

DateKeeper – kalendarz, terminarz

DateKeeper to kalendarz/terminarz, który umożliwia dodawanie, wyświetlanie jak i usuwanie wydarzeń dla danego dnia. Aplikacja desktopowa ta jest połączona z lokalną bazą danych SQL Server.

Uruchomienie aplikacji/Odczytanie kodu źródłowego

Do uruchomienia aplikacji potrzebny jest plik **DateKeeper.exe** oraz środowisko 'SQL Server 2022' opisane poniżej.

Do odczytania kodu źródłowego aplikacji niezbędne będą:

- Środowisko Visual Studio 2022 z pobranym rozszerzeniem 'C++ Windows Forms for VS 2022 .NET Framework' autorstwa Richard Kaiser
 - Link do rozszerzania: [C++ Windows Forms for VS 2022 .NET Framework - Visual Studio Marketplace](#)
 - Instrukcja instalacji: [cpp-2019-WinForms-Anleitung-EN.pdf \(rkaiser.de\)](#)



C++ Windows Forms for VS 2022 .NET Framework
Richard Kaiser | 57,938 installs | ★★★★★ (3) | Free

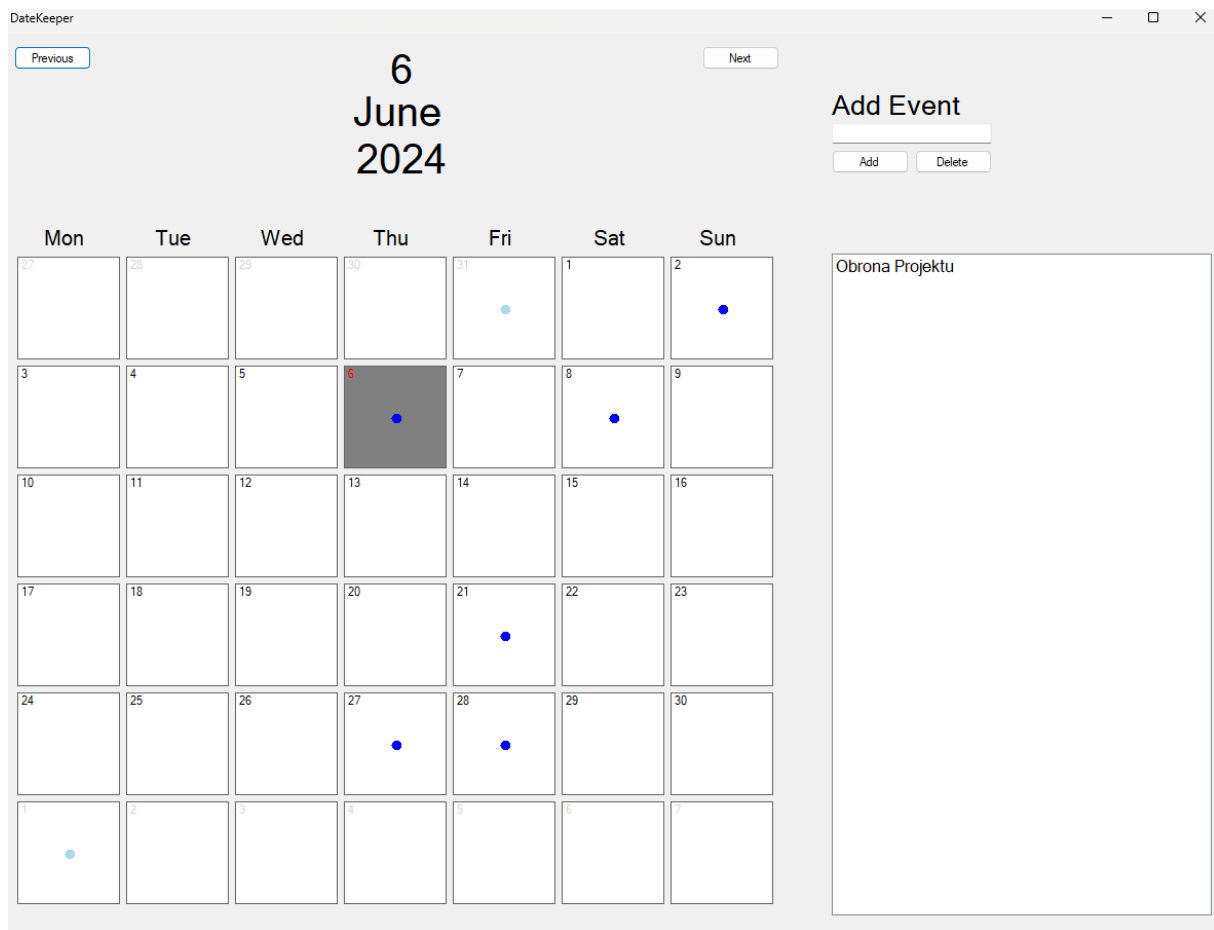
Extends Visual Studio by Windows Forms projects written in C++. WinForm apps are Windows applications with a graphical user interface (buttons, menus etc.) based on CLR. Use Standard C++ (including mo

[Download](#)

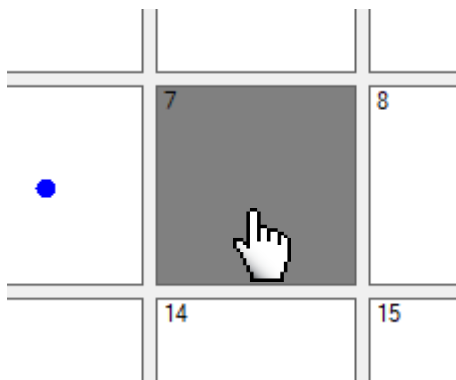
- Środowisko 'SQL Server 2022' oraz narzędzie do zarządzania bazą danych 'SSMS 20' która znajduje się w pliku **DataKeeper.sql**, który należy zaimportować za pomocą programu.
 - Link do 'SQL Server 2022': [SQL Server — pliki do pobrania | Microsoft](#)
 - Link do 'SSMS 20': [Download SQL Server Management Studio \(SSMS\) - SQL Server Management Studio \(SSMS\) | Microsoft Learn](#)



Instrukcja obsługi





Po otwarciu aplikacji pojawia się obecna data, wraz z kalendarzem oraz elementami dodawania lub usuwania wydarzeń, gdzie poniżej znajdują się wydarzenia przewidziane na wybraną datę. Datę wybiera się po przez kliknięcie jednego z kafelków pokazanym na kalendarzu

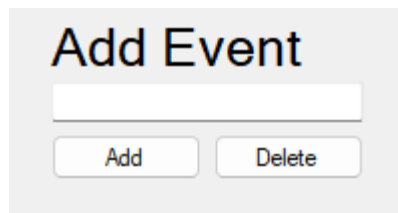


Niebieskie kropki informują czy w danym dniu są wydarzenia.

Opis przycisków:

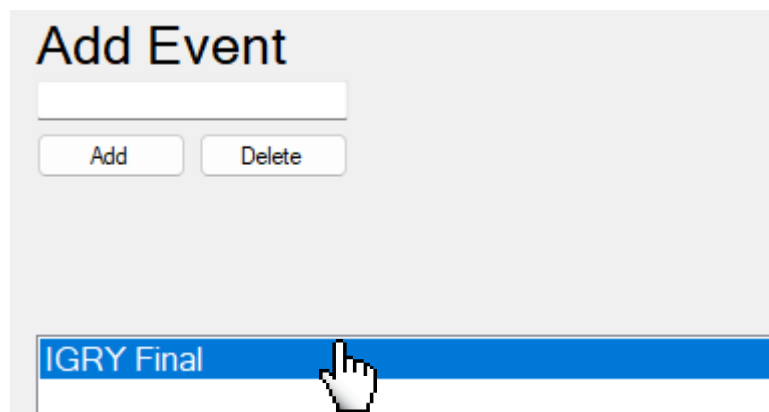
-  - odpowiada za przejście do poprzedniego miesiąca
-  - odpowiada za przejście do następnego miesiąca

Opis formularza:



The image shows a small, light gray rectangular box titled "Add Event" in a large, bold, black font. Below the title is a white text input field. Underneath the input field are two buttons: "Add" and "Delete", both with rounded corners and a light gray background.

