

**Przedmiot:** Informatyka w inżynierii produkcji

**Forma:** Laboratorium

**Temat:** Zadanie 5. MessageBox, InputBox, instrukcja Select Case i instrukcje pętli.

Celem ćwiczenia jest nabycie umiejętności wykorzystania w praktyce obsługi komunikatów użytkownika oraz instrukcji Select Case i pętli warunkowych w języku VBA. Wykonanie zadań umożliwi jednocześnie utworzenie uproszczonych narzędzi pomocnych w zagadnieniach inżyniersko-menadżerskich.

### Instrukcja MessageBox

Przykład składni:

```
MsgBox(tekst wyświetlany)
```

```
odp = MsgBox(tekst wyświetlany)
```

```
odp = MsgBox(tekst wyświetlany, styl okna+rodzaj przycisków, tytuł w nagłówku)
```

```
odp = MsgBox(tekst wyświetlany, , tytuł w nagłówku)
```

```
odp = MsgBox(tekst wyświetlany, styl okna+rodzaj przycisków, tytuł w nagłówku,  
pomoc,wartość numeryczna do pomocy)
```

```
MsgBox
```

**Odp:** zwracana wartość typu Integer o informacji, który przycisk został kliknięty.

| Stała (odp) | Wartość liczbowa | Opis                         |
|-------------|------------------|------------------------------|
| vbOK        | 1                | Naciśnięto przycisk OK       |
| vbCancel    | 2                | Naciśnięto przycisk Anuluj   |
| vbAbort     | 3                | Naciśnięto przycisk Przerwij |
| vbRetry     | 4                | Naciśnięto przycisk Ponów    |
| vbIgnore    | 5                | Naciśnięto przycisk Ignoruj  |
| vbYes       | 6                | Naciśnięto przycisk Tak      |
| vbNo        | 7                | Naciśnięto przycisk Nie      |

Opis wszystkich opcji dla argumentu **styl okna+rodzaj przycisków:**

| Stała              | Opis                                                                                                                                   |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| vbCritical         | Wyświetla komunikat w formie ikony błędu krytycznego i przycisku OK                                                                    |
| vbQuestion         | Wyświetla przycisk OK i ikonę znaku zapytania                                                                                          |
| vbExclamation      | Wyświetla komunikat w formie ikony ostrzeżenia i przycisku OK                                                                          |
| vbInformation      | Wyświetla komunikat w formie ikony informacji i przycisku OK                                                                           |
| vbOKCancel         | Wyświetla przyciski OK i Cancel                                                                                                        |
| vbOKOnly           | Wyświetla wyłącznie przycisk OK. Przycisk wyświetla się z defaultu. tzn. Jeśli go nie zdefiniujemy, wyświetli nam się właśnie ten styl |
| vbAbortRetryIgnore | Wyświetla przyciski Przerwij, Ponów próbę i Ignoruj                                                                                    |
| vbRetryCancel      | Wyświetla przyciski Ponów próbę i Anuluj                                                                                               |
| vbYesNo            | Wyświetla przyciski Tak i Nie                                                                                                          |
| vbYesNoCancel      | Wyświetla przyciski Tak, Nie i Anuluj                                                                                                  |

## ZADANIE 1.

Dokończ realizację narzędzia do kontroli realizacji zleceń. W arkuszu „Zadanie1” (jak poniżej) znajduje się przykład arkusza pozwalającego na rejestrację zleceń w przedsiębiorstwie. Dodaj podprogram o nazwie **Zad1** (uruchamiany w momencie aktywacji arkusza „Zadanie1”), który wyświetli powiadomienie o następującej treści:

(tytuł w nagłówku)

Firma ABC

(treść zasadnicza)

Realizacja zadań na dzień (data dzisiejsza):

Liczba zleceń niewykonanych: (dane z D12)

Liczba zleceń wykonanych: (dane z D13)

|    | A                 | B                | C                            | D           | E      | F           |
|----|-------------------|------------------|------------------------------|-------------|--------|-------------|
| 1  |                   |                  |                              |             |        |             |
| 2  | REALIZACJA ZLECEŃ |                  |                              |             |        |             |
| 3  | ZLECENIA          | DATA ROZPOCZĘCIA | DATA UKOŃCZENIA              | % WYKONANIA | GOTOWE | UWAGI       |
| 4  | Zlecenie nr 1     | 12.02.2018       | 18.02.2019                   | 0%          |        | brak części |
| 5  | Zlecenie nr 2     | 14.02.2018       | 23.02.2019                   | 50%         |        |             |
| 6  | Zlecenie nr 3     | 19.11.2018       | 28.11.2018                   | 100%        | Done   |             |
| 7  | Zlecenie nr 4     | 12.02.2018       | 04.03.2019                   | 25%         |        |             |
| 8  | Zlecenie nr 5     | 14.02.2018       | 03.03.2019                   | 25%         |        |             |
| 9  | Zlecenie nr 6     | 22.11.2018       | 30.11.2018                   | 100%        | Done   |             |
| 10 |                   |                  |                              |             |        |             |
| 11 |                   |                  |                              |             |        |             |
| 12 |                   |                  | Liczba zleceń niewykonanych: | 1           |        |             |
| 13 |                   |                  | Liczba zleceń wykonanych:    | 2           |        |             |
| 14 |                   |                  |                              |             |        |             |
| 15 |                   |                  |                              |             |        |             |

