Aplikacja klasyczna systemu Windows

source-code annotation language (SAL)

zestaw adnotacji, które można użyć do opisania, w jaki sposób funkcja wykorzystuje swoje parametry – dzięki temu kompilator może sprawdzić czy w kodzie nie ma błędów

```
/***
*sal.h - markers for documenting the semantics of APIs
*
*       Copyright (c) Microsoft Corporation. All rights reserved.
*
*Purpose:
*       sal.h provides a set of annotations to describe how a function uses its
*       parameters - the assumptions it makes about them, and the guarantees it makes
*       upon finishing.
*
*       [Public]
*
****/
```

source-code annotation language (SAL)

```
void * memcpy(
    void *dest,
    const void *src,
    size_t count
);
```

przykładowa funkcja

dokumentacja na temat funkcji

> "`memcpy` copies *count* bytes from *src* to *dest*; `wmemcpy` copies *count* wide characters (two bytes). If the source and destination overlap, the behavior of `memcpy` is undefined. Use `memmove` to handle overlapping regions.
>
> **Important:** Make sure that the destination buffer is the same size or larger than the source buffer. For more information, see Avoiding Buffer Overruns."

adnotacja SAL

kompilator nie umie czytać dokumentacji, dlatego wpisujemy adnotacje

```
void * memcpy(
    _Out_writes_bytes_all_(count) void *dest,
    _In_reads_bytes_(count) const void *src,
    size_t count
);
```

adnotacja SAL

Understanding SAL

source-code annotation language (SAL)

dzięki adnotacjom SAL kompilator wychwyci off-by-one błąd – wyjście poza zakres tablicy (powinno być: i < count;)

```
wchar_t * wmemcpy(
    _Out_writes_all_(count) wchar_t *dest,
    _In_reads_(count) const wchar_t *src,
    size_t count)
{
    size_t i;
    for (i = 0; i <= count; i++) { // BUG: off-by-one error
        dest[i] = src[i];
    }
    return dest;
}
```

Understanding SAL

source-code annotation language (SAL)

SAL Basics

SAL defines four basic kinds of parameters, which are categorized by usage pattern.

Category	Parameter Annotation	Description
Input to called function	`_In_`	Data is passed to the called function, and is treated as read-only.
Input to called function, and output to caller	`_Inout_`	Usable data is passed into the function and potentially is modified.
Output to caller	`_Out_`	The caller only provides space for the called function to write to. The called function writes data into that space.
Output of pointer to caller	`_Outptr_`	Like **Output to caller**. The value that's returned by the called function is a pointer.

Understanding SAL

Aplikacja klasyczna systemu Windows

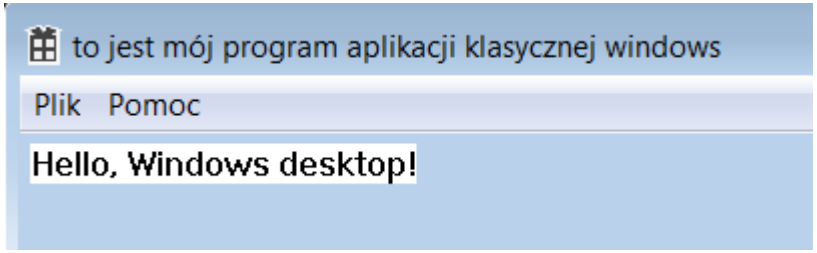
Zadanie:

- Stworzyć projekt aplikacji klasycznej systemu windows AplikacjaKlasyczna
- Przejrzeć przewodnik wyjaśniający szczegóły kodu aplikacji:

Przewodnik: tworzenie tradycyjnej aplikacji klasycznej Windows klasycznej (C++)

- wykonać zadania:

    - zmienić kolor tła aplikacji (windows form)
    - zmienić tytuł aplikacji na pasku tytułowym na:
      "to jest mój program aplikacji klasycznej windows"
    - dodać napis na formie aplikacji "Hello, Windows desktop!"
    - zamienić okno dialogowe menu pomoc/informacje na własny MessageBox



Aplikacja klasyczna systemu Windows