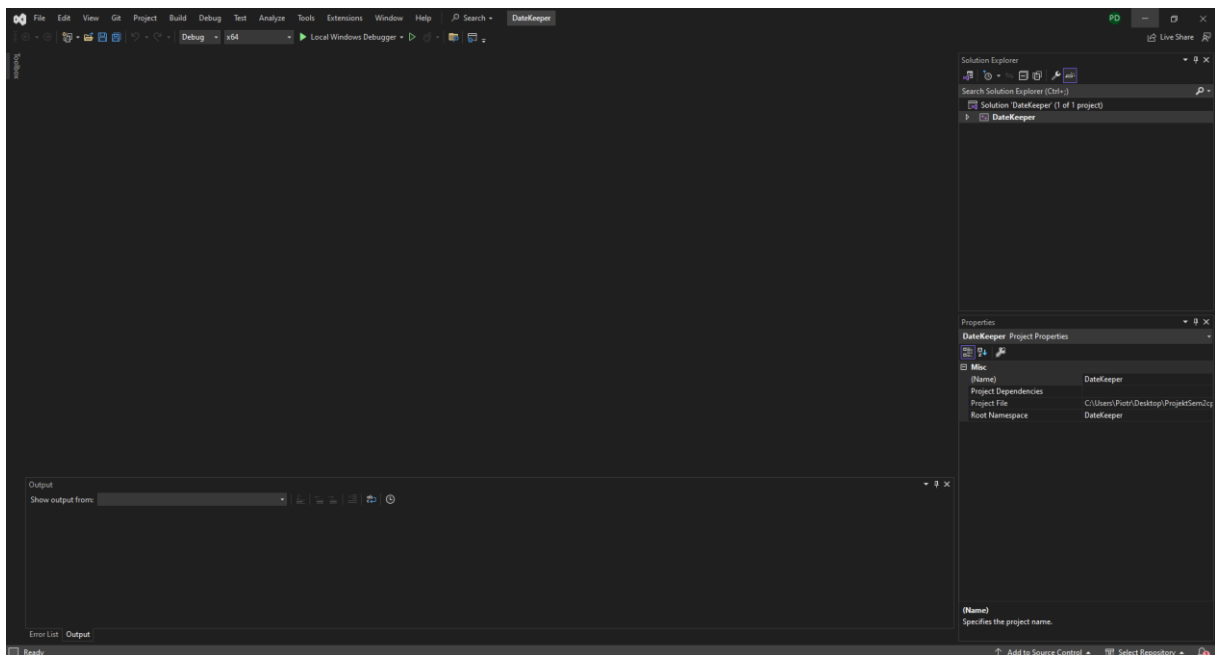
- - odpowiada za treść wydarzenia
 - - odpowiada za dodanie wydarzenia jeśli treść wydarzenia nie jest pusta
 - - odpowiada za usunięcie wydarzenia jeśli jest ono wybrane z listy poniżej
- np.:



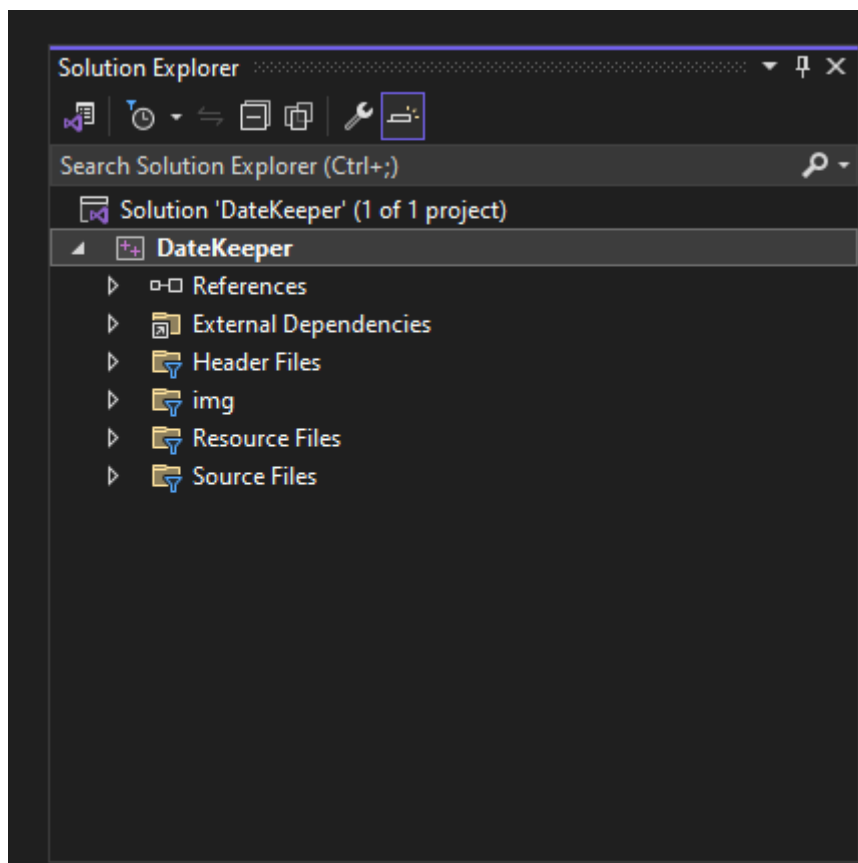
The image shows a larger version of the "Add Event" form. It has the same title, input field, and "Add" and "Delete" buttons. Below the buttons is a blue horizontal bar with the text "IGRY Final" in white. A white mouse cursor is pointing at the bar.

Kod źródłowy – nawigacja po projekcie

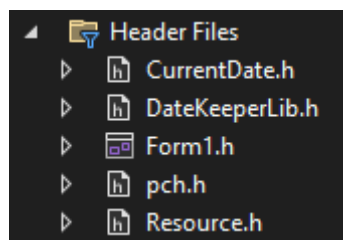
Po otwarciu pliku **DateKeeper.sln** powinno się ukazać taki widok:



Należy rozwinąć listę **DateKeeper** po prawej stronie:



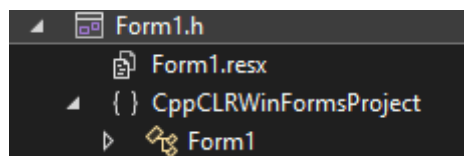
Header Files



Header Files zawiera:

- Nagłówek **CurrentDate.h** – deklaracja klasy która tworzy obiekt zawierający obecną datę.
- Nagłówek **DateKeeperLib.h** – deklaracja biblioteki funkcji często używanej w projekcie

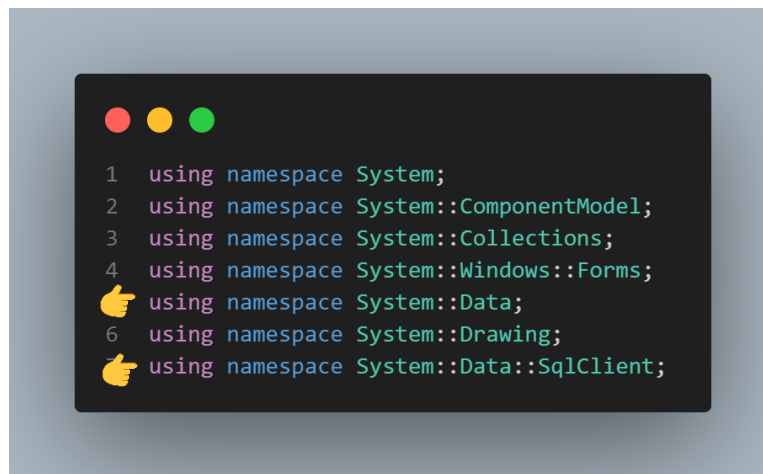
Form1.h:



- Po dwukrotnym kliknięciu **Form1.h** pokazuje się konstrukcja wyglądu aplikacji
- **Form1** – pokazuje zaplecze dla działania aplikacji

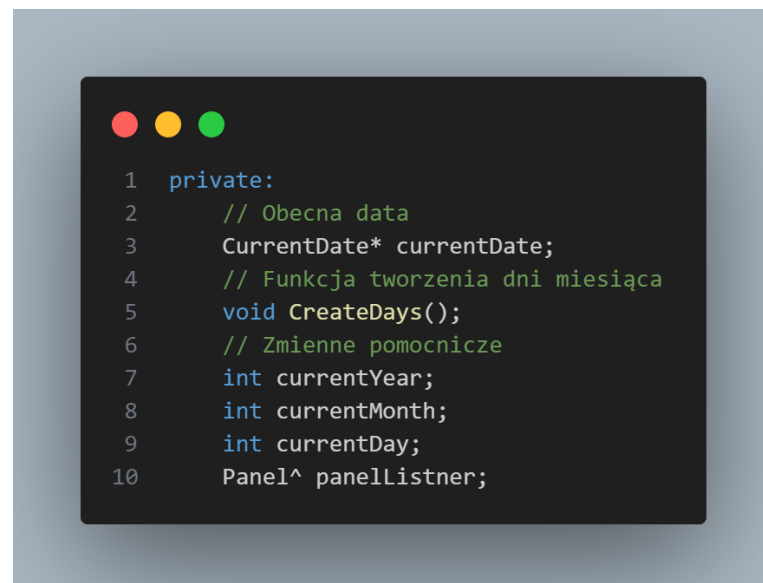
Opis kodu:

Form1:



'using' przedstawione w **Form1.h** odpowiadają za odpowiednie działanie aplikacji, można wyróżnić **using namespace System::Data;** oraz **using namespace System::Data::SqlClient;** które odpowiadają za połączenie z bazą danych.

Poniżej znajdują się deklaracje funkcji, obiektów użytych do działania aplikacji:



Wymienione powyżej deklaracje funkcji są wykorzystywane zaraz po inicjalizacji projektu oraz jego załadowaniu.

```

1  /*--- Funkcje przycisków ---*/
2
3      // Dodawanie elementów do listBox i bazy danych
4  void AddItemToListBox(String^ stringEvent);
5  // Czyszczenie eventów z listBox
6  void ClearEvents();
7  // Przejście do poprzedniego miesiąca
8  void handlePrevMonth();
9  // Przejście do następnego miesiąca
10 void handleNextMonth();
11 // Pokazanie eventów w listBox na zanaonczy panel z bazy danych
12 void ShowEvents();
13 // Sprawdzanie czy dzień może być oznaczony
14 bool dayHasEvent(std::tm timeToCheck);
15 // Dodawanie kropki
16 void addDot(Panel^ panel);
17 // Dodawanie kropki jaśniejszej
18 void addDotLighter(Panel^ panel);
19 // Usuwanie kropki
20 void Form1::removeDot(Panel^ panel);
21 void makePictureBoxRound(PictureBox^ pictureBox);

```

Te funkcje są używane w przyciskach. Działanie ich jak i `void CreateDays();` znajdują się w `Source Files → From1.cpp`.