

Zajęcia 1. Wprowadzenie

1. Zapoznanie się ze środowiskiem pracy

Praca w sieci lokalnej

Laboratorium prowadzone jest na komputerach wyposażonych w system operacyjny **Linux** z Oracle VM VirtualBox. Ponadto katalogi robocze są umieszczone na serwerze plikowym pracującym pod kontrolą systemu **Linux**. Aby przystąpić do zajęć laboratoryjnych, należy uruchomić program Oracle VM VirtualBox Manager, a następnie wybrać maszynę wirtualną **Lab_A300_VS_Community** i wcisnąć ikonę **Start**.

Sposób logowania

Aby się zalogować do systemu należy wykonać następujące czynności:

1. Wybrać: **Inny użytkownik** (w dolnym lewym rogu)
2. W miejscu **Nazwa użytkownika** należy podać domenę oraz identyfikator: **TD1labvxyz**, gdzie **vx** oznacza numer grupy laboratoryjnej; **yz** oznacza numer zespołu w grupie (przykładowy identyfikator **TD1lab0205** – co oznacza grupę L02, stanowisko nr 5)
3. W miejsce **Hasło** wpisujemy: student

Uwaga! Identyfikatory **TD1labvxyz** zostaną nadane poszczególnym studentom przez prowadzącego zajęcia. Są ważne do końca semestru i powiązane z danym studentem, a nie z konkretnym komputerem.

Katalogi robocze

Pliki robocze (źródłowe i pliki projektu) powinny się znaleźć w katalogach umieszczonych na serwerze plikowym na dysku **H:**.

Praca w środowisku Visual Studio

Aby uruchomić środowisko programistyczne należy kliknąć na ikonę **Visual Studio 2022**, która znajduje się na pulpicie lub pasku zadań.

Tworzenie projektu

Visual Studio 2022 (VS 2022) nie jest środowiskiem programistycznym bezpośrednio przeznaczonym do programowania w języku C. Bazuje ono na platformie **.NET**, która nie wspiera składni oraz zaplecza bibliotecznego tego języka. Twórcy VS 2022 udostępnili jednak możliwość programowania w języku C. Wykorzystanie języka C może być zrealizowane przy użyciu prostych programów w ramach projektów, których efekt działania przedstawiony jest w oknie linii poleceń. Szablony takich projektów mogą być różne. Dla uproszczenia na zajęciach wykorzystywany będzie **Pusty projekt**, który nie oferuje żadnych plików początkowych.

Aby utworzyć nowy **Pusty projekt** należy:

- kliknąć w kafelek **Utwórz nowy projekt** w widoku głównym VS 2022;
- z menu języków programowania wybrać **C++**;
- z menu platformy wybrać **Windows|Konsola**;
- zaznaczyć docelowy szablon **Pusty projekt**;
- kliknąć przycisk **Dalej**.

W ramach końcowej konfiguracji, konieczne jest podanie nazwy projektu (w niniejszym przykładzie **PierwszyProjekt**), nazwy rozwiązania (np. **Zajęcia**) oraz lokalizacji – dysk sieciowy **H:**. Opcję **Umieść rozwiązanie i projekt w tym samym katalogu** należy pozostawić niezaznaczoną. Po poprawnym

wprowadzeniu wszystkich w/w ustawień wymagane jest zatwierdzenie etapu tworzenia projektu przyciskiem **Utwórz**.

Po utworzeniu nowego rozwiązania, z menu lokalnego **Projekt** należy wybrać opcję **Dodaj nowy element**. Ukaże się wówczas okno dialogowe służące do dodania do projektu **PierwszyProjekt** pliku umożliwiającego pisanie kodu w języku C++. Można to również osiągnąć klikając prawym klawiszem myszki na nazwie projektu w panelu o nazwie **Eksplorator rozwiązań**. Kolejne kroki sprowadzają się do:

- zaznaczenia typu pliku – **Plik C++ (.cpp)**,
- podania nazwy pliku z rozszerzeniem ***.c** w polu nazwy,
- wskazania lokalizacji (zawartość tego pola musi pozostać domyślna),
- zatwierdzenia wszystkich operacji za pomocą przycisku **Dodaj**.

Uwaga: pozostawienie nazwy pliku z rozszerzeniem ***.cpp** zablokuje możliwość tworzenia kodu w języku C.

Pionowy panel po lewej stronie prezentuje strukturę projektu z poziomu plików (zakładka **Eksplorator rozwiązań**) oraz z poziomu klas/funkcji (zakładka **Widok klas**). Duże okno edycyjne z zakładką **Źródło.c** służy do pisania kodu, natomiast podłużny panel **Dane wyjściowe** u dołu daje m.in. możliwość przeglądania informacji wygenerowanych przez środowisko po etapie kompilacji (np. na temat błędów lub ostrzeżeń). Panel ten może być również wykorzystany do weryfikacji kodu w trakcie pracy krokowej.

Kompilacja

Narzędzia umożliwiające kompilację i konsolidację znajdują się w menu **Kompilowanie**. Poszczególne pliki źródłowe można kompilować z osobna poleceniem **Kompiluj** (po poprzednim wskazaniu pliku) albo wybrać **Kompiluj rozwiązanie**, aby przeprowadzić proces kompilacji i konsolidacji całego projektu. Istnieje również możliwość wyboru opcji **Kompiluj ponownie rozwiązanie**, aby skompilować ponownie wszystkie pliki źródłowe i przeprowadzić wtórną konsolidację. Wyniki kompilacji i konsolidacji są pokazywane w panelu **Dane wyjściowe**. W przypadku pojawienia się błędu w kodzie, można dwukrotnie kliknąć na linii z komunikatem o błędzie, co spowoduje przeniesienie kursora w panelu edycyjnym do odpowiedniej linii w celu skorygowania błędu.

Uruchamianie programu

Aby uruchomić stworzony program wystarczy z menu **Debugowanie** wybrać polecenie **Uruchom bez Debugowania**. Rezultat działania widoczny jest w postaci „okienka konsolowego”. Jeżeli program zostanie uruchomiony za pomocą opcji **Uruchom bez Debugowania**, po zakończeniu działania programu proces aplikacji jest wstrzymywany i oczekiwana jest reakcja użytkownika, aby ten proces zakończyć (np. wciśnięcie dowolnego przycisku klawiatury).

Praca krokowa

Praca krokowa polega na wstrzymywaniu działania programu po wykonaniu instrukcji zakodowanych w jednej linii. Rozpoczęcie tego trybu pracy odbywa się przez wybranie polecenia **Wkrocz do** (skrót **F11**) z menu **Debugowanie**. Wykonanie pojedynczej linii programu odbywa się przez polecenie **Przekrocz nad** (skrót **F10**). Można również korzystać z palety uruchomieniowej menu **Debugowanie**. Warto również pamiętać o możliwości jaką daje skrót klawiszowy **Ctrl+F10**: po umieszczeniu kursora w dowolnej linii kodu i wciśnięciu takiej kombinacji klawiszy kompilator automatycznie przenosi nas do tej linii wykonując wcześniej napotkanie instrukcje.

W czasie pracy krokowej dolny panel zawiera listę zmiennych (obiektów) tworzonych w trakcie działania programu umożliwiając śledzenie zmian ich wartości. Ponadto w panelu **Wyrażenia kontrolne** można utworzyć własną listę zmiennych do podglądnięcia. Zakończenie pracy krokowej i powrót do trybu edycji odbywa się przez wybranie polecenia **Zatrzymaj debugowanie** z menu **Debugowanie** albo z palety uruchomieniowej.

2. Opracowanie prostych programów w języku C

1. Dodaj do projektu plik **pierwszy.c**, zdefiniuj w nim funkcję **main**, a następnie napisz kod pozwalający wprowadzić dowolną liczbę całkowitą z klawiatury i wypisać ją na ekran. Zmodyfikuj powyższy kod tak, aby na ekranie wyświetliła się informacja, czy wczytana liczba całkowita jest parzysta.
2. Dodaj do projektu plik **drugi.c** i zdefiniuj w nim funkcję **main**. Napisz w nim kod umożliwiający wczytanie znaku z klawiatury. Jeżeli wczytany znak jest małą literą, zamień go na dużą i odwrotnie. Na koniec wypