INFORMATYKA

OPROGRAMOWANIE KOMPUTEROWE

http://www.infoceram.agh.edu.pl

OPROGRAMOWANIE

Zbiór informacji w postaci zestawu instrukcji wraz z danymi przeznaczonymi dla komputera w celu rozwiązania wyznaczonych problemów (ang. software, w odróżnieniu od hardware – czyli sprzętu komputerowego, tj. materialnych części komputera. Firmware – oprogramowanie wbudowane w urządzenie i stanowiące jego integralną część).
Zadaniem oprogramowania jest przetwarzanie danych w określony sposób.
Oprogramowanie jest synonimem terminów program komputerowy oraz aplikacja, ale stosuje się go zwykle w odniesieniu do większych programów.

Rodzaje oprogramowania

- systemowe, realizujące funkcje konieczne dla działania systemu komputerowego,
- narzędziowe, wspomagające obsługę komputera, np. rozszerzenia systemu operacyjnego, programy diagnostyczne, narzędziowe, itp.
- oprogramowanie do tworzenia oprogramowania,
- biblioteki programistyczne, oprogramowanie używane przez inne programy,
- oprogramowanie użytkowe, realizujące cele danego użytkownika poprzez odpowiednie aplikacje

OPROGRAMOWANIE

System komputerowy zawiera zasadniczo trzy rodzaje oprogramowania:

- Oprogramowanie systemowe , tj. podstawowe, bez którego komputer nie będzie wykonywał żadnych operacji na plikach ani działań matematycznych. Jest programem, który działa jako pośrednik między użytkownikiem komputera, a sprzętem komputerowym. Jego zadaniem jest tworzenie środowiska, sprzyjającego stosowanie innych programów w wygodny i wydajny sposób.
- Oprogramowanie narzędziowe, usprawniające konfigurację lub naprawę systemu, wspomaga zarządzaniem zasobami sprzętowymi poprzez dogodne interfejsy użytkowe oraz modyfikuje oprogramowanie systemowe w celu usprawnienia wykonywania programów.
- Oprogramowanie użytkowe, zwane też aplikacyjnym lub aplikacjami, określa sposób w jaki zostają użyte zasoby systemowe do rozwiązywania konkretnych problemów obliczeniowych postawionych przez użytkownika (kompiler, systemy baz danych, gry, oprogramowanie biurowe).

OPROGRAMOWANIE UŻYTKOWE

Zapewnia bezpośredni kontakt komputera z użytkownikiem. Jest przeznaczone do realizacji czynności poleconych przez użytkownika oraz rozwiązywania zadanych przez niego problemów. Określenia: program użytkowy, aplikacja, czy aplikacja użytkowa są bliskoznacznymi dla terminu oprogramowanie użytkowe. Oprogramowanie użytkowe wymaga uprzedniego uruchomienia systemu operacyjnego.

Przykłady oprogramowania użytkowego:

- programy biurowe, w tym arkusz kalkulacyjny i edytory tekstu
- programy do zarządzania firmą: finansowo-księgowe, magazynowe, kadrowo-płacowe itp.
- programy do obsługi multimediów
- gry komputerowe
- inne rodzaje oprogramowania przeznaczone do realizacji różnych potrzeb użytkowników.

PROGRAMOWANIE KOMPUTERÓW

Jest to proces projektowania, tworzenia, testowania i utrzymywania kodu źródłowego programów komputerowych lub urządzeń mikroprocesorowych. Kod źródłowy jest napisany w języku programowania, z użyciem określonych reguł, może on być modyfikacją istniejącego programu lub czymś zupełnie nowym. Programowanie wymaga wiedzy m.in. z zakresu struktury danych i znajomości języków programowania oraz narzędzi programistycznych, wiedza nt. kompilatorów, czy sposób działania podzespołów komputera.

Czynniki wpływające na jakość programów

Tekst programu powinien być zapisany

- zrozumiale i czytelnie: nazwy zmiennych powinny odzwierciedlać ich funkcje, nazwy różnych zmiennych nie powinny być podobne
- w miejscach wprowadzania danych powinno zamieszczać się kontrolę ich poprawności (np. niedozwolone dzielenie przez 0)
- każda instrukcja powinna być w osobnym wierszu
- program powinien zawierać wcięcia
- należy używać komentarzy tam gdzie można mieć wątpliwości co do działania programu

JĘZYK PROGRAMOWANIA

Jest to zbiór zasad określających, kiedy ciąg symboli tworzy program komputerowy oraz jakie obliczenia opisuje.

PODZIAŁ JĘZYKÓW PROGRAMOWANIA

- **Zorientowane maszynowo** (asemblery): posługują się pojęciami na poziomie przesyłania informacji pomiędzy poszczególnymi komórkami pamięci

- **Zorientowane problemowo**: rozwiązują zagadnienia określonej klasy, np.: konstruowanie mostów, dyfuzja, termodynamika

- **Języki wysokiego poziomu**: są uniwersalnymi językami, tzn. za ich pomocą można rozwiązać problemy różnych dziedzin

JĘZYKI PROGRAMOWANIA

Kompilacja – proces, w którym program w ***języku wysokiego poziomu*** jest tłumaczony na ***język adresów symbolicznych*** (***asembler***). Program realizujący ten proces nazywany jest **kompilatorem**.

Interpreter – program tłumaczący każdą instrukcję na instrukcje poziomu maszyny i natychmiast ją wykonujący.

Fortran	do obliczeń numerycznych.
Cobol	język dla przedsiębiorstw i handlu
Pascal, C++	języki uniwersalne
Snobol	manipulowanie tekstami i napisami
Prolog	oparty na logice faktów
Lisp	przetwarzanie list, obliczenia symboliczne

NAJPOPULARNIEJSZE JĘZYKI PROGRAMOWANIA

C
Java
Objective-C
C++
PHP
C#
(Visual) Basic
Python
Transact-SQL
MATLAB

JavaScript
Visual Basic .NET
Perl
Ruby
Pascal
PL/SQL
Lisp
Delphi/Object Pascal
Groovy
COBOL
Lua

BŁĘDY WYSTĘPUJĄCE W PROGRAMACH

1) Błędy kompilacji:
   a) Polegają na *niezgodności programu z regułami budowy instrukcji*, np. przecinek zamiast kropki, zła liczba nawiasów itp.,
   b) Kompilator *sygnalizuje obecność* takiego błędu odpowiednim komunikatem i podpowiada miejsce jego wystąpienia.
   c) są łatwe do znalezienie i poprawienia

2) Błędy wykonania:
   a) Polegają na *żądaniu wykonania niedopuszczalnej operacji*, np. brak definicji funkcji, przekroczenie zakresu tablicy, wyczerpaniu obszaru pamięci itp,
   b) Są *wykrywane podczas kompilacji*
   c) Błędy takie mogą świadczyć o ***błędnej logice programu*** oraz o ***wadach*** zaprojektowanego algorytmu

BŁĘDY WYSTĘPUJĄCE W PROGRAMACH

3) Błędy logiki:
   a) **Zapętlenie się** programu
   b) Złe zaprojektowanie algorytmu, powodują, że poprawnie skompilowany program **wykonuje się nie poprawnie** lub drukuje błędne wyniki.
   c) Błędnie zapisany algorytm może **pomijać** pewne jego fragmenty lub **ograniczać** jego możliwości

4) Błędy danych:
   a) Wprowadzenie błędnych danych co powoduje **niepoprawne działanie** algorytmu

TESTOWANIE PROGRAMÓW

- testowanie programu dla danych testowych – testowanie odbywa się poprzez uruchamianie programu dla danych testowych
- debugging – proces wielokrotnego uruchamiania programu dla wykrycia i usunięcia błędów. Większość środowisk programistycznych jest wyposażona w narzędzia wspomagające proces debuggowania

Obydwie metody nie dają gwarancji wykrycia i usunięcia wszystkich błędów.

