























Zasady budowania algorytmów z klocków

Początek pracy

Otwieramy nowy projekt *Plik/Nowy projekt*, a następnie planszę *Plik/Plansza/Nowa*, na której będziemy budowali algorytmy. Po lewej stronie widzimy paletę klocków (również z liniami)

Klocki Podstawowe

	Początek algorytmu		Wywołanie procedury
	Koniec algorytmu		Początek procedury
	Wykonanie obliczeń		Koniec procedury
	Sprawdzenie warunku		Deklaracja zmiennych globalnych
	Wprowadzenie danej		Start zegara
	Wyprowadzenie wyniku		Oczekiwanie na zegar
	Odczyt z tablicy		Wywołanie konstruktora
	Zapis do tablicy		Wywołanie metody
	Odczyt z taśmy		Początek konstruktora
	Zapis na taśmę		Początek destruktora
	Przewinięcie taśmy		Deklaracja zmiennych globalnych
	Przesunięcie wskaźnika na określoną pozycję		Początek metody
	Sygnał dźwiękowy		Wyświetlenie filmu
	Przepisanie z tablicy na taśmę		Odegranie muzyki
	Przepisanie z taśmy do tablicy		

Układanie Klocków

Za pomocą myszy można **podnieść klocek i położyć na planszy**. Przeniesienie klocka **poza górną krawędź planszy lub naciśnięcie Delete** (przy zaznaczonym klocku) powoduje jego **skasowanie**.

Edycja klocka

Większość klocków ma wewnętrzne pola, które trzeba wypełnić, żeby określić co będzie się działo podczas wykonywania klocka. Wskazujemy klocek, **naciskamy prawy przycisk myszy i otwieramy okno dialogowe** charakterystyczne dla danego klocka.

Paleta klocków

Podstawową paletę klocków można rozszerzyć o dwa zestawy wywołane z menu

Opcje/Paleta klocków

Dodatkowe – pozwalające korzystać z procedur i obiektów

Interfejsowe – obsługujące czujniki i elementy wykonawcze podłączone do interfejsu.

Budowanie algorytmu

Każdy algorytm musi mieć punkt startu i to dokładnie jeden. Jest nim **klocek startowy**. Powinien też mieć jeden klocek, na którym **kończy się działanie algorytmu**. Nie może być **pustych miejsc między klockami**.

Utworzony algorytm można zapisać *Plik/Plansza/Zachowaj...*

Tworzenie procedur

W zestawie klocków **dodatkowe** znajdują się klocki obsługujące procedury.

Można zbudować algorytm, który będzie się składał z kilku plansz zawierających procedury. Procedura musi mieć klocek **początek procedury** i kończyć na klocku **koniec procedury**. Lepiej żeby procedura była na osobnej planszy. Aby procedura była funkcją i zwracała wartość, w klocku wywołującym ją trzeba wypełnić pole nazwa zmiennej. Posługiwanie się procedurami jest bardzo wygodne gdyż skraca algorytm oraz łatwiejszy staje się proces poprawiania błędów. Utworzone algorytmy procedur można wykorzystywać wielokrotnie. Plansza głównego algorytmu i plansze procedur tworzą **projekt**. Jest on zapisywany *Plik/Zachowaj projekt*.

Wykonanie algorytmu

Algorytm ułożony na planszy można wykonać. Wybieramy *Wykonanie/Uruchom* (lub klawisze CTRL+F9) - powoduje to oznaczenie **czerwoną obwódką klocka startowego**. Mamy teraz możliwość wykonania algorytmu w sposób ciągły (*Wykonanie/Wykonaj* lub F9) albo krokowo (*Wykonanie /Krok* lub F8).

Obliczenia w algorytmie, funkcje standardowe

Często używany w algorytmach **kłosek wykonania** obliczeń może zawierać 32 linie tekstu (przejdźcie do nowej linii CTRL+Enter) Każda linia stanowi oddzielną operację. Linie mają postać ***nazwa_zmiennnej := wyrażenie***

Funkcja to ciąg znaków i po niej argument w okrągłych nawiasach, np. Sin(alfa). Możemy skorzystać z funkcji standardowych.

Sprawdzanie warunku – rozgałęzienia algorytmu

W klocku **Sprawdzanie warunku** wpisujemy **Wyrażenie Relacja Wyrażenie**

W zależności od wyboru języka w menu *Opcje/Srodowisko* z takich relacji korzystamy.

