Podstawy programowania Uwagi wstępne

Uwagi

- W zależności od używanego kompilatora i standardu języka, może się zdarzyć konieczność używania nieco innych funkcji lub importowania innych plików nagłówkowych.
- Do wersji **C99** do obsługi wejścia i wyjścia należy używać *tradycyjnych* funkcji **printf()** i **scanf()** we wszystkich popularnych kompilatorach (zarówno MSVC i GCC).
- W standardzie C11 i późniejszych została wprowadzona możliwość implementacji tzw. bezpiecznych (secure) funkcji obsługi standardowego wejścia i wyjścia (np. zapewnia mechanizm ochrony przed przepełnieniem bufora). Nie są one jednak obligatoryjne. Takie funkcje są dostępne w kompilatorze MSVC (firmy Microsoft) ale nie są dostępne w kolekcji GCC (projekt GNU).
- W przypadku korzystania z MSVC należy korzystać z funkcji printf_s() i scanf_s() lub dodać dyrektywę: #define _CRT_SECURE_NO_WARNINGS.
- Listingi będą prezentowane w wersji z użyciem *tradycyjnych* funkcji wejścia i wyjścia.

Środowisko Linux

- W systemie plików dysku wskazanym przez prowadzącego (prawdopodobnie **/home/pol**) utwórz katalog o nazwie **CodeBlocks**, który będzie zawierał projekty aplikacji.
- Pamiętaj, aby każdy nowy projekt był zapisywany w *nowym katalogu o unikalnej* nazwie, najlepiej zawierającej elementy wiążące nazwę katalogu z autorem projektu (np. inicjały lub nazwisko), nie można nadpisywać istniejących projektów.
- Uruchom program Code::Blocks IDE.
- Zmień domyślny terminal do uruchomiania skompilowanych programów, użyj opcji Settings | Environment, następnie w panelu po lewej stronie wybierz opcję General settings i ustaw opcję Terminal to launch console programs: na gnome-terminal --title=\$TITLE -x.

Środowisko Windows

- Na dysku wskazanym przez prowadzącego (prawdopodobnie D:\) utwórz katalog o nazwie **VisualStudio**, który będzie zawierał projekty aplikacji.
- Pamiętaj, aby każdy nowy projekt był zapisywany w *nowym katalogu o unikalnej nazwie*, najlepiej zawierającej elementy wiążące nazwę katalogu z autorem projektu (np. inicjały lub nazwisko), nie można nadpisywać istniejących projektów.
- Uruchom program *Microsoft Visual Studio 2017* lub *Microsoft Visual Studio 2019*.
- Jeśli środowisko programistyczne uruchomi się w języku polskim, zmień język na angielski, wybierz Narzędzia | Opcje..., następnie w panelu po lewej stronie wybierz opcję Środowisko | Ustawienia międzynarodowe i wybierz język English, uruchom ponownie środowisko.
- Zmień katalog przechowywania projektów, użyj opcji Tools | Options..., następnie w panelu po lewej stronie wybierz opcję Projects and Solutions | Locations i ustaw opcję Projects location na D: \VisualStudio.

Korzystanie z debugera w Code::Blocks IDE

 Krokowe uruchomienie programu: opcja Debug | Step into (klawisze [SHIFT + F7]) i następnie Debug | Next line (klawisz [F7]). Możesz przerwać krokowe wykonywanie programu za pomocą Debug | Stop Debbuger (klawisze [SHIFT + F8]).

Korzystanie z debugera w Microsoft Visual Studio

Krokowe uruchomienie programu: opcja Debug | Step Over (klawisz [F10]). Po zakończeniu śledzenia, powtórz zadanie z wykorzystaniem Debug | Step Into (klawisz [F11]). Jakie zauważyłeś różnice? Możesz przerwać krokowe wykonywanie programu za pomocą Debug | Stop Debbuging (klawisze [SHIFT + F5])