VISUAL BASIC

Jest to język programowania wysokiego poziomu i narzędzie programowania firmy Microsoft. Składnia jest oparta na języku BASIC, ale unowocześniona. Zawiera kilkaset instrukcji, funkcji i słów kluczowych. Nie jest językiem w pełni obiektowym.

Jest dostępny w trzech wersjach płatnych:
Learning Edition
Professional Edition
Enterprise Edition
oraz darmowej Express.

Dostępne są także wersje demonstracyjne środowiska Visual Basic:
Working Model
Control Creation Edition

Wraz z pojawieniem się platformy .NET, ukazała się nowa wersja Visual Basica pod nazwą Visual Basic .NET. Środowisko programistyczne ma mechanizmy importu starszych wersji programów, jednak w pewnych sytuacjach mogą pojawiać się komplikacje.
Język **Visual Basic** zastosowano również w wielu rozbudowanych aplikacjach jako język skryptowy do tworzenia zarówno prostych makr, jak i rozbudowanych aplikacji. Najbardziej znanym przykładem jest **Visual Basic for Applications** firmy Microsoft zastosowany w pakietach MS Office.

VISUAL BASIC FOR APPLICATIONS

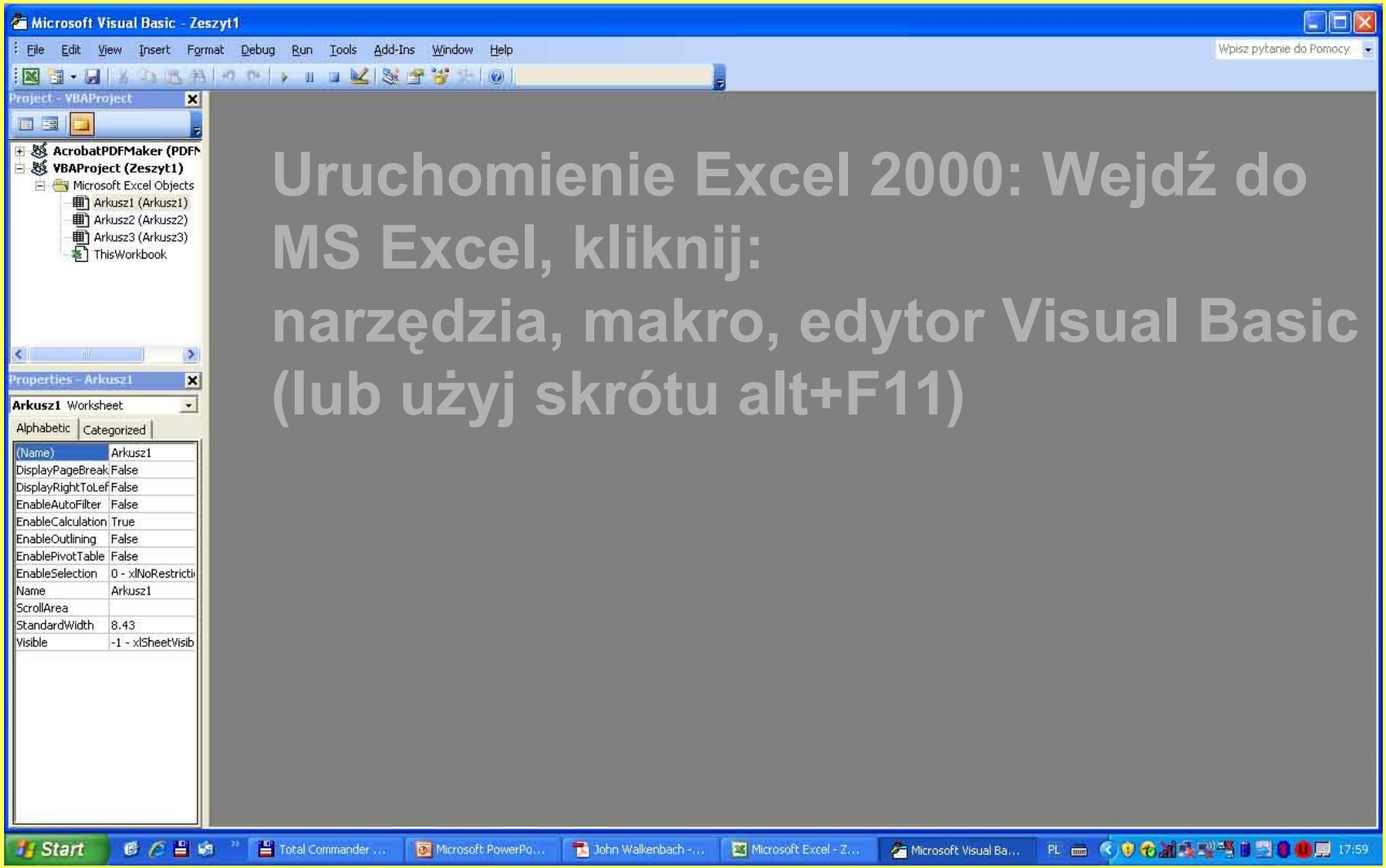
Visual Basic for Applications (VBA) – język programowania oparty na Visual Basicu (VB) zaimplementowany w aplikacjach pakietu Microsoft Office oraz kilku innych, jak na przykład AutoCAD i WordPerfect. Ta uproszczona wersja Visual Basica służy przede wszystkim do automatyzacji pracy z dokumentami, na przykład poprzez makropolecenia.

Podstawową różnicą między VBA a VB jest to, że VBA nie pozwala na tworzenie samodzielnych skompilowanych aplikacji typu EXE. Kod programu napisanego w VBA zawsze zawarty jest w dokumencie utworzonym przy pomocy programu obsługującego VBA - na przykład w pliku *.DOC edytora MS Word lub pliku *.XLS arkusza MS Excel. Program taki wymaga zatem środowiska uruchomieniowego, którym jest zainstalowana na komputerze aplikacja obsługująca dany dokument.