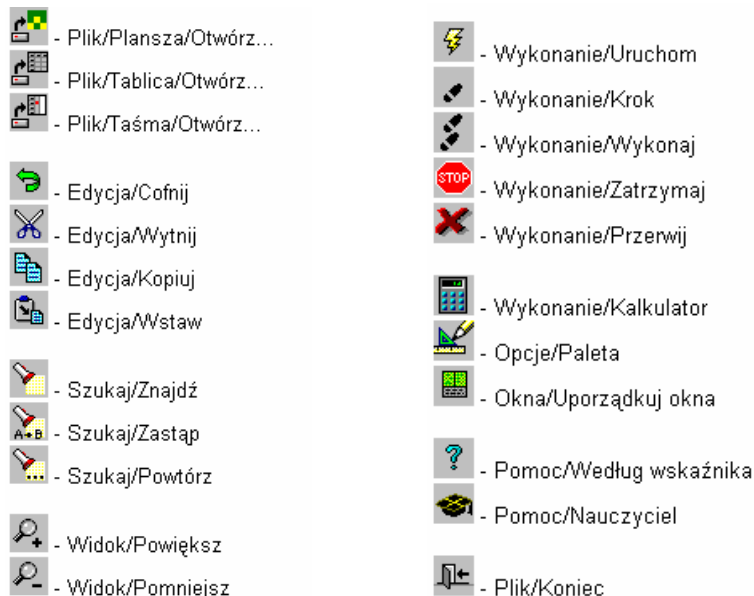
Wprowadzanie danych i wyprowadzanie wyników

Najprostszym sposobem jest wprowadzanie danych i wyprowadzanie wyników za pomocą klocków **Wprowadzanie danej; Wyprowadzanie wyniku**.

Do wprowadzania, przechowywania i wyprowadzania większej liczby danych służą dwie struktury: **tablica** i **taśma** (odpowiednik pliku).

Pasek Narzędzi

Pasek narzędzi umieszczony na górze ekranu umożliwia szybkie wywołanie poleceń.



Okna

W trakcie wykonywania algorytmu mogą być otwarte okna pozwalające podglądać.

Okno zmienne

Otwierając okno **Zmienne** możemy uzyskać przegląd zmiennych i ich wartości (Polecenie *Widok/zmienne* lub przycisk po prawej stronie ekranu). Wartości zmiennych można modyfikować.

Okno Ślad

W oknie **Ślad** (*Widok/Ślad* lub przycisk po prawej stronie ekranu) uzyskujemy dokładną relację z przebiegu wykonania algorytmu.

Okno Komentarz

Okno **Komentarz** (*Widok/Komentarz* lub przycisk po prawej stronie ekranu) jest dodatkowym opisem algorytmu, który może być dołączony do każdego klocka.

Okno Pułapki

W oknie **Pułapki** (*Widok /Pułapki*) można uzyskać informacje o pułapkach ustawionych na klockach

Okno Projekt

W oknie **Projekt** (*Widok/Projekt*) jest wyświetlana lista okien należących do projektu: plansze, tablica, taśma, które są pamiętane w oddzielnych plikach

• Wprowadzenie do budowania algorytmów

□ **Algorytm liniowy - obliczanie sześcianu liczby**

Zbudujemy algorytm, który będzie *obliczał trzecią potęgę podanej liczby*. Plan działań wygląda następująco:

1. start
2. wczytanie liczby
3. obliczenie jej trzeciej potęgi
4. wyprowadzenie wyniku.
5. zakończenie

Ćwiczenie 1.

Proszę narysować **schemat blokowy** algorytmu na kartce, a następnie **zbudować go w programie ELI**, korzystając z poniższej instrukcji.

