

Bartoń Robert  
Podstawy  
Visual Studio 2017  
Visual Basic

Podstawy  
Visual Studio 2017  
Visual Basic  
Część II

# Wstęp

Książka w odcinkach na stronie Internetowej .Nie jest to może imponujące pisać książki w odcinkach ale biorę pod uwagę realia są osoby które korzystają ze Smartfonów ,Tabletów i innych Gadżetów .Które to obsługują pliki Przykład.pdf .Połączenia przez GPRS .Co za sobą ciągnie wolny dostęp przez GPRS ,do 56 Kb/s .A także chcą zapoznać się z nowinkami programowymi .Typu Visual Studio 2017 ,czy też inne informacje są bardziej interesujące , zapisane w Pdf- ach. Tak jak pod tytuł mówi są to Przykłady ,Instrukcje . „Z mojej szuflady” Dla początkujących mam nadzieję ciekawe źródło informacji . Dla Starych wyjadaczy może to być inspiracja do podzielenia się wiedzą z początkującymi Internautami , Programistami.

# Podziękowania

Dziękuję rodzinie za cierpliwość. Unii Europejskiej za program pomocy we wdrażaniu Cyfrowej Europy a raczej Świata . Firmie Microsoft za System operacyjny Windows na którym działa Visual Studio 2017

# Czym jest język programowania?

W pewnym sensie możesz postrzegać jako język programowania cokolwiek, co potrafi podejmować decyzje w środowisku komputera. Komputery potrafią doskonale podejmować decyzje, te jednak muszą być stosunkowo proste, na przykład „czy dana liczba jest większa od trzech?” lub „czy ten samochód jest niebieski?”. Jeśli chcesz podjąć skomplikowaną decyzję, proces jej podejmowania należy rozbić na mniejsze elementy zrozumiałe dla komputera. Do rozbijania złożonych decyzji na proste służą algorytmy. Dobrym przykładem problemu, z którym komputery radzą sobie niezbyt dobrze, jest rozpoznawanie ludzkich twarzy. Nie można po prostu zadać komputerowi pytania „czy to zdjęcie Kasi?”. W zamian trzeba rozbić pytanie na serię prostszych pytań zrozumiałych dla komputera. Decyzja podejmowana przez komputer to jedna z dwóch możliwych odpowiedzi: tak lub nie. Te możliwości określa się także jako prawdę i fałsz lub 1 i 0. Używając pojęć obecnych w programie, nie można kazać komputerowi podjąć decyzji na podstawie pytania „o ile większa jest liczba 10 w porównaniu z liczbą 4?”. Poprawnie zadane pytanie brzmi: „czy liczba 10 jest większa od 4?”. Różnica nie jest duża, ale istotna — pierwsze pytanie nie pozwala udzielić odpowiedzi tak lub nie, podczas gdy drugie to umożliwia. Komputer oczywiście potrafi udzielić odpowiedzi na pierwsze z tych pytań, ale wymaga to wykonania pewnych operacji. Mówiąc inaczej, aby odpowiedzieć na pierwsze pytanie, należy odjąć 4 od 10 i użyć wyniku tego działania w odpowiednim miejscu algorytmu. Wymóg stawiania pytań pozwalających na udzielenie odpowiedzi tak lub nie może Ci się wydawać pewnym ograniczeniem, jednak nie należy tak go traktować. Decyzje podejmowane w życiu codziennym wyglądają tak samo. Kiedy podejmujesz jakąś decyzję, możesz coś zaakceptować (tak, prawda, 1) lub odrzucić (nie, fałsz, 0). Ta książka opisuje język Visual Basic 2017, ale istotne aspekty programowania są w dużym stopniu niezależne od języka. Kluczowe jest zrozumienie, że każdy program, niezależnie od naszpikowania nowinkami i języka, w którym został napisany, składa się z metod (funkcji i procedur, które są wierszami kodu służącymi do implementacji algorytmu) oraz zmiennych (miejsc przechowywania danych, którymi manipulują metody)