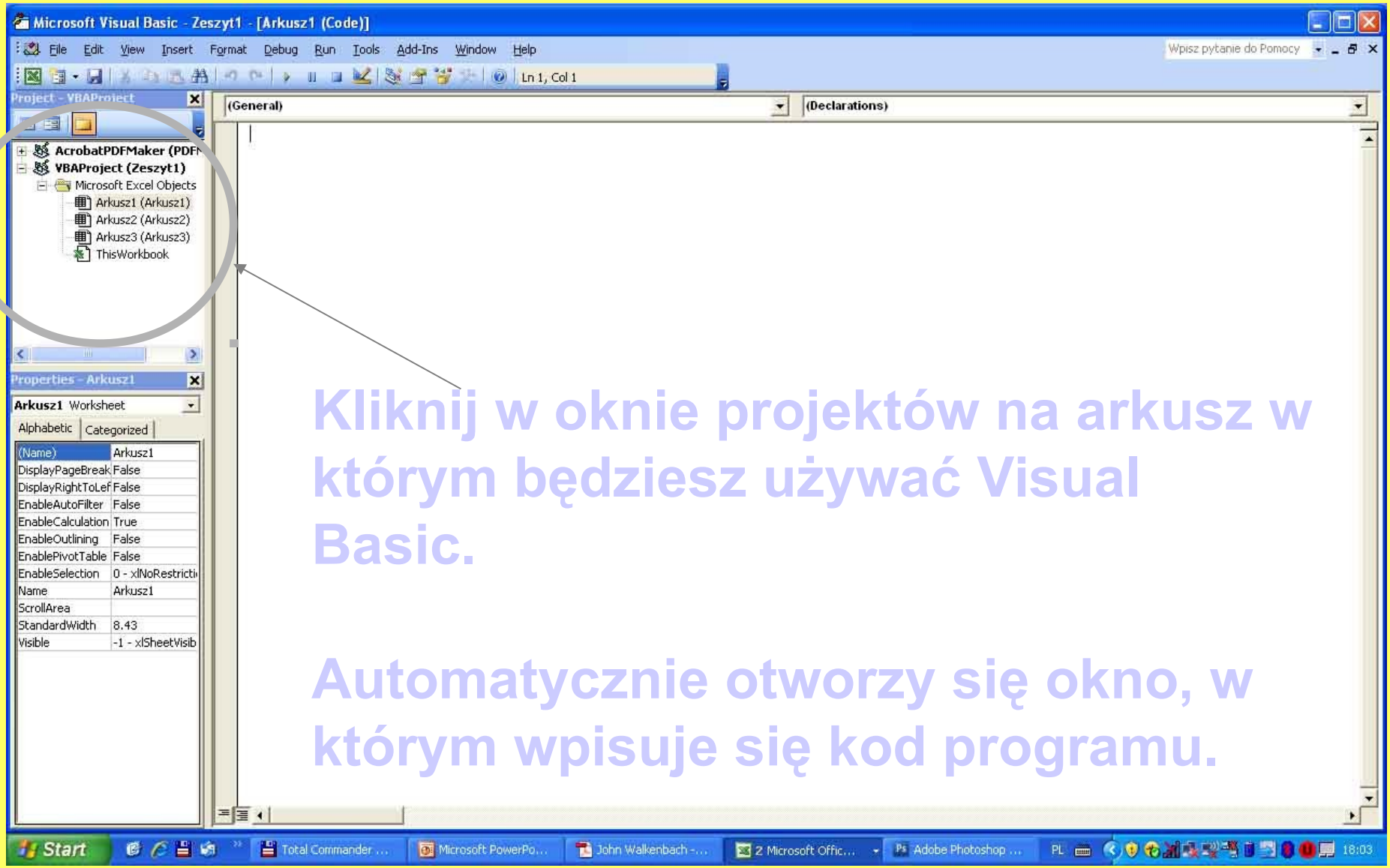
Wyjątkiem symulującym samodzielnie działające aplikacje są pliki utworzone w programie Microsoft Access, które - przy zakupie rozszerzenia Microsoft Office Developer lub innego, pozwalają na uruchamianie plików Accessa na dowolnej ilości komputerów w tzw. Microsoft Access Runtime, bez konieczności wyposażania każdego pojedynczego komputera w pełny pakiet Microsoft Office.

Z naszej strony internetowej:

Język programowania Visual Basic for Applications jest domyślnie zaimplementowany w pakiecie Microsoft Office. Uruchamianie edytora języka VBA jest uzależnione od posiadanej wersji pakietu Office'a. Aby uruchomić edytor języka Visual Basic for Applications należy najpierw otworzyć program Office Excel. Skrót klawiszowy do edytora jest na szczęście identyczny dla wszystkich wersji – **„Alt + F11"**.



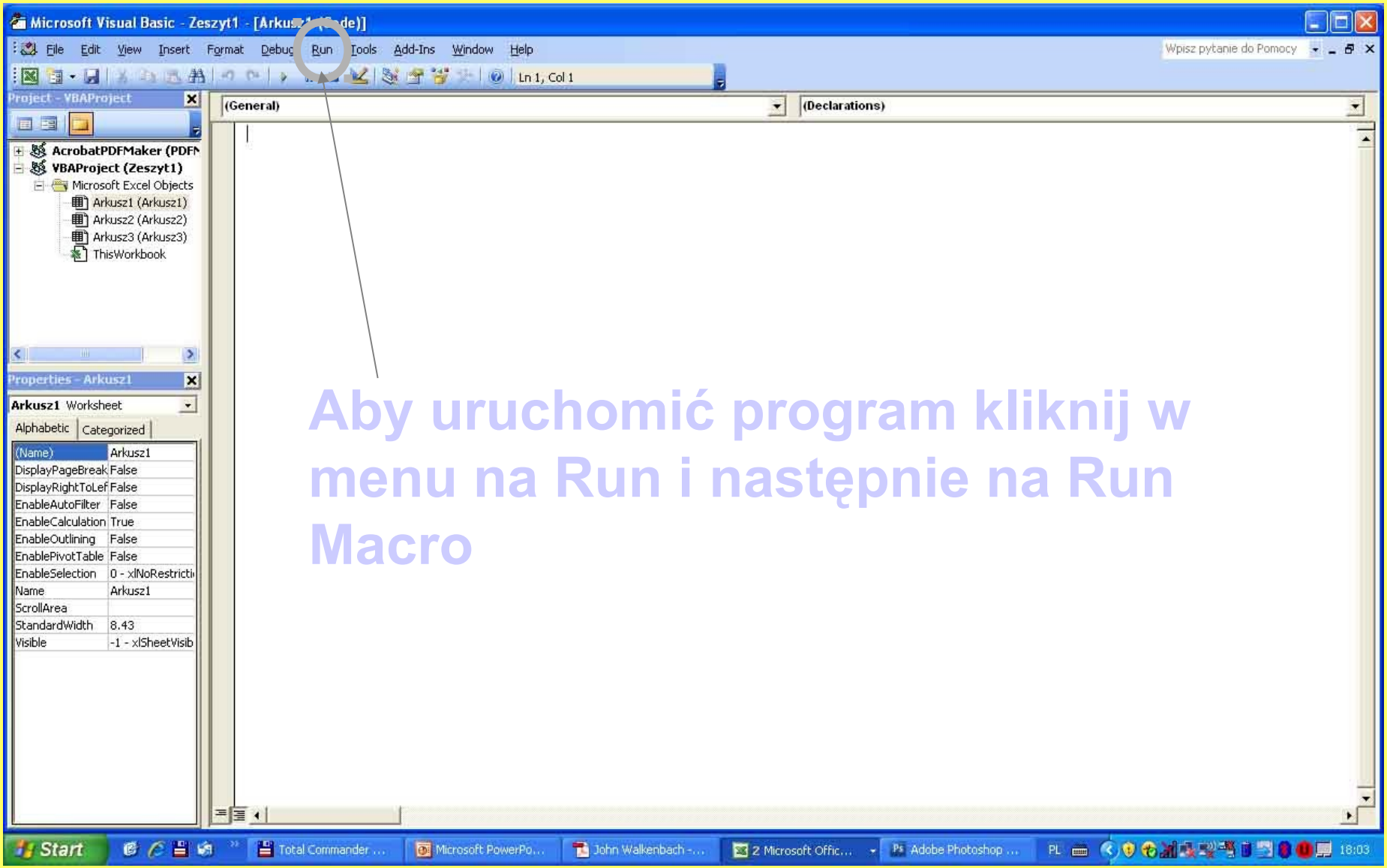
VISUAL BASIC FOR APPLICATIONS



Kliknij w oknie projektów na arkusz w którym będziesz używać Visual Basic.

Automatycznie otworzy się okno, w którym wpisuje się kod programu.

VISUAL BASIC FOR APPLICATIONS



VISUAL BASIC FOR APPLICATIONS

Każdy program napisany w VBA zaczyna się linijką „**Sub *nazwa_wlasna*()**” i kończy się „**End Sub**”:

```
Sub nazwa_wlasna()
‘cialo programu
End Sub
```

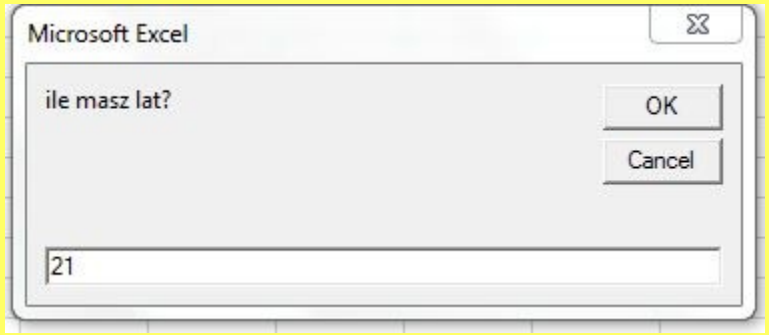
Wszystko co napiszemy po apostrofie ‘ będzie komentarzem (VBA nie „wykona” tej części).

Prawie każda zmienna, którą będziemy się posługiwali podczas pisania programu powinna być zadeklarowana, a więc powinna mieć zarezerwowane miejsce na dysku, powinna mieć przypisaną wartość, adres oraz typ. Podstawowe typy zestawione zostały poniżej oraz w suplemencie do notatki (w podpunkcie 3.3).

Zmienne deklarujemy w następujący sposób:

Dim nazwaWlasna As Typ

np.
Dim x As Double



Jeżeli chcemy wprowadzić dane do programu, musimy ją wprowadzić do jakiejś zmiennej istniejącej w programie tzn.:

Dim zmienna As Double
zmienna = InputBox(“ile masz lat?”)

Po uruchomieniu programu pojawi się okno dialogowe:

Typ danych	Rozmiar	Opis i zakres wartości
Byte	1 bajt	Liczby całkowite od 0 do 255
Boolean	2 bajty	Wartości logiczne: True (prawda) lub False (fałsz)
Integer	2 bajty	Liczby całkowite od -32 768 do 32 767
Long	4 bajty	Liczby całkowite od -2 147 483 648 do 2 147 483 647
Single	4 bajty	Liczby zmiennoprzecinkowe pojedynczej precyzji: od -3,402823E38 do -1,401298E-45 dla wartości ujemnych od 1,401298E-45 do 3,402823E38 dla wartości dodatnich
Double	8 bajtów	Liczby zmiennoprzecinkowe podwójnej precyzji: od -1,79769313486231E308 do -4,94065645841247E-324 dla wartości ujemnych od 4,94065645841247E-324 do 1,79769313486232E308 dla wartości dodatnich
Decimal	14 bajtów	Bardzo duża, bardzo precyzyjna liczba; może zawierać 29 cyfr oraz do 28 miejsc na prawo od przecinka. Wartości z przedziału: +/-79 228 162 514 264 337 593 543 950 335 bez przecinka dziesiętnego +/-7,9228162514264337593543950335 z 28 miejscami po przecinku Najmniejsza wartość niezerowa to: +/-0,0000000000000000000000000001
Date	8 bajtów	Daty i godziny: od 1 stycznia 100 do 31 grudnia 9999
String	liczba znaków	Tekst o stałej długości od 1 do 65 400 znaków
Variant	16 bajtów	Dowolna wartość liczbowa w zakresie określonym dla typu Double

VISUAL BASIC FOR APPLICATIONS

Pisząc kod programu rzadko posługujemy się konkretnymi wartościami liczbowymi lub tekstowymi. Częściej posługujemy się pewnymi symbolami (nazwami), którym podczas działania programu możemy przypisywać odpowiednie wartości. Symbole te nazywamy zmiennymi. Dzięki zmiennym możemy pisać programy, których sposób działania zależy od aktualnych informacji. Kiedy zmienia się wartość zmiennej, zmienia się sposób działania programu. A więc aby w pełni wykorzystać możliwości języka VBA należy stosować zmienne.

zmienna - opatrzone nazwą miejsce w pamięci do przechowywania danych, które mogą ulegać modyfikacjom w trakcie wykonywania programu. Każda zmienna zaopatrzona jest w unikatową nazwę, która identyfikuje ją w obrębie danego zakresu. Typ danych może być określony lub nie. Nazwy zmiennych muszą zaczynać się literą, muszą być unikatowe w obrębie swego zakresu, nie mogą być dłuższe niż 255 znaków i nie mogą zawierać kropki ani znaku deklarującego typ.

VISUAL BASIC FOR APPLICATIONS

Niejednokrotnie w kodzie programu stosowane są wartości, które nie zmieniają się podczas jego wykonywania lub też stosujemy wartości trudne do zapamiętania i nie mające oczywistego znaczenia. Można jednak kod programu uczynić łatwiejszym do czytania i modyfikowania wykorzystując stałe. Stała jest nazwą o określonym znaczeniu, która zastępuje niezmienną w kodzie programu wartość liczbową lub ciąg znaków. Nie można zmodyfikować stałej lub przypisać do niej nowej wartości, tak jak jest to możliwe w przypadku zmiennej. Stałą możemy zastosować w kodzie programu celem na przykład zagwarantowania niezmienności pewnej wartości.

stała - element o nadanej nazwie, który zachowuje stałą wartość przez cały czas działania programu. Stała może być ciągiem znaków lub literałem numerycznym, inną stałą lub dowolną kombinacją zawierającą operatory arytmetyczne i logiczne, z wyjątkiem operatora Is oraz operatora potęgowania. Każda aplikacja główna może definiować własny zestaw stałych. Dodatkowe stałe mogą być definiowane przez użytkownika za pomocą instrukcji Const. Stałych można użyć w dowolnym miejscu kodu programu zamiast ich rzeczywistych wartości.

VISUAL BASIC FOR APPLICATIONS

Zmienne tablicowe można wykorzystać na przykład do pracy ze zbiorem powiązanych ze sobą informacji. Zmienna tablicowa inaczej tablica jest zmienną zawierającą wiele komórek przeznaczonych do przechowywania wartości, podczas gdy typowa zmienna ma jedynie jedną komórkę, w której można przechowywać tylko jedną wartość. Obrazowo można to przedstawić w następujący sposób: Zwykłą zmienną możemy porównać do kontenera, który zawiera jeden pojemnik do przechowywania zmieniających się zawartości, tablice zaś możemy porównać do zestawu pojemników umieszczonych w takim właśnie kontenerze z których każdy może przechowywać inną zawartość. Pojemniki te ułożone są w odpowiednim porządku inaczej strukturze. Każdy taki pojedynczy pojemnik (element) zmiennej tablicowej jest oznaczony indeksem liczbowym określającym jego miejsce w danej strukturze tablicy.

tablica - zbiór kolejno indeksowanych elementów mających ten sam wewnętrzny typ danych. Każdy element tablicy posiada unikatowy numer indeksu. Przeprowadzenie zmian dla jednego elementu tablicy nie wpływa na inne jej elementy.

Visual Basic for Applications

Zmienne powinno się deklarować na początku każdej procedury.

Uwaga: w języku VBA nie trzeba deklarować typów zmiennych. Zmienna będzie miała wtedy typ Variant. Wiąże się to jednak czasami z błędami podczas wykonywania programu.

Aby wymusić na programiście podawanie typów danych na początku programu należy wpisać:

Option Explicit

Visual Basic for Applications

Nazwy zmiennych:
– powinny mieć **zrozumiałe nazwy**, w nazwie można stosować znaki alfanumeryczne liczby, ale pierwszy znak musi być literą

- język VBA **nie rozróżnia** wielkości liter

- nie można stosować **spacji lub kropek**

- nie można stosować **znaków specjalnych** (#, $, &, !)

- mogą mieć maksymalnie **124 znaki**

Visual Basic for Applications

Nazwy zmiennych – Przykłady

Niepoprawne nazwy zmiennych:

Aaaaa

4aaaa

aa.aa

aa#aa

Poprawne nazwy zmiennych:

Silnia

Silnia4

Visual Basic for Applications

Stałe – nie zmieniają swojej wartości podczas działania programu. Np. stałe fizyczne (stała gazowa)

Deklarowanie stałych:

Const nazwa_stałej as Integer = 4
Const nazwisko as String = „Nowak”

Visual Basic for Applications

Tablice – grupa elementów tego samego typu posiadających wspólną nazwę.
Deklarowanie Tablic:

```
Dim Tablica(1 To 100) As Integer
```

Tablice wielowymiarowe:

```
Dim Tablica2d(1 To 100, 1 To 20) As Double
```

```
Tablica2d(43,12) = 100.1
```

Wczytywanie za pomocą Arkuszy

Wczytywanie oraz wypisywanie danych można również wykonywać za pomocą arkuszy excelowskich (bez użycia okna dialogowego):

- Wczytywanie:
zmienna = Worksheets(“Arkusz1”).Range (“A1”)

lub krócej:
zmienna = Range (“A1”)

wówczas jednak odwołujemy się do arkusza, w którym piszemy skrypt (program).

- Wypisywanie:
Worksheets(“Arkusz1”).Range (“A1”) = zmienna

lub krócej:
Range (“A1”) = zmienna

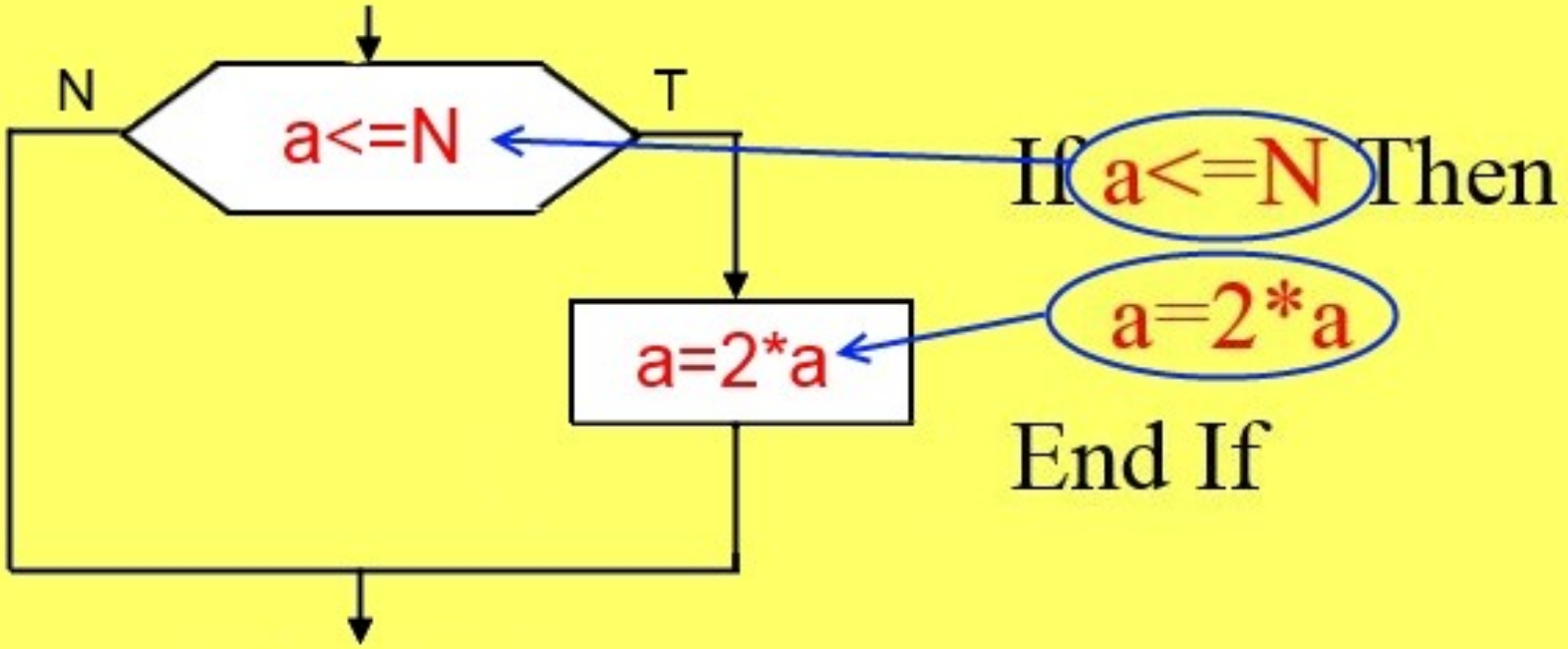
Pytanie warunkowe

If warunek Then

    instrukcja

End If

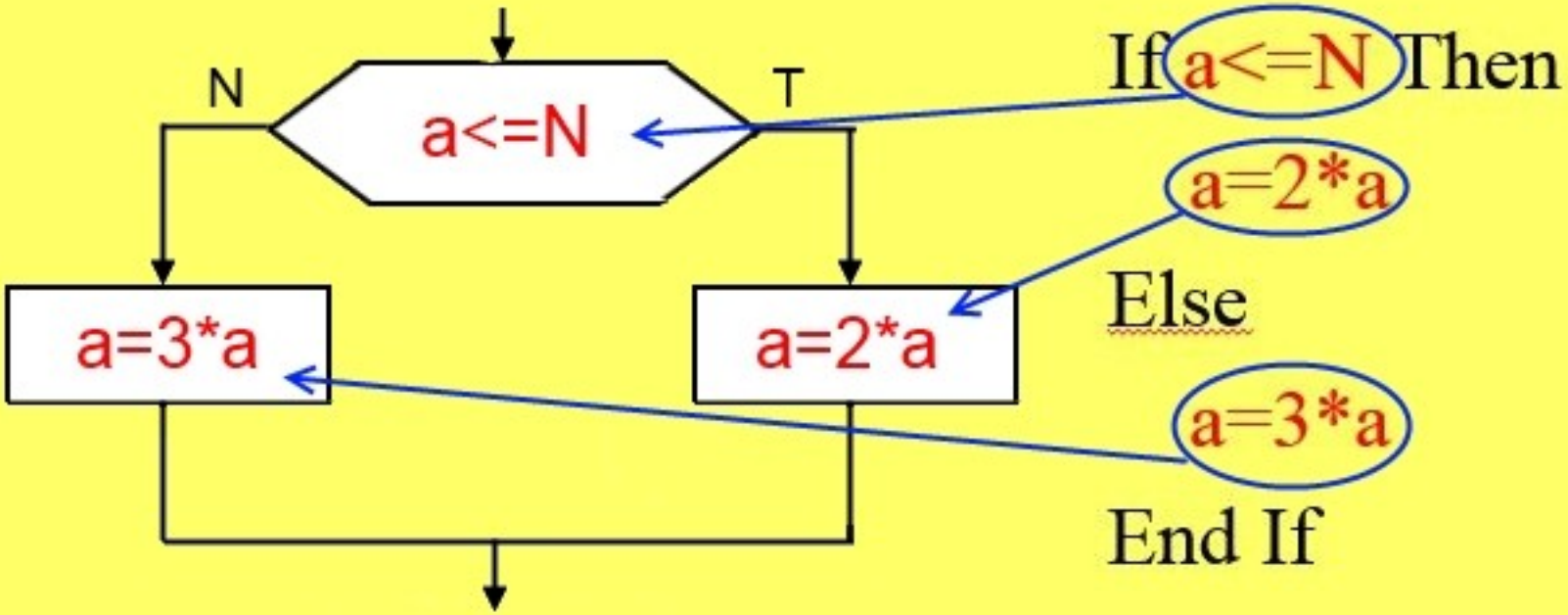
Pytanie warunkowe - przykład



Pytanie warunkowe c.d.

```
If warunek Then
    instrukcja
Else
    instrukcja2
End If
```