1. Każdy algorytm musi mieć punkt startu (tylko jeden). Jest nim klocek **Początek algorytmu**. Wstawmy go na górze planszy.
2. Możliwość wczytywania wartości, której chcemy użyć w obliczeniach. daje klocek **Wprowadzenia danej**. Wstawmy go pod klokiem startowym. Teraz mając wskaźnik na kloku naciskamy *prawy przycisk myszy* i otwieramy w ten sposób *okno dialogowe*, gdzie w górnej części można wpisać komunikat np.: Podaj liczbę całkowitą, której potęgę będą obliczane. W dolnej części wpisujemy *nazwę zmiennej*, której wartość będzie wczytywana np. *liczba*. Po wpisaniu wybieramy przycisk OK i okno się zamyka.
3. Teraz weźmy klocek **Wykonania obliczeń** i dołączmy do poprzedniego. Co chcemy obliczać?
Otwieramy prawym przyciskiem myszy *okno dialogowe* i piszemy **p2 := liczba*liczba**, żeby przejść do następnej linii naciskamy Ctrl + Enter i piszemy dalej **p3 := p2*liczba**. Użyliśmy tu **symbolu przypisania** (:=), który oznacza, że należy zmiennej po jego lewej stronie nadać wartość określoną po prawej stronie. Jak widać, jest to coś innego niż znak równości (=), nie można przenosić wyrażeń z jednej jego strony na drugą. Znaki działań musimy dokładnie wypisywać. Ich operatory to: + dodawanie, - odejmowanie, * mnożenie i / dzielenie.
4. Chcemy jeszcze wypisać wynik obliczeń. W tym celu wstawiamy klocek **Wyprowadzenia wyniku**. Tak jak poprzednio prawym przyciskiem myszy otwieramy jego *okno dialogowe* i w górnej części wpisujemy komunikat np.: *Liczba podniesiona do 3 potęgi to:* a w dolnej - podajemy nazwę zmiennej, której wartość zostanie wypisana - p3. Potwierdzamy wypełnienie okna wybierając przycisk OK.
5. Musimy jeszcze wskazać, że to koniec algorytmu - wstawiamy klocek **Koniec algorytmu**.

Mamy gotowy pierwszy algorytm, zapisujemy go w projekcie **ALG1.PRJ** na planszy **ALG1.SCH**.

Pora na **wykonanie** algorytmu. Naciskamy **przycisk z błyskawicą** lub wybieramy z menu *Wykonanie/Uruchom*. Powoduje to oznaczenie *klocka startowego czerwona obwódka*. Wybieramy **przycisk z jedną stopą** (*Wykonanie/Krok*) powodujący wykonanie jednego kroku lub sąsiedni z **dwoma stopami** (*Wykonanie/Wykonuj*) powodujący wykonanie całego algorytmu.

Przegląd zmiennych możemy uzyskać otwierając okno **Zmienne** (*Widok/Zmienne* lub przycisk z okularami po prawej stronie).

W trakcie symulacji działania algorytmu, poza zmianami wartości zmiennych, można jeszcze dokładniej obserwować przebieg wykonywania algorytmu.

W tym celu należy otworzyć okno **Ślad** (*Widok/Ślad* lub przycisk z kartką po prawej stronie)

□ Rozgałęzienie algorytmu - sprawdzanie warunku

Pierwszy algorytm miał budowę liniową - realizował jeden ciąg obliczeń. Jest jednak wiele problemów, w których obliczenia mogą przebiegać w różnych wariantach w **zależności od spełnionego warunku**.

Zbudujmy algorytm, który będzie **wybierał większą z dwóch podawanych liczb**.

1. Start
2. Wczytujemy pierwszą, a następnie drugą liczbę;
3. Sprawdzamy czy większa jest pierwsza liczba;
Jeśli TAK, to 4A. wypisujemy ją z objaśnieniem, że jest ona większa;
Jeśli NIE, to 4B. wypisujemy drugą liczbę i informujemy, że jest ona większa.
5. Zakończenie

Ćwiczenie 2.

Proszę narysować **schemat blokowy** algorytmu na kartce, a następnie **zbudować go w programie ELI**, korzystając z poniższej instrukcji.

Zaczynamy od **klocka startowego**, potem dostawiamy **dwa klocki wprowadzania danych**, by wczytać obie liczby. Nazwijmy je pierwsza i druga. Teraz trzeba sprawdzić, która z nich jest większa.

W tym celu wstawiamy **klocek sprawdzenia warunku**.

Prawym przyciskiem myszy otwieramy jego okno i wpisujemy warunek np. *pierwsza > druga*. Do dyspozycji mamy relacje: = *równe*, > *większe*, < *mniejsze*, <> *nierówne*, >= *większe lub równe*, <= *mniejsze lub równe*.

Z klocka wychodzą dwie linie: jedna w lewo - wybierana w przypadku gdy warunek jest spełniony, druga w prawo, oznaczona literą N - gdy nie jest on spełniony. Następnie używamy linii łamanych, by do obu odgałęzień wstawić klocki wyprowadzenia wyniku. Będą się one jednak różnić komunikatami i wyprowadzaną liczbą. Odgałęzienie lewe powinno wyprowadzać pierwszą liczbę i wypisywać komunikat np.: *Większa jest pierwsza liczba*. Klocek połączony z N natomiast ma wyprowadzać drugą liczbę i wypisywać stosowny komunikat. Teraz możemy połączyć obie linie i wstawić **klocek końcowy**. Zapisujemy planszę w pliku **ALG2.SCH** i przechodzimy do wykonania algorytmu jak w poprzednim zadaniu.

Ćwiczenie 3.

Przedstawiony algorytm „gubi” szczególnie przypadek gdy liczby są równe. Proszę poprawić algorytm tak, by rozpoznał ten przypadek.

□ Pierwiastki równania kwadratowego

Niektóre problemy z matematyki i fizyki wymagają **rozwiązania równania kwadratowego** o ogólnej postaci:

$$a \cdot x^2 + b \cdot x + c = 0$$

gdzie **a, b, c** są współczynnikami liczbowymi (przy czym $a \neq 0$).

Znana jest ogólna metoda rozwiązywania takich równań. Polega ona na obliczeniu wyróżnika:

$$\text{delta} := b \cdot b - 4 \cdot a \cdot c$$

Jeśli jest on ujemny, to równanie nie ma rozwiązań; jeśli nie to obliczamy pierwiastki ze wzorów:

$$x1 := (-b - \text{sqrt}(\text{delta})) / (2 \cdot a)$$

$$x2 := (-b + \text{sqrt}(\text{delta})) / (2 \cdot a)$$

Pozwala to na ułożenie schematu obliczeń.

Ćwiczenie 4.

Plan działań - czyli algorytm jest następujący:

1. Zaczynamy
2. Wczytujemy trzy współczynniki
3. Obliczamy wyróżnik (delte)
4. Sprawdzamy, czy delta jest nieujemna

Jeśli TAK, to

- 5a. Obliczamy pierwiastki
- 6a. Wypisujemy oba pierwiastki
- 7a. Kończymy pracę.

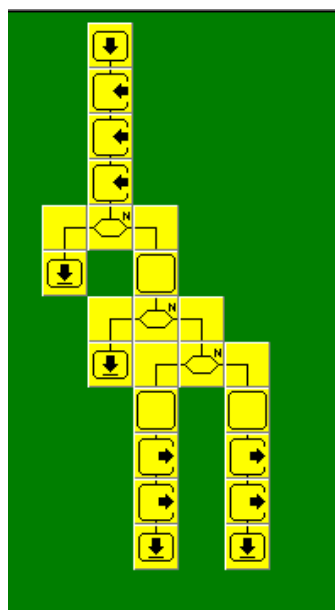
Jeśli NIE, to:

- 5b. Wypisujemy delte, informujemy o braku rozwiązań
- 6b. Kończymy pracę.

Znamy już wszystkie klocki potrzebne do ułożenia tego algorytmu.

Przy obliczaniu pierwiastków wykorzystujemy funkcję - **pierwiastek kwadratowy**. Jest ona zdefiniowana jako funkcja standardowa **sqrt()** podobnie jak inne podstawowe funkcje znane z lekcji matematyki.

W naszym algorytmie jest sporo obliczeń. Dobrze jest je opisać - skomentować tak, aby gdy wrócimy do zagadnienia po pewnym czasie, było ono dla nas (i dla innych) czytelne. Każdy klocek można wyposażyć w **komentarz** - by to zrobić należy **wskazać klocek i nacisnąć klawisz Shift oraz prawy przycisk myszy**. Pojawi się okienko, w którym można wpisać krótki tekst. Jeśli chcemy żeby pojawił się on w trakcie wykonania algorytmu, trzeba otworzyć okno komentarza (menu *Widok/Komentarz*). Zapisujemy w pliku **ALG4.SCH**



Ćwiczenie 5.

1. Ten algorytm można jeszcze usprawnić. Zauważmy, że gdy $\text{delta} = 0$, to oba pierwiastki są identyczne. Warto ten przypadek uwzględnić w algorytmie. Trzeba sprawdzić czy $\text{delta} = 0$. Napisanie szczegółowego rozwiązania zostawiamy Państwu.
2. Jeszcze jedno zadanie to zabezpieczenie algorytmu przed wprowadzeniem $a = 0$. Wtedy równanie przestaje być kwadratowe, wzory tracą sens. Można skonstruować algorytm uwzględniający ten przypadek i rozwiązujący wtedy równanie liniowe $b \cdot x + c = 0$.

Przykładowy algorytm może wyglądać jak na rysunku obok.