

Część 3 C++

Będziemy programować w języku C++ za pomocą aplikacji Dev-C++ .

Poznamy sposób w jaki sposób wygenerować **kod źródłowy** najprostszego programu.

Dowiemy się, w jaki sposób dołączyć niezbędne biblioteki, co to jest **funkcja główna** oraz jak wyświetlić na ekranie dowolny komunikat.

Nauczmy się również komentować kod źródłowy oraz schemat blokowy. Dzięki temu w przyszłości łatwiej zmodyfikujemy nasz program. Przekonamy się też, że błędy w kodzie nie są wcale takie straszne - spróbujemy **skompilować** program z umyślnie wstawionym błędem i zobaczymy, jak radzić sobie z nieprawidłowościami w kodzie.

Ćwiczenie nr 1 –utworzenie nowego projektu

- 1.na pulpicie tworzymy folder o nazwie **nazwisko1** (wstaw swoje nazwisko i jedynekę)
- 2.uruchamiamy aplikację Dev-C++ .
2. tworzymy w niej nowy projekt (z menu **plik** wybieramy podmenu **Nowy** i pozycję **projekt**).
- 3 Pojawia się okno, w którym klikamy na ikonę **Console Application**(oznacza ona, że będziemy tworzyć aplikację konsolową, czyli uruchamianą w oknie MS-DOS-u).
- 4.W pole **nazwa** wpisujemy nazwę projektu tj. nazwisko1
- 5.potwierdzamy **ok**.
6. Pojawia się okno **Create new Project** i zapisujemy na pulpicie w folderze z swoim nazwiskiem
7. Projekt jest gotowy i widzimy szkielet naszej aplikacji automatycznie wygenerowany przez program.

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

using namespace std;

int main(int argc, char *argv[])
{
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

8. Analiza pierwszego programu

Dla ułatwienia pracy programisty edytor Dev--C++ koloruje kod źródłowy. Na przykład wszystkie polecenia zaczynające się od znaku # oznaczone są **kolorem zielonym**, znaki umieszczone między cudzysłowami kolorowane są na **czerwono**. Warto również zauważyć, że wszystkie słowa kluczowe (wyrazy, które mają szczególne znaczenie w C++) zostają **pogrubione**. Dzięki kolorowaniu kodu jest on bardziej przejrzysty.

9. Bardzo ważny fragment naszego szkieletu kodu źródłowego to linijki 3 . Pierwsza linia to definicja tak zwanej funkcji głównej (która zawsze nazywa się **main** -z ang. główna).

Wewnątrz funkcji **main** znajduje się **kod główny** naszego programu. Wewnątrz funkcji - czyli

między znakiem rozpoczynającym nawias klamrowy { umiejscowionym bezpośrednio po linijce z definicją funkcji a znakiem zamykającym nawias klamrowy }

10. W wierszu 4 zostały wstawione automatycznie polecenia przez aplikację Dev-C++ charakterystyczne jedynie dla aplikacji konsolowych uruchamianych w Windows. Ponieważ nasz program będzie uruchamiany w oknie MS-DOS-u (stworzyliśmy projekt aplikacji konsolowej), musimy sprawić, aby po zakończeniu wykonywania wszystkich poleceń okno nie zostało od razu zamknięte. Polecenie `system („PAUSE”)` nakazuje więc systemowi wstrzymać zamknięcie okna z efektami pracy naszego programu.

Ostatnie polecenie funkcji `main return EXIT_SUCCESS` informuje system, że wykonanie instrukcji znajdujących się wewnątrz niej zakończyło się sukcesem.

Ćwiczenie 2 Skompilowanie programu i uruchomienie

1. Szkielet naszego programu składa się z kilku linijek kodu

2. Kompilujemy kod źródłowy wybierając z menu polecenie `Uruchom` a następnie `Kompiluj`, widzimy postęp kompilacji

3. Po zakończeniu kompilacji pojawia się napis `Done`. (zrobiono), a w polach `Errors` i `Warnings` zostaje podana liczba błędów, które wystąpiły w trakcie kompilacji, oraz liczba ostrzeżeń, na które warto zwrócić uwagę. Skompilowanie wygenerowanego automatycznie szkieletu programu powinno zakończyć się bez błędów.

4. Po skompilowaniu programu możemy go uruchomić. Z menu `Uruchom` wybieramy pozycję `Uruchom`. Na ekranie pojawi się okno MS-DOS-u z komunikatem:

`Aby kontynuować, naciśnij dowolny klawisz` – wywołuje go polecenie `system („PAUSE”)` wstrzymujące zamknięcie okna konsoli. Dzięki temu widzimy efekt działania programu.

5. Otwieramy folder z nazwiskiem (na pulpicie) w którym zapisaliśmy plik z kodem źródłowym. W środku powinny być:

- plik projektu z rozszerzeniem `.dev`
- plik z kodem źródłowym `.cpp`
- plik wykonywalny naszego programu `.exe`

oraz

- plik `main.o` – plik binarny skompilowany, efekt pracy kompilatora
- plik `Makefile.win` – zbiór reguł dla kompilatora wykorzystywanych przy kompilacji programu

Ćwiczenie nr 3 Błędy kompilacji

1. Pierwszy program skompilował się bez błędów. Jednak musimy się przyzwyczaić do myśli, że prawie nigdy nie będzie tak dobrze i przed zobaczeniem efektów naszego programu będziemy musieli poświęcić sporo czasu na usunięcie błędów z kodu źródłowego. Warto więc już teraz zobaczyć, co robić, jeśli kompilator zakomunikuje nam, że kompilacja z powodu występujących błędów nie została zakończona sukcesem.

2. Na początku wprowadzimy drobną zmianę do naszego kodu - usuwamy na przykład średnik (;) znajdujący się na końcu linii `system („PAUSE”)`

3. Włączamy kompilację kodu (z menu `Uruchom`) i wybieramy `Kompiluj`

4. Tym razem nie pojawi się okno informacji o zakończonej sukcesem kompilacji.

5. W wyświetlonym oknie Dev C++ przyjrzyj się komunikatom - nr wiersza oraz zaznaczeniu na czarno w kodzie źródłowym.

6. Popraw błąd (usuń średnik) i ponownie skompiluj program. Zamknij program Dev C++

Całe ćwiczenie przeprowadź od nowa - zakładając nowy folder na pulpicie z twoim nazwiskiem2.

Po zakończeniu ćwiczenia utwórz folder o nazwie nazwiskocplusplus umieść w nim dwa swoje foldery a następnie umieść folder nazwiskocplusplus w folderze sieciowym twojej grupy (klasy).