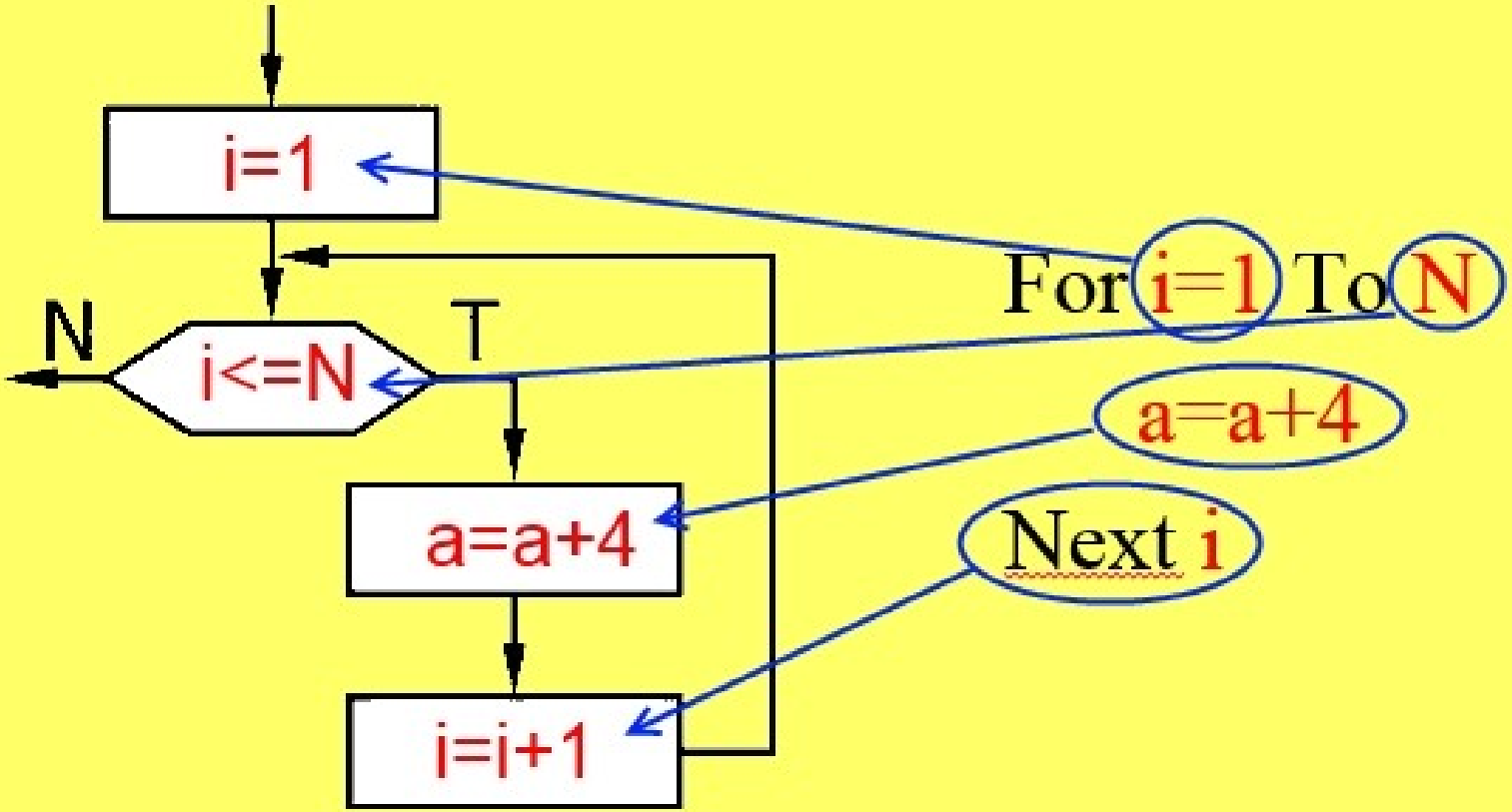
Pytanie warunkowe c.d. - przykład



Pętla for

```
For licznik=1 To N
    instrukcja
Next licznik
```

Pętla for - przykład



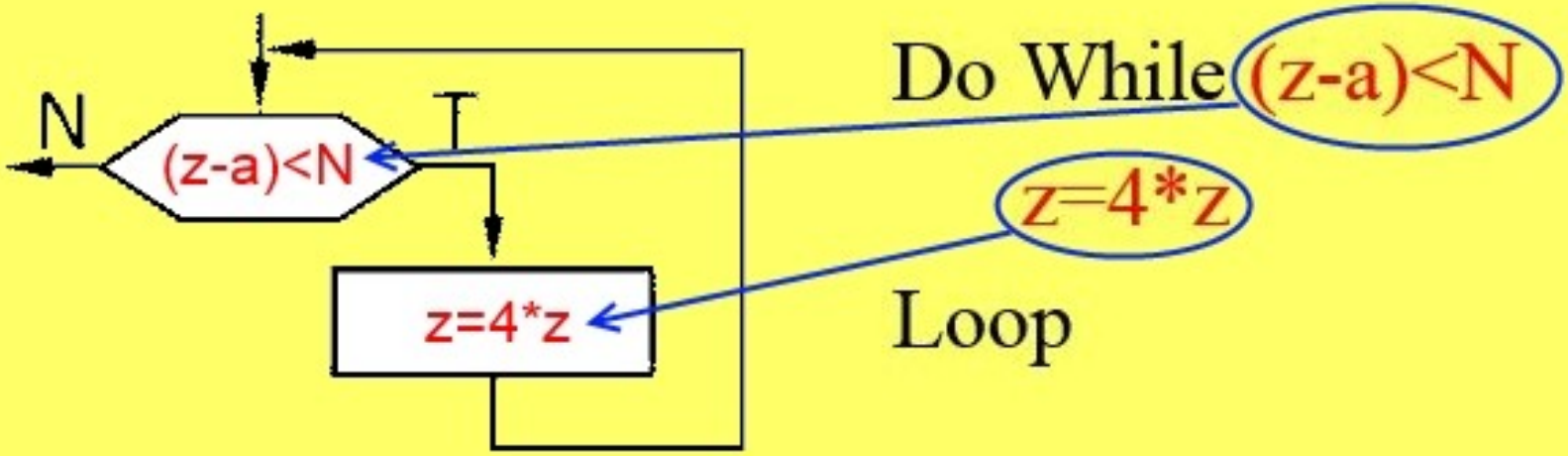
Pętla while

Do While warunek

    instrukcja

Loop

Pętla while - przykład



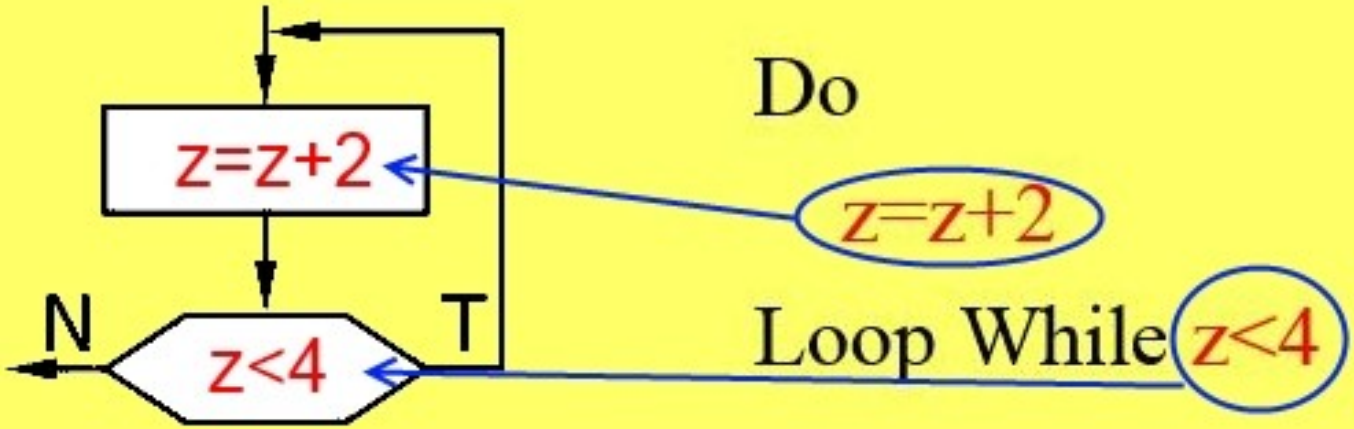
Pętla do while

Do

instrukcja

Loop While warunek

Pętla do while - przykład



VBA w Excelu - kurs dla początkujących http://dzono4.w.interia.pl/index.htm

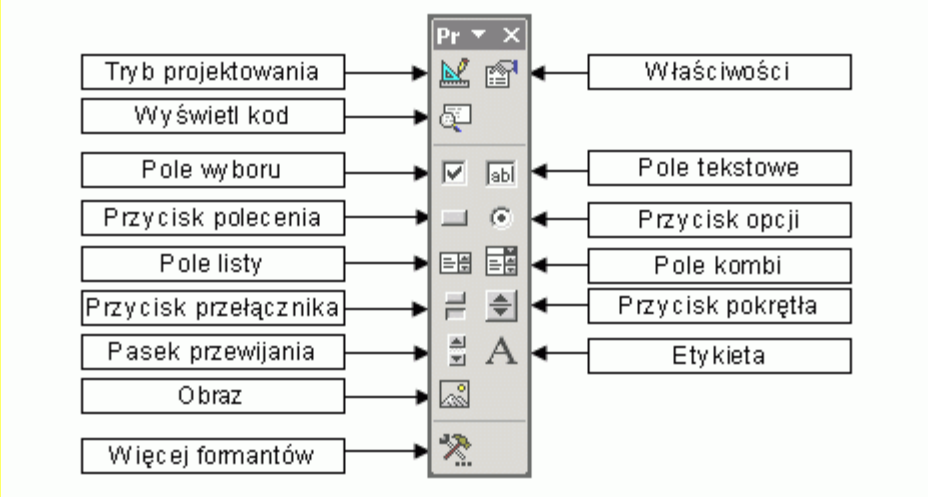
Visual Basic dla Aplikacji (VBA) to środowisko programowania dzięki któremu możemy wykorzystać w pełni (często ukryte) możliwości pakietu Microsoft Office lub innych aplikacji, w których jest wykorzystywane np. AutoCAD. Mimo że język VBA służy do doskonalenia i rozszerzenia możliwości macierzystej Aplikacji warto go poznać z kilku powodów:

- VBA jest językiem programowania praktycznie dostępnym wszędzie wystarczy mieć Worda, Accessa czy naszego Excela.
- Jest to język popularny i coraz więcej aplikacji go wykorzystuje.
- VBA jest językiem prostym doskonale nadającym się do nauki programowania, znając podstawy VBA o wiele łatwiej nauczyć się innych języków i tworzyć samodzielnie działające programy.
