

Interfejsy aplikacji w środowisku Windows

Aplikacje z wywołaniem parametrycznym

Przekazywanie parametrów wywołania

1. Przykład

Poniższy program pokazuje sposób obsługi parametrów wywołania.

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char **argv)
{
    cout << "Liczba argumentow: " << argc << endl << endl;

    for(int i = 0; i < argc; i++)
        cout << "Argument " << i << ": " << argv[i] << endl;

    return EXIT_SUCCESS;
}
```

- Zmienna **argc** zawiera liczbę argumentów wywołania a dwuwymiarowa tablica znakowa **argv** zawiera nazwę programu i jego parametry.
- Aby można było debugować program pod kontrolą środowiska programistycznego, należy dodać parametry wywołania w ustawieniach projektu:
 - zaznacz projekt w oknie **Solution Explorer**, wybierz z menu **Project | Properties**,
 - następnie w lewym panelu okna **Configuration Properties | Debugging**,
 - w prawym panelu ustaw opcję **Command Arguments** na **-a bb "c c c" 1 2 33 3.141 2,718** i zaakceptuj zmiany przyciskiem **OK**.
- Przepisz, uruchom i przetestuj program. Zwróć uwagę na wynikowe parametry przekazane do programu.

2. Przykład

Poniższy program sumuje liczby całkowite podane w parametrach wywołania.

```
#include <cstdlib>
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char **argv)
{
    long suma = 0;
    string napis;

    if(argc < 3)
    {
        cout << "Podaj minimum dwa argumenty" << endl;
        return EXIT_FAILURE;
    }
}
```

```
try
{
    for(int i = 1; i < argc; i++)
    {
        napis = argv[i];
        suma += stoi(napis);
    }
    cout << "Suma liczb wynosi: " << suma << endl;
}
catch(exception const &e)
{
    cout << "Bład: " << e.what() << endl;
    return EXIT_FAILURE;
}

return EXIT_SUCCESS;
}
```

- Dodaj odpowiednie parametry wywołania w ustawieniach projektu.
- Przepisz, uruchom i przetestuj działanie programu dla odpowiednich parametrów z wykorzystaniem środowiska programowania.
- Przetestuj działanie programu z odpowiednimi i nieodpowiednimi parametrami uruchamiając go z linii poleceń (uruchom konsolę CLI, zmień katalog roboczy i uruchom program wpisując jego nazwę).

3. Zadanie.

Popraw program tak, żeby wyświetlał pełne wyrażenie arytmetyczne:

```
1 + 2 + 3 = 6
```

4. Zadanie.

Popraw program z zadania 3 tak aby pracował z liczbami zmiennoprzecinkowymi.

5. Przykład.

Program do pracy z datami.

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <iomanip>
#include "iadata.h"

using namespace std;

const string nazwaMiesieca[] = {"", "Styczen", "Luty", "Marzec",
                                "Kwiecien", "Maj", "Czerwiec",
                                "Lipiec", "Sierpień", "Wrzesień",
                                "Pazdziernik", "Listopad", "Grudzien"};

int main(int argc, char **argv)
{
    string napis;
    int miesiac, rok;

    napis = argv[1];
    miesiac = stoi(napis);
}
```

```

    napis = argv[2];
    rok = stoi(napis);

    cout << "Miesiac: " << miesiac << endl;
    cout << "Rok: " << rok << endl << endl;

    cout << "Nazwa miesiaca: " << nazwaMiesieca[miesiac] << endl << endl;

    cout << "Wynik czyPrzestepny(): " << czyPrzestepny(rok) << endl;
    cout << "Wynik dniMiesiaca(): " << dniMiesiaca(miesiac, rok) << endl;
    cout << "Wynik dzienRoku(): " << dzienRoku(1, miesiac, rok) << endl;
    cout << "Wynik dzienTygodnia(): " << dzienTygodnia(1, miesiac, rok) << endl << endl;

    return EXIT_SUCCESS;
}

```

Program do poprawnego działania wymaga włączenia dodatkowej biblioteki (dyrektywa **#include "iadata.h"**).

- Pobierz bibliotekę ze strony internetowej przedmiotu i przegraj do katalogu projektu.
- Dodaj bibliotekę do projektu: zaznacz projekt w oknie **Solution Explorer** i wybierz z menu **Project | Add Existing Item...**, w oknie dialogowym wybierz oba pliki biblioteki (**iadata.cpp**, **iadata.h**), zaakceptuj wybór. Pliki powinny się pokazać w drzewie projektu w oknie **Solution Explorer** (foldery: **Header Files** i **Source Files**).
- Skompiluj i przetestuj program, program należy uruchamiać z parametrami w postaci numeru miesiąca (bez zera dla miesięcy jednocyfrowych) i roku, np.: **3 2021**.

6. Zadanie.

Popraw zadanie z poprzedniego punktu tak, aby wyświetlał w konsoli kartkę z kalendarza:

```

    Marzec 2020
    Pn Wt Sr Cz Pt So Nd
          1
    2  3  4  5  6  7  8
    9 10 11 12 13 14 15
   16 17 18 19 20 21 22
   23 24 25 26 27 28 29
   30 31

```

- Wydruk kalendarza umieść w dodatkowej funkcji **void drukujMiesiac(int m, int r)**.
- Zwróć uwagę, że nazwa miesiąca i rok są wyśrodkowane.
- Sprawdź w jaki dzień się urodziłeś.