# Algorytmy

Zmiany w przemyśle komputerowym zachodzą z niesamowitą szybkością. Większość profesjonalnych programistów musi cały czas uczyć się czegoś nowego, aby ich umiejętności były aktualne. Jednak niektóre aspekty tworzenia programów nie zmieniły się od czasu ich wymyślenia i prawdopodobnie nie zmienią się w najbliższym czasie. Dobrym przykładem aspektu technologii komputerowej, którego istota nie zmieniła się od początku, jest proces i dyscyplina tworzenia oprogramowania. Aby program działał, musi mieć dane, na których może pracować. Następnie program bierze takie dane i przekształca je na inną postać. Na przykład aplikacja może pobierać bazę z danymi klientów zapisaną w pamięci komputera jako zestaw jedynek i zer, a następnie przekształcać ją na postać możliwą do odczytania na ekranie monitora. Komputer pokładowy w samochodzie nieustannie analizuje informacje dotyczące środowiska oraz stanu pojazdu i przystosowuje mieszankę paliwa tak, aby zapewnić jak najwydajniejszą pracę silnika. Dostawca usług telekomunikacyjnych zapisuje wykonane telefony i na podstawie tych informacji generuje rachunki. Wspólną podstawą tych programów są algorytmy. Przed napisaniem programu rozwiązującego dany problem, trzeba rozbić go na pojedyncze kroki opisujące rozwiązanie problemu. Algorytm jest niezależny od języka programowania, dlatego możesz zapisać go zarówno za pomocą języka naturalnego, jak i diagramów lub w inny sposób ułatwiający wizualizację problemu. Wyobraź sobie, że pracujesz dla firmy telekomunikacyjnej i masz za zadanie wygenerować rachunki na podstawie telefonów wykonanych przez klientów. Poniższy algorytm opisuje jedno z możliwych rozwiązań:

1. Pierwszego dnia miesiąca musisz utworzyć rachunki dla wszystkich klientów.
2. Z każdym klientem powiązana jest lista połączeń wykonanych w ubiegłym miesiącu.
3. Znasz długość każdej rozmowy, a także porę jej przeprowadzenia. Na podstawie tych informacji możesz obliczyć koszt każdego połączenia.
4. Każdy rachunek to łączna suma wszystkich połączeń.
5. Jeśli klient rozmawiał dłużej, niż wynosi ustalony limit, musi zapłacić określoną kwotę za każdą dodatkową minutę.
6. Do każdego rachunku dodajesz podatek.
7. Po przygotowaniu rachunku trzeba go wydrukować i wysłać pocztą.

Tych siedem punktów opisuje — dość wyczerpująco — algorytm programu generującego rachunki za połączenia wychodzące w systemie operatora telefonii komórkowej. Nie jest istotne, czy gotowe rozwiązanie napisane będzie w języku C++, Visual Basic 2017, C#, Java, czy w innym — podstawowy algorytm programu nie zmieni się. Warto jednak pamiętać, że poszczególne punkty powyższego algorytmu można rozbić na mniejsze, bardziej szczegółowe algorytmy. Dobrą wiadomością dla osób uczących się programować jest to, że zwykle utworzenie takiego algorytmu jest dość proste. Bardzo łatwo zrozumieć działanie powyższego algorytmu.

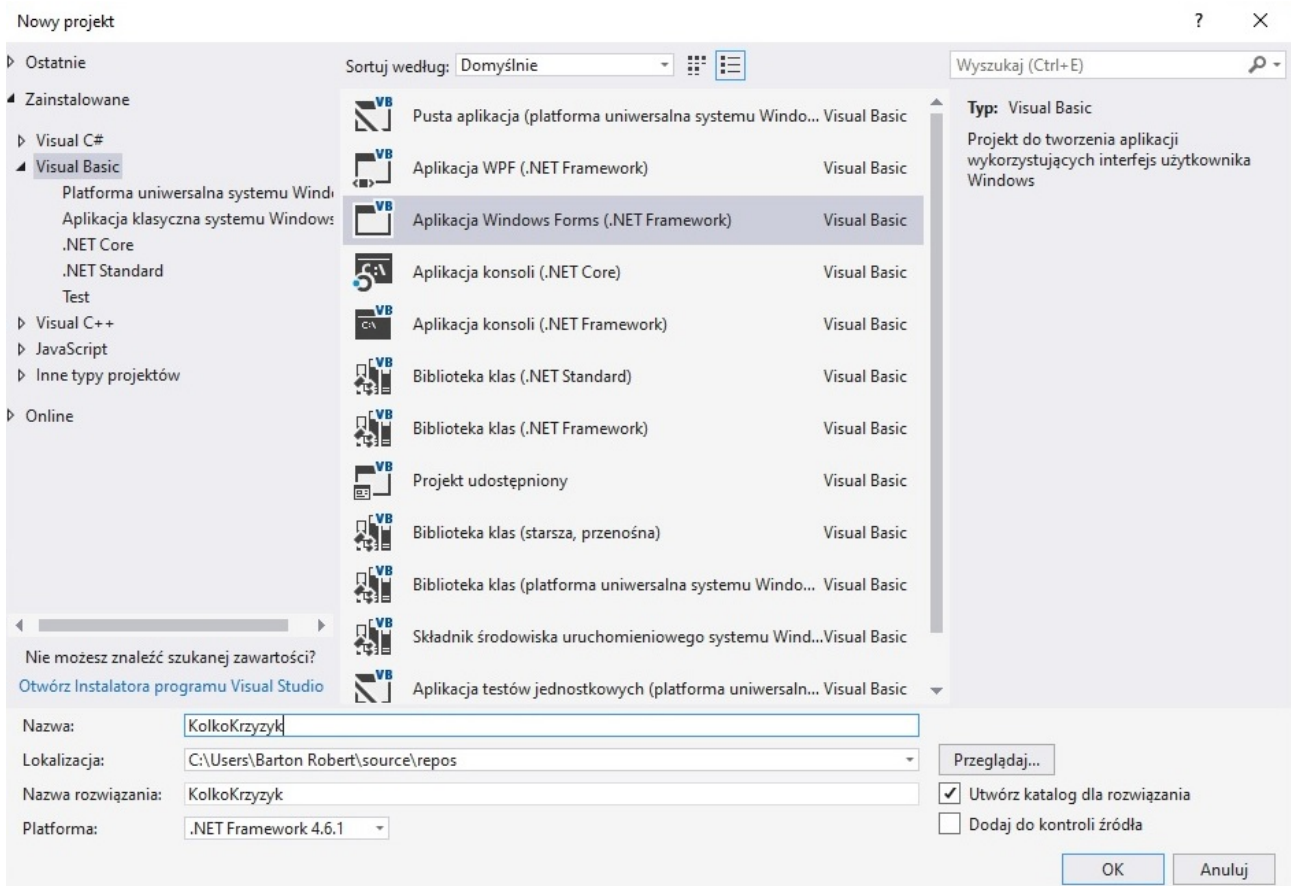
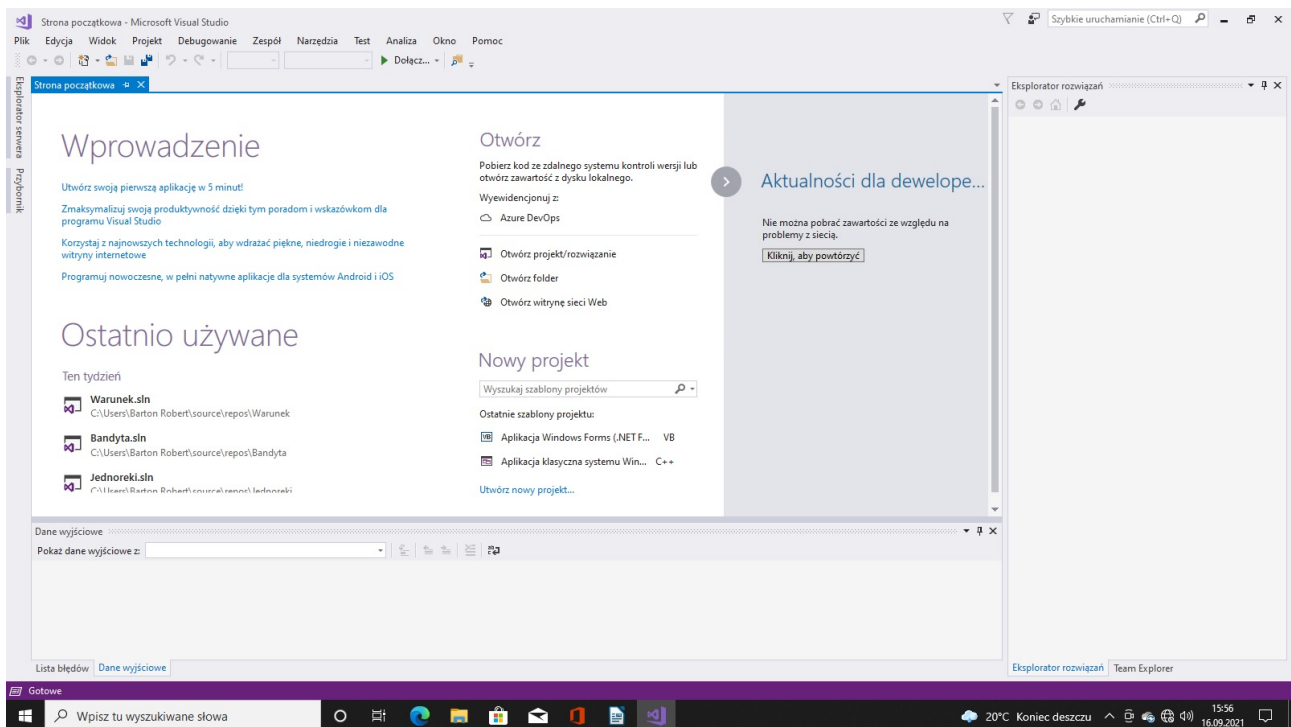
Algorytmy zawsze tworzy się, wykorzystując zdrowy rozsądek, choć może się okazać, że trzeba napisać kod dla algorytmu zawierającego skomplikowane rozumowanie matematyczne lub naukowe. Możliwe, że nie uznasz tego za algorytm zdroworozsądkowy, jednak będzie on taki dla jego autora. Zła wiadomość jest taka, że proces przekształcania algorytmu na kod może być trudny. Dla programisty nauczanie się tworzenia algorytmów to kluczowa umiejętność. Wszyscy dobrzy programiści uwzględniają to, że wybór języka programowania nie jest zbyt istotny. Różne języki są dobre do wykonywania odmiennych zadań. Język C++ daje programistom dużą kontrolę nad działaniem programów, jednak pisanie aplikacji w tym języku jest trudniejsze niż na przykład w Visual Basic 2017, podobnie jak tworzenie interfejsu użytkownika. Niektóre z problemów związanych z językiem C++ rozwiązuje jego zarządzana wersja udostępniana przez platformę .NET, dlatego powyższe stwierdzenia są mniej prawdziwe dziś niż jeszcze kilka lat temu. Jako programista musisz nauczyć się wykorzystywać różne języki do rozwiązywania różnych problemów w jak najwydajniejszy sposób. Choć na początku większość programistów koncentruje się na jednym języku, warto pamiętać, że różne języki pozwalają tworzyć najlepsze rozwiązania dla odmiennych problemów. W pewnym momencie do napisania programu w nowym języku może okazać się konieczne użycie podstawowych umiejętności związanych z projektowaniem algorytmów i kodowaniem.

## Informacje i dane

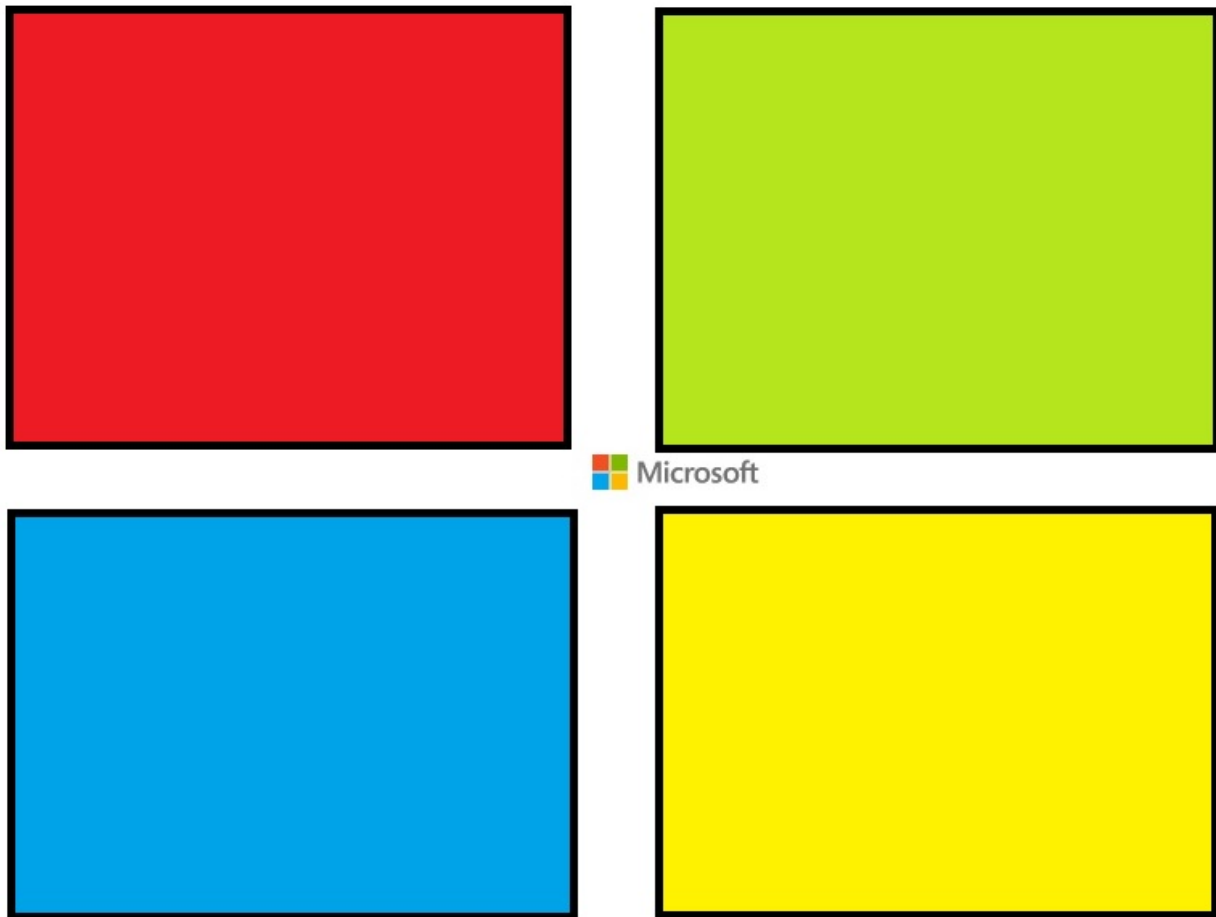
Informacje opisują fakty i można je przedstawiać oraz wyszukiwać w dowolnym formacie, niezależnie od tego, czy dany format bardziej nadaje się dla ludzi, czy dla komputerów. Na przykład, jeśli cztery osoby będą miały za zadanie mierzyć natężenie ruchu na czterech różnych skrzyżowaniach, po zakończeniu pracy przedstawią cztery ręcznie zapisane listy z liczbą samochodów, które przejechały w danym okresie (może to być jedna lista na każdą godzinę). Pojęcie dane służy do opisu informacji, które zostały zestawione, uporządkowane i sformatowane w taki sposób, aby możliwe było ich bezpośrednie wykorzystanie przez program komputerowy. Komputer nie może bezpośrednio użyć informacji zebranych przez osoby mierzące natężenie ruchu w postaci zestawu kartek pełnych odręcznych zapisków. Ktoś musi najpierw przekształcić te zapiski na dane. Na przykład można przepisać liczby do arkusza programu Excel, który można następnie bezpośrednio wykorzystać w programie zaprojektowanym do analizy wyników.

# Drugi Program.

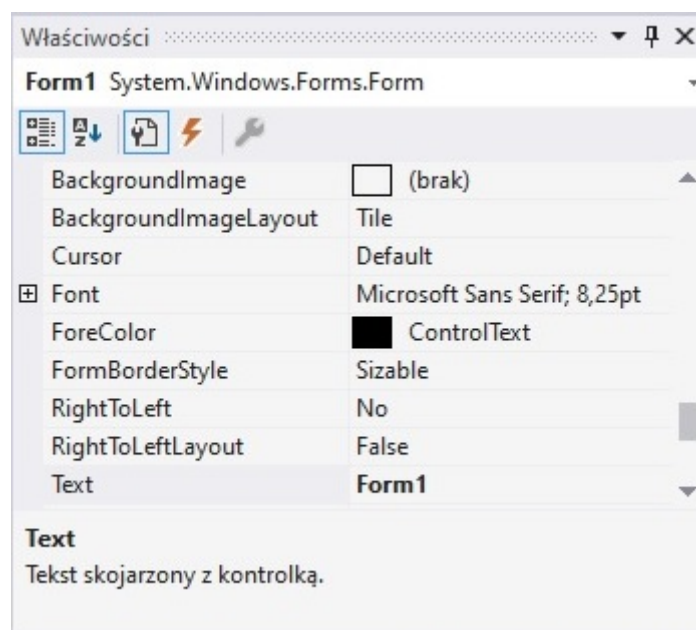
Napiszemy program który coś robi a ten to gra w kółko i krzyżyk ,zaczynamy od uruchomienia Visual Studio 2017 Community.



Nazwiemy projekt „KolkoKrzyzyk” dla Aplikacji Windows Forms (.NET Framework)



Prawy dolny róg pokazuje właściwości każdej kontrolki oraz formy wizualnej programu ,który można zmieniać np.: Text – Tytuł , Wielkość ,Kolor itp.