- Nie trzeba zbytnio zagłębiać się w środowisko VBA by przedstawiać gotowe efekty pracy.

Kurs ten jest opisem niektórych aspektów języka VBA w Excelu, przeznaczony jest dla osób początkujących. Powstał on z myślą o osobach chcących programować a które nie wiedzą jak w łatwy i w miarę przyjemny sposób zacząć.

VBA w Excelu - kurs dla początkujących
http://dzono4.w.interia.pl/index.htm

Pasek narzędzi **Przybornik formantów** dostarczający formanty ActiveX jest jednym z podstawowych narzędzi. W Przyborniku formantów zawarte są następujące elementy:

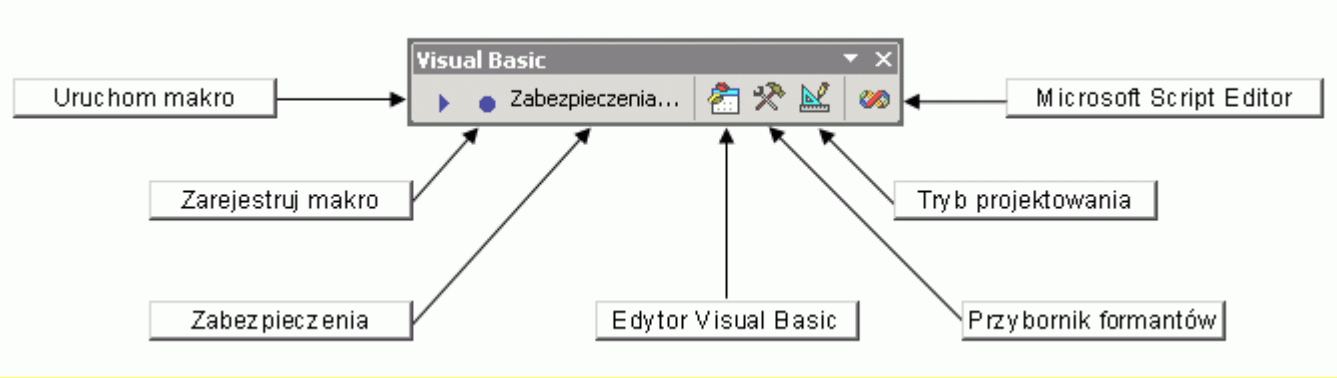


Aby Przybornik formantów był widoczny w arkuszu Excela: z menu **Widok** arkusza wybierz polecenie **Paski narzędzi** a następnie opcję **Przybornik formantów**. Innym sposobem jest kliknięcie na ikonę **Przybornik formantów** w pasku narzędzi **Visual Basic**. Aby uaktywnić pasek narzędzi **Visual Basic** z menu **Widok** wybieramy **Paski narzędzi** a następnie **Visual Basic**. Jeżeli chcemy dodać (wstawić) jakiś formant do arkusza Excela: w Przyborniku formantów klikamy na przycisk odpowiadający formantowi, który chcemy dodać a następnie miejsce w arkuszu gdzie ma się znajdować.

VBA w Excelu - kurs dla początkujących http://dzono4.w.interia.pl/index.htm

•**Tryb projektowania** - za pomocą tego przycisku włączamy lub wyłączamy tryb projektowania. Przycisk Tryb projektowania powinien być włączony jeżeli chcemy edytować, zmieniać właściwości lub przypisać kod do formantów.
•**Właściwości** - po kliknięciu na ten przycisk wyświetlane jest okno z wartościami właściwości zaznaczonego (aktywnego) w danym momencie formantu czy obiektu.
•**Wyświetl kod** - uruchamiany jest Edytor VisualBasic i wyświetlane jest okienko Kod programu aktywnego elementu.
•**Pole wyboru** - tworzy pole, poprzez które użytkownik może wskazać, czy jakieś stwierdzenie jest prawdziwe czy fałszywe. Jednocześnie w arkuszu można zaznaczyć więcej niż jedno pole wyboru.
•**Pole tekstowe** - przechowuje tekst, który użytkownik może wprowadzić lub zmienić.
•**Przycisk polecenia** - element z którego będziemy najczęściej korzystać, tworzy przycisk, który po kliknięciu inicjuje akcję.
•**Przycisk opcji** - przycisk używany do wybierania jednej opcji z grupy opcji.
•**Pole listy** - pole zawierające listę elementów.
•**Pole kombi** - pole tekstowe zawierające pole listy rozwijanej. Można wybrać element z listy albo wpisać własną pozycję.
•**Przycisk przełącznika** - tworzy przycisk, który można włączać i wyłączać.
•**Przycisk pokrętła** - przycisk, który może być połączony z komórką lub polem tekstowym. Aby zwiększyć wartość, należy kliknąć strzałkę w górę, aby zmniejszyć wartość klikamy strzałkę w dół.
•**Pasek przewijania** - formant służący do przewijania zakresu wartości.
•**Etykieta** - często stosowany formant, pozwala umieścić tekst, którego użytkownik nie będzie mógł zmienić, na przykład podpis pod ilustracją.
•**Obraz** - specjalny formant do wstawiania grafiki.
•**Więcej formantów** - jak sama nazwa wskazuje za pomocą tego przycisku uruchamiamy listę dodatkowych formantów ActiveX.

VBA w Excelu - kurs dla początkujących
http://dzono4.w.interia.pl/index.htm



Uruchom makro - za pomocą tego przycisku możemy uruchomić, edytować lub usunąć istniejące makro.
Zarejestruj makro - przycisk pozwala na zarejestrowanie (nagranie, utworzenie) nowego makra. Po zakończeniu rejestrowania makra klikamy na przycisk **Zatrzymaj rejestrowanie**.
Zabezpieczenia - możemy ustawić poziom zabezpieczeń przed wirusami makr.
Edytor Visual Basic - przycisk uruchamia Edytor Microsoft Visual Basic: Środowisko, w którym można edytować zarejestrowane makra oraz pisać nowe makra i programy w języku Visual Basic for Application. Jest to praktycznie właściwe środowisko naszej pracy w którym będziemy pisać kody naszych programów.
Przybornik formantów - pozwala wyświetlić i zamknąć pasek narzędzi Przybornik formantów dostarczający formanty ActiveX. Pasek ten opisałem na poprzedniej stronie kursu.
Tryb projektowania - za pomocą tego przycisku włączamy lub wyłączamy Tryb projektowania. Przycisk Tryb projektowania powinien być włączony jeżeli chcemy edytować lub zmieniać właściwości formantów.
Microsoft Script Editor - po kliknięciu na przycisk uruchamiany jest Microsoft Script Editor program używany do edytowania tekstu, tagów HTML i dowolnego kodu Microsoft Visual Basic Scripting Edition (VBScript) na stronie dostępu do danych. W programie Script Editor można również wyświetlić stronę w takiej postaci, w jakiej będzie się pojawiać w przeglądarce sieci Web. My na razie nie będziemy korzystać z tego przycisku.

OPERATORY STOSOWANE W VBA

- operatory arytmetyczne
- operatory porównania
- operatory logiczne

OPERATORY ARYTMETYCZNE

Operator	Operacja i opis	Przykład

^	**Potęgowanie** - podnosi wartość do potęgi określonej w wykładniku.	`Dim Wynik`
		`Wynik = 10 ^ 3 ' Wynikiem jest 1000`
		`MsgBox Wynik`

*	**Mnożenie** - wykonuje mnożenie .	`Dim Wynik`
		`Wynik = 10 * 3 ' Wynikiem jest 30`
		`MsgBox Wynik`

/	**Dzielenie** - wykonuje dzielenie i zwraca wynik w postaci zmiennoprzecinkowej.	`Dim Wynik`
		`Wynik = 10 / 3 ' Wynikiem jest 3.333333`
		`MsgBox Wynik`

\	**Dzielenie** - wykonuje dzielenie i zwraca wynik w postaci liczby całkowitej.	`Dim Wynik`
		`Wynik = 10 \ 3 ' Wynikiem jest 3`
		`MsgBox Wynik`

Mod	**Modulo** - wykonuje dzielenie i zwraca tylko resztę z przeprowadzonego dzielenia.	`Dim Wynik`
		`Wynik = 10 Mod 3 ' Wynikiem jest 1`
		`MsgBox Wynik`

+	**Dodawanie** - sumuje dwie wartości (operatora tego możemy też użyć do łączenia ciągów).	`Dim Wynik`
		`Wynik = 10 + 3 ' Wynikiem jest 13`
		`MsgBox Wynik`

-	Operator ten stosuje się do znajdowania różnicy - **Odejmowanie** lub do zaznaczania ujemnej wartości wyrażenia numerycznego - **Negacja**.	`Dim Wynik`
		`Wynik = 10 - 3 ' Wynikiem jest 7`
		`MsgBox Wynik`
		`Wynik = -Wynik ' Wynikiem jest -7`
		`MsgBox Wynik`

OPERATORY PORÓWNANIA

Wynikiem porównania dwu łańcuchów znaków lub wartości numeryczne jest wartość **True** (Prawda) lub **False** (Fałsz).

Operator	Znaczenie	Przykład wyniku porównania
<	Mniejsze niż	`10 < 5 ' Wynikiem jest False`
<=	Mniejsze lub równe	`10 <= 5 ' Wynikiem jest False`
>	Większe niż	`10 >= 5 ' Wynikiem jest True`
>=	Większe lub równe	`10 >= 5 ' Wynikiem jest True`
=	Równe	`10 = 5 ' Wynikiem jest False`
<>	Nierówne	`10 <> 5 ' Wynikiem jest True`

OPERATORY LOGICZNE

Operatory logiczne - operatory stosowane do wykonywania operacji logicznych.

Operator logiczny sprawdza wartość (*True* lub *False*) każdego z dwóch podwyrażeń wyrażenia warunkowego, a następnie określa (w zależności od operacji logicznej) końcowy wynik wyrażenia.

And - operator ten służy do wyznaczania iloczynu logicznego dwóch wyrażeń (Koniunkcja). Przy zastosowaniu tego operatora zwracana jest wartość *True* (Prawda) jeżeli oba podwyrażenia mają wartość *True*. W innym wypadku zwracana jest wartość *False* (Fałsz).

Or - operator ten służy do wyznaczania sumy logicznej dwóch wyrażeń (Alternatywa). Przy zastosowaniu tego operatora zwracana jest wartość *True* o ile przynajmniej jedno z podwyrażeń ma wartość *True*.

Koniunkcja (AND)		
p	q	p□q
0	0	0
0	1	0
1	0	0
1	1	1

Alternatywa (OR)		
p	q	p□q
0	0	0
0	1	1
1	0	1
1	1	1

gdzie:

1 – zdanie prawdziwe

0 – fałszywe

Visual Basic for Applications

Zasady konstruowania programów w VBA:

Funkcję rozpoczynamy słowem **Function**
nazwa(argumenty) As Typ_danych_wyjściowych
Kończymy słowem **End Function**

Przykład:

```
Function VBA_Demo(Tekst As String) As Integer
   MsgBox (Tekst)
   VBA_Demo = 1
End Function
```

Visual Basic for Applications

Funkcje wbudowane:

Sqr(value) – pierwiastek z liczby
MsgBox(komunikat) – wyświetla komunikat
Timer – zwraca czas procesora
A Mod b – a modulo b

KONIEC