiadać kolejności operacji realizujących dany algorytm.
- Wykonywanie wielu zadań nawet podobnych na stosowanie poszczególnych instrukcji i zasad programowania.
- Umiejętność zastosowania poznanej zasady w innym zadaniu.
- Na samodzielną pracę.

A wszystko to będzie możliwe,
gdy uczeń ma swój podręcznik,
który zawiera rzetelne,
dostosowane dla niego, źródło
wiedzy!



Dziękuję za uwagę

Grażyna Koba
grazyna.koba@migra.pl