

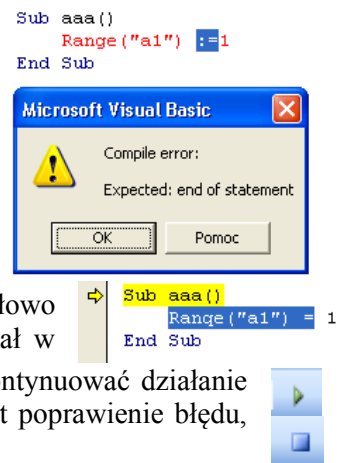
Rodzaje błędów

Podczas tworzenia makropolecień możemy spotkać się z trzema typami błędów.

Błędy składni, to błędy spowodowane niepoprawnie napisanym kodem programu. Sygnalizowane są okienkiem z napisem **Compile error**, oznaczeniem wiersza z błędem czerwonym kolorem i zaznaczeniem błędu. Takie błędy muszą być poprawione – makro się nie uruchomi. W pokazanym przykładzie powinien być znak „=” a nie „:=”.

Błędy podczas działania powodują nieoczekiwane zatrzymanie działania programu i wyświetlenie okienka błędu. Błąd taki powstanie, gdy niepoprawna składnia lub dane, na których działa program zmuszą program do jego zatrzymania. W przykładzie obok wpisano słowo „Range” zamiast poprawnie „Range”. Program już się uruchomił i nieoczekiwanie zatrzymał w wierszu oznaczonym żółtym podświetleniem. Wiersz z błędem jest podświetlony. Można kontynuować działanie programu po poprawieniu błędu, wciskając przycisk **RUN**. Najlepszym jednak wyjściem jest poprawienie błędu, zatrzymanie działania programu przyciskiem **RESET** i wykonanie makra jeszcze raz od nowa.

Błędy logiczne, to błędy spowodowane przez źle napisany program przez programistę – innymi słowy program generuje niepoprawne wyniki lub też zawiesza w trakcie działania. Składnia jest poprawna, ale źle dobrany jest algorytm. Jeśli programista użyje np. polecenia „suma= ”=SUMA(A1:A10)”, a miał na myśli podsumowanie wszystkich komórek z zakresu A1:A10, to program będzie cały czas generował złe wyniki sumowania – zamiast dwukropka użyto średnika i sumowane będą tylko komórki A1 i A10.



Obsługa błędów w programie

Obowiązkiem programisty jest zapewnienie takiego działania programu, aby program był odporny na różnego rodzaju nieprzewidziane okoliczności. Program powinien informować o błędzie i podjąć takie działanie, aby niedopuszczyć do jego przerwania. W normalnej sytuacji, gdy pojawia się błąd – program zostanie przerwany i przechodzimy do edytora. Dla użytkownika programu jest to sytuacja niedozaakceptowania. Jeżeli jednak dołączymy do kodu programu instrukcję **On Error** i pojawi się błąd obsługa błędu zostanie skierowana w miejsce wskazane przez programistę.

W opisanym obok przykładzie do komórki A1 wpisujemy to, co użytkownik wpisze do okienka InputBox podniesione do kwadratu. Jeżeli jednak użytkownik wpisze tekst, program zostanie przerwany (nie można wykonywać działań arytmetycznych na tekstach). Wpisanie liczby spowoduje przeliczenie i wpisanie wyniku do komórki, a instrukcja **Exit Sub** kończy działania procedury. Gdy wystąpi błąd program przeskakuje automatycznie do instrukcji po słowie opisanym w poleceniu **On Error GoTo** (tutaj błąd). W naszym przykładzie pojawia się komunikat, a po nim zakończenie działania procedury. Dodatkowo zastosowana instrukcja **Resume** nie spowoduje zakończenia lecz przeskoczenie do instrukcji, która spowodowała błąd – w tym wypadku poproszeni zostaniemy ponownie o wczytanie liczby.

```
Sub program()  
On Error GoTo błąd  
Range("A1") = InputBox("podaj liczbę") ^ 2  
Exit Sub  
błąd:  
MsgBox ("wpisano tekst zamiast liczby")  
Resume  
End Sub
```

Funkcje VB

Funkcje matematyczne

Rnd – liczba losowa

$x = \text{Int}((6 * \text{Rnd}) + 1)$ – w zmiennej x liczby losowe z zakresu 1 do 6

Abs – wartość bezwzględna

$x = \text{Abs}(-1)$ – w zmiennej x liczba 1

Int – część całkowita

$x = \text{Int}(10/3)$ – w zmiennej x liczba 3 – zaokrąglenie zawsze w dół

Round – miejsc po przecinku

$x = \text{Round}(10/3; 3)$ – w zmiennej x liczba 3,333 – zaokrąglenie, jak w matematyce

Funkcje daty i czasu

Date – aktualna data z zegara komputera

Day – zwraca dzień z daty

$d = \text{Day}(\text{Date})$ – w zmiennej d dzień aktualnej daty

Month – zwraca miesiąc z daty

$m = \text{Month}(\text{Date})$ – w zmiennej m miesiąc aktualnej daty

Year – zwraca rok z daty

$r = \text{Year}(\text{Date})$ – w zmiennej r rok aktualnej daty

CDate – zamienia tekst na datę

$d = \text{CDate}("2011-04-01")$ – w zmiennej d data (w formacie daty)

Funkcje tekstowe

Val – zamienia tekst na liczbę

$l = \text{Val}("123")$ – w zmiennej l liczba 123

Str lub **CStr** – zamienia liczbę na tekst

$t = \text{Str}(123)$ – w zmiennej t tekst "123"

LCase – małe litery

$m = \text{LCase}("Wacek")$ – w zmiennej m tekst „wacek”

UCase – duże litery

$m = \text{UCase}("Wacek")$ – w zmiennej m tekst „WACEK”

Len – długość tekstu

$d = \text{Len}("Wacek")$ – w zmiennej d liczba 5

Trim – usuwa początkowe i końcowe spacje

$t = \text{Trim}(" Wacek ")$ – w zmiennej m tekst „Wacek”

Left – część tekstu z lewej

$m = \text{Left}("Wacek", 2)$ – w zmiennej m tekst „Wa”

Right – część tekstu z prawej

$m = \text{Right}("Wacek", 2)$ – w zmiennej m tekst „ek”

Mid – część tekstu ze środka

$m = \text{Mid}("Wacek", 2, 3)$ – w zmiennej m tekst „ace”

InStr – szuka jednego tekstu w drugim

$m = \text{InStr}("Wacek", "c")$ – w zmiennej m liczba 3 – „c” na 3 pozycji

Wykorzystanie komórek arkusza do obliczeń VB

Przedstawione funkcje, to tylko fragment tych podstawowych. Prócz nich korzystać można również z pełnego zestawu funkcji, które posiada arkusz kalkulacyjny. Problemem jest ich angielskie nazewnictwo i inny zapis. bardzo prostym sposobem na ominięcie tych niedogodności jest zastosowanie „sztuczki”, która polega na wykorzystaniu komórek arkusza. Zamiast w makropoleceniu używać np. `Range("G1").FormulaR1C1 = "=VLOOKUP(100,RC[-6]:R[31]C[-6],3)"` można wpisać do komórki np. M1 formułę `=WYSZUKAJ.PIONOWO(100;A1:A32;3)`, a w makropoleceniu po prostu odczytać zawartość tej komórki i wpisać je do G1 np. poleceniem `Range("G1") = Range("M1")`

Obsługa innych arkuszy

Zapis do komórki i odczyt z komórki realizujemy za pomocą funkcji `Range` i `Cells`. Bardziej ogólny zapis pozwala na odwołanie do komórki, która znajduje się w innym arkuszu niż bieżący. W pokazanej w ramce sytuacji, pierwsza instrukcja zapisze do komórki A1 liczbę 1 na bieżącym arkuszu (z którego wywołano makro). Druga instrukcja zapisze na pewno do komórki A1 na arkuszu „Arkusz2”.

<pre>Range("A1")=1 Sheets("Arkusz2").Range("A1")=1</pre>
--

Ćwiczenia

1. Wczytujemy za pomocą `InputBox` liczbę dziesiętną. Zapisujemy do komórki A1. Część całkowita i ułamkowa wyprowadzona w komórkach A2 i A3 oraz w okienku `MsgBox` w dwóch osobnych wierszach. Gdyby użytkownik wpisał tekst zamiast liczby, program może zostać przerwany. Dopisz obsługę tego błędu, która spowoduje wypisanie odpowiedniego komunikatu i zakończenie działania procedury.
2. Do komórki B1 wpisujemy za pomocą okienka `InputBox` aktualną datę. W komórkach B2, B3 i B3 wyliczamy rok, miesiąc i dzień tej daty. W komórce B5 wypisujemy datę w postaci DD.MM.RRRR. Gdyby użytkownik nie wpisał oczekiwanej daty, program mógłby zostać przerwany. Dopisz obsługę tego błędu, która spowoduje wypisanie odpowiedniego komunikatu i zakończenie działania procedury.
3. Za pomocą okienka `InputBox` wczytaj datę swoich urodzin. W okienku `MsgBox` napisz liczbę dni, które minęły od dnia twoich urodzin. Oblicz również lata oraz miesiące. Gdyby użytkownik wpisał dowolny tekst zamiast oczekiwanej daty, program mógłby zostać przerwany. Dopisz obsługę tego błędu, która spowoduje wypisanie odpowiedniego komunikatu i zakończenie działania procedury.
4. Za pomocą okienka `InputBox` wczytaj do komórki C1 swoje imię i nazwisko. W komórkach C2 i C3 wypisz osobno imię i nazwisko (wyszukaj spację, wytnij z prawej i lewej). W komórce C4 wypisz inicjały (zawsze) dużymi literami. W komórce C5 wypisz napis postaci „nazwisko imię” (odwrotnie).
5. Za pomocą okienka `InputBox` wczytaj do komórki C1 swoje imię i nazwisko. W okienku `MsgBox` wypisz skrót utworzony na potrzeby pewnego programu w postaci: pierwsza litera nazwiska - duża, kolejne dwie litery nazwiska – małe i pierwsza litera imienia – duża. Zamiana na duże i małe litery powinna odbywać się zawsze, niezależnie od wpisanego tekstu. Np. wpisano tekst „libront waclaw”, program wypisuje „LibW”.
6. Otwieramy plik „studenci.xls”. Tworzymy w nim nowy arkusz (domyślnie „Arkusz1”) i zmieniamy nazwę na „OBSŁUGA”. Do komórki A1 arkusza „OBSŁUGA” wpisujemy wyraz „Anna”, a do komórki A2 formułę: „=LICZ.JEŻELI(Studenci!C3:C375;A1)”, która liczy ilość osób o podanym w komórce A1 imieniu. Utwórz makropolecenie, które za pomocą okienka `InputBox` wczytujemy do komórki A1 arkusza „OBSŁUGA” imię, a następnie pobiera z komórki A2 wyliczoną wartość i wypisuje za pomocą okienka `MsgBox` komunikat: „Osób o imieniu ... jest ...” – w miejsca kropek oczywiście powinno makro wpisywać odpowiednie wartości.
7. Zarejestruj makropolecenie, które w całym arkuszu zmieni tło komórek na czarne, a teksty na żółte oraz drugie makro, które powróci do pierwotnych kolorów. Wykonaj odpowiednie przyciski.
8. Zarejestruj makropolecenie, które schowa nagłówki wierszy i kolumn i podobne makro, które pokaże nagłówki. Wykonaj odpowiednie przyciski. (Narzędzia / Opcje / Widok)