KOD:

```
Public Class Form1
    Dim a, b As Double

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Button4.Text = ""
        Button5.Text = ""
        Button6.Text = ""
        Button7.Text = ""
        Button8.Text = ""
        Button9.Text = ""
        Button10.Text = ""
        Button11.Text = ""
        Button12.Text = ""
        Beep()

    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        a = 0
        b = 1
        Beep()

    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        If a = 1 Then Button4.Text = "+"
        If b = 1 Then Button4.Text = "0"
        Beep()

    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        If a = 1 Then Button5.Text = "+"
        If b = 1 Then Button5.Text = "0"
        Beep()

    End Sub

    Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        If a = 1 Then Button6.Text = "+"
        If b = 1 Then Button6.Text = "0"
        Beep()

    End Sub

    Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
        If a = 1 Then Button7.Text = "+"
        If b = 1 Then Button7.Text = "0"
        Beep()

    End Sub

    Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
        If a = 1 Then Button8.Text = "+"
        If b = 1 Then Button8.Text = "0"
        Beep()

    End Sub

    Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
        If a = 1 Then Button9.Text = "+"
        If b = 1 Then Button9.Text = "0"
        Beep()

    End Sub

    Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
        If a = 1 Then Button10.Text = "+"
        If b = 1 Then Button10.Text = "0"
```



```

        Beep()
    End Sub

    Private Sub Button11_Click(sender As Object, e As EventArgs) Handles Button11.Click
        If a = 1 Then Button11.Text = "+"
        If b = 1 Then Button11.Text = "O"
        Beep()
    End Sub

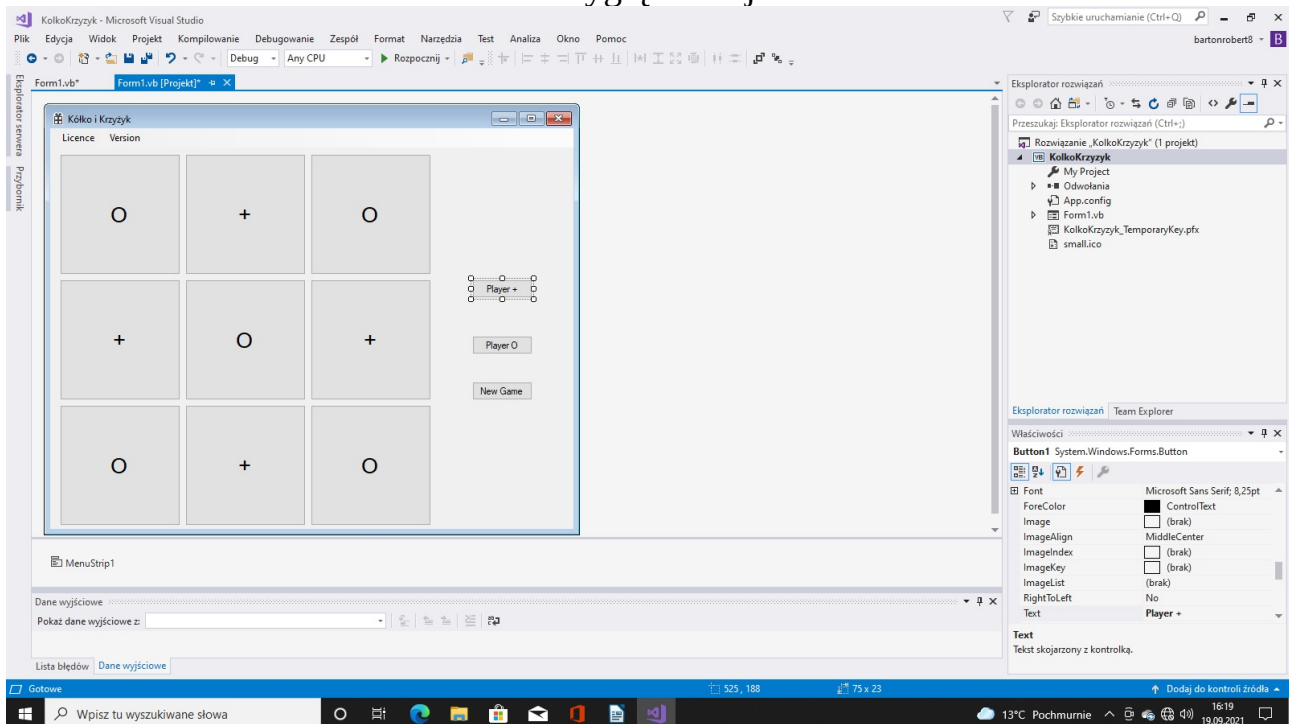
    Private Sub Button12_Click(sender As Object, e As EventArgs) Handles Button12.Click
        If a = 1 Then Button12.Text = "+"
        If b = 1 Then Button12.Text = "O"
        Beep()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        a = 1
        b = 0
        Beep()

    End Sub
End Class

```

## Tak Wygląda Projekt



Pierwsze trzy Buttony to Player + ,Player O oraz New Game .  
Reszta Buttonów to szachownica 3 X 3 ,Trzy rzędy na trzy Kolumny .

Koniec

Autor  
Bartoń Robert  
WWW  
bartonrobert8.hpage.com  
E-mail  
bartonrobert8@gmail.com