## Rozwiązanie.

Otwieramy edytor *VisualBasic* w karcie *Developer*. W lewej zakładce *Project* edytora z *MicrosoftExcelObject* wybieramy „Arkusz1”. W lewej górnej liście wybieramy obiekt „Worksheet”, a w prawej „Activate” jako wyzwalacz. Wpisujemy do naszego edytora następujący komunikat, który będzie wyświetlany przy każdorazowej aktywacji arkusza zawierającego plan realizacji zleceń:

```
Private Sub Worksheet_Activate()  
    Dim tresc As String  
    Dim naglowek As String  
    Dim odp As Integer  
  
    naglowek = "Firma ABC"  
    tresc = "Realizacja zadań na dzień " & Date & ":"  
    & Chr(13) & Chr(13) & "Liczba zleceń niewykonanych: " & Range("D12")  
    & Chr(13) & "Liczba zleceń wykonanych: " & Range("D13")  
  
    odp = MsgBox(tresc, , naglowek)  
  
End Sub
```

Można zmienić styl okna i rodzaj przycisku np.:

```
odp = MsgBox(tresc, vbInformation + vbOKOnly, naglowek)
```

## ZADANIE 2. (do samodzielnego rozwiązania)

Zmień podprogram z *zadania 1* tak, aby komunikat otwierał się przy uruchomieniu pliku (a nie aktywacji Arkusza1). Podpowiedź: w obiekcie *Ten\_skoroszyt* tworzymy podprogram o nazwie *Workbook\_Open* i kopiujemy treść procedury z *zadania 1*.

## ZADANIE 3. (do samodzielnego rozwiązania)

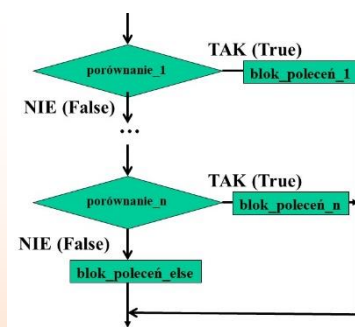
Zmień kod podprogramu z *zadania 1* tak, aby komunikat miał następujący styl okna i rodzaj przycisku:

- Komunikat pytający z przyciskiem Ok,
- Komunikat ostrzegawczy z przyciskami Ok, anuluj,
- Komunikat błędu krytycznego z przyciskami przerwij, ponów próbę, ignoruj.

### Instrukcja SELECT CASE

#### Postać blokowa

```
Select Case wyrażenieTekstowe
  Case warunek1
    [blok_poleczeń_1]
...
  Case warunekn
    [blok_poleczeń_n]
  Case Else
    [blok_poleczeń_else]
End Select
```



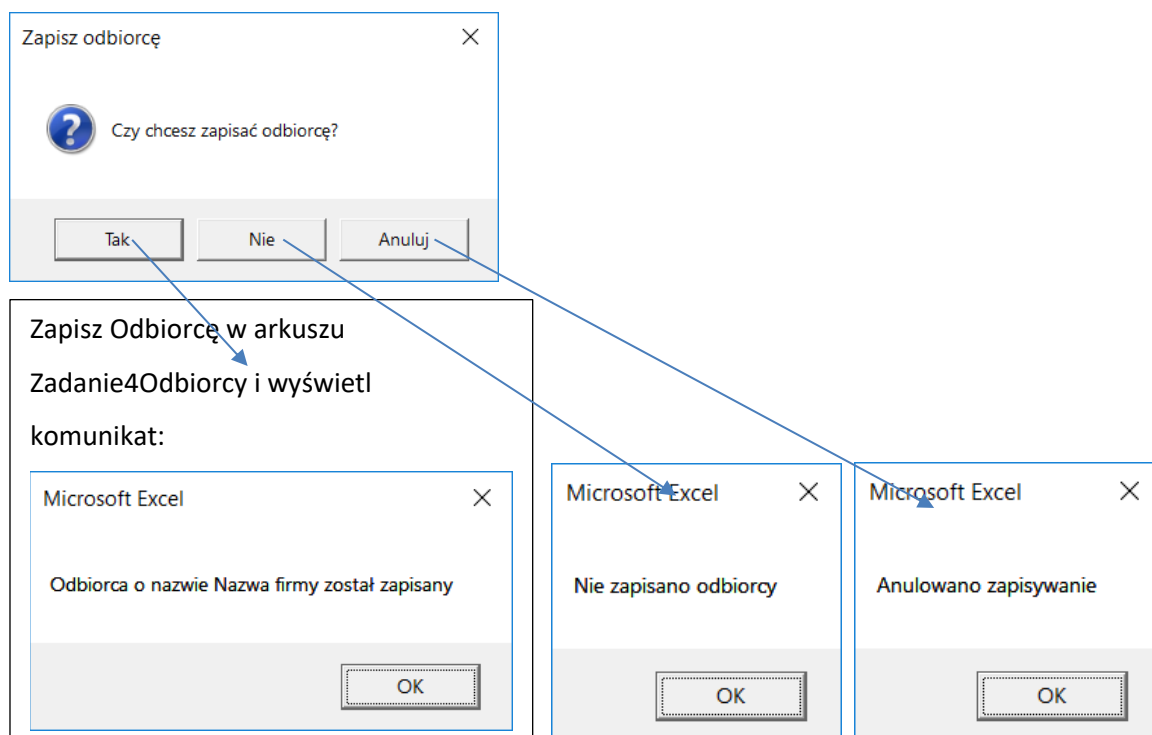
Wy Jeśli *warunek1* jest prawdziwy to następuje wykonanie instrukcji w tym warunku oraz wyjście z instrukcji *Select*. W przeciwnym przypadku następuje sprawdzenie *warunkun*. Kroki są powtarzane podobnie jak w przypadku *warunku1*. Jeśli żaden z warunków nie zostanie spełniony, to zostanie wykonana instrukcja *Case Else*.

## ZADANIE 4.

Dokończ realizację prostego narzędzia do tworzenia faktur. W arkuszu „*Zadanie4Faktura*” (jak poniżej) znajduje się wzór faktury, która oprócz prostych podsumowań, zawiera przycisk „*Dodaj odbiorcę*” pozwalający na dodanie do arkusza „*Zadanie4Odbiorcy*” danych odbiorcy, zapisanego na fakturze.

| A  | B                            | C          | D                        | E |
|----|------------------------------|------------|--------------------------|---|
| 1  | Nazwa Twojej firmy           |            | FAKTURA                  |   |
| 2  | Motto firmy                  |            |                          |   |
| 3  | Ulica                        | DATA:      | 11 styczeń 2019          |   |
| 4  | Kod pocztowy i miasto        | NR FAKTURY | 100                      |   |
| 5  | Telefon [numer] Faks [numer] | DOTYCZY:   | Opis projektu lub usługi |   |
| 6  | Odbiorca faktury:            |            | Dodaj odbiorcę           |   |
| 7  | Imię i nazwisko              |            |                          |   |
| 8  | Nazwa firmy                  |            |                          |   |
| 9  | Adres                        |            |                          |   |
| 10 |                              |            |                          |   |
| 11 |                              |            |                          |   |
| 12 | OPIS                         |            | KWOTA                    |   |
| 13 |                              |            |                          |   |
| 14 |                              |            |                          |   |
| 15 |                              |            |                          |   |

Oprogramuj przycisk „Dodaj odbiorcę” tak, aby realizował zadania według algorytmu (wykorzystaj instrukcję Select Case):



### Rozwiązanie.

Procedurę *Zad4()* należy uzupełnić o następujący kod program:

```
Sub Zad4()
    Dim odp As Integer
    Dim nazwa As String
    Dim i_n As String
    Dim adres As String
    Dim ostatni As Integer

    odp = MsgBox("Czy chcesz zapisać odbiorcę?", vbQuestion + vbYesNoCancel, "Zapisz odbiorcę")

    Select Case odp
        Case 6 'przycisk Tak
            i_n = Range("B7")
            nazwa = Range("B8")
            adres = Range("B9")

            With Worksheets("Zadanie4Odbiorcy")
                ostatni = .Cells(65536, 2).End(xlUp).Row + 1
                .Cells(ostatni, 2) = i_n
                .Cells(ostatni, 3) = nazwa
                .Cells(ostatni, 4) = adres
            End With
            MsgBox "Odbiorca o nazwie " & nazwa & " został zapisany"

        Case 7 'przycisk Nie
            MsgBox "Nie zapisano odbiorcy"

        Case 2 'przycisk Anuluj
            MsgBox "Anulowano zapisywanie"
    End Select
End Sub
```

### ZADANIE 5. (do samodzielnego rozwiązania)

Uzupełnij arkusz z fakturą (*Zadanie4Faktura*) o przycisk pozwalający na wydruk faktury. Podpowiedź: zarejestruj najpierw makro, które zapisze proces drukowania, a następnie skorzystaj z zarejestrowanego kodu.

## ZADANIE 6. (do samodzielnego rozwiązania)

Uzupełnij arkusz z fakturą (*Zadanie4Faktura*) o przycisk pozwalający na zapisanie historii wystawionych faktur (zapisanie wybranych danych faktury w osobnym arkuszy, analogicznie jak dla odbiorców).

### Pętla FOR

#### Postać blokowa

```
For Licznik=Początek To Koniec [Step Krok]
    [blok_poleceń]
[Exit For]
    [blok_poleceń]
Next [Licznik]
```

Pętla for pozwala nam wykonać kod określoną ilość razy. Wymaga ona dodatkowej zmiennej do przechowywania licznika pętli. Zmienna może mieć dowolną nazwę, musi być jednak typu liczbowego. Przed pierwszym wykonaniem przyjmuje on wartość *Początek*. Po każdym wykonaniu licznik zwiększany jest o jeden (lub o wartość *Krok* – jeżeli istnieje *Step Krok*). Pętla jest przerywana kiedy licznik jest większy od wartości *Koniec*.

## ZADANIE 7.

Dokończ narzędzie, pozwalające na tworzenie Specyfikacji Jakościowymi Produktu. Narzędzie składa się z dwóch arkuszy Excela: jeden arkusz (*Zadanie7Specyfikacja*) zawiera wzór specyfikacji, drugi arkusz (*Zadanie7Dane*) zawiera dane na temat jakości poszczególnych produktów. Dokument wzoru posiada odwołania do danych znajdujących się we wzorcowej kolumnie B (w arkuszu *Zadanie7Dane*). Utwórz makro pozwalające na wydruk do PDF dokumentu specyfikacji (arkusz *Zadanie7Specyfikacja*) dla wybranej ilości produktów (arkusz *Zadanie7Dane*).

### Rozwiązanie.

Zadanie można zrealizować w następujących krokach:

- Nagraj makro, które wydrukuje arkusz *Zadanie7Specyfikacja* do dokumentu PDF (zapisze jako pdf) o nazwie stanowiącej nazwę produktu, dla którego specyfikacja jest drukowana. Wygenerowany kod stanowi odpowiedź jak zapisać ten arkusz jako PDF pod określoną nazwą.
- Nagraj makro, które skopiuje w arkuszu *Zadanie7Dane* skopiuje kolumnę C a następnie wklei ją specjalnie jako wartości do kolumny B. Wygenerowany kod stanowi odpowiedź jak kopiować i wkleić specjalnie.
- Utwórz podprogram o nazwie *Zad7*, który:
  - wyświetli za pomocą INPUTBOX zapytanie o ilość produktów, dla których chcemy wydrukować dokumenty (zapisanie odpowiedzi do zmiennej typu Integer o nazwie *ilosc*),

- w pętli:
  - w arkuszu *Zadanie7Dane* skopiuje kolejno kolumny od C do C+ilość-1, następnie wklei je specjalnie jako wartości do kolumny B,
  - po każdym kopiowaniu wydrukuje arkusz *Zadanie7Specyfikacje* do PDF pod nazwą danego produktu.

```

Sub Zad7()
  Dim ilosc As Integer
  Dim nazwa_prod As String
  Dim i As Integer

  ilosc = InputBox("Podaj ile chcesz wydrukować dokumentów?")

  For i = 1 To ilosc

    With Worksheets("Zadanie7Dane")
      .Columns(i + 2).Copy
      'wklejanie specjalnie jako wartości do kolumny B
      .Columns("B:B").PasteSpecial Paste:=xlPasteValues, Operation:=xlNone
      'czyszczenie pamięci podręcznej (schowka)
      Application.CutCopyMode = False
    End With

    With Worksheets("Zadanie7Specyfikacja")
      nazwa_prod = .Cells(5, 3).Value
      'zapisanie arkusza jako pdf pod nazwą pliku nazwa_prod.pdf
      .ExportAsFixedFormat Type:=xlTypePDF, Filename:=nazwa_prod & ".pdf", _
        Quality:=xlQualityStandard, IncludeDocProperties:=True, IgnorePrintAreas _
        :=False, OpenAfterPublish:=False
    End With

  Next

End Sub

```

### ZADANIE 8. (do samodzielnego rozwiązania)

W zadaniu 7, zamiast funkcji INPUTBOX zastosuj funkcję APPLICATION.INPUTBOX tak, aby mieć pewność że użytkownik wpisze dane liczbowe.

### ZADANIE 9. (do samodzielnego rozwiązania)

W zadaniu 7, zamiast pętli FOR zastosuj pętlę WHILE.

Literatura

Wzory dokumentów po niewielkich zmianach pochodzą ze strony: <https://templates.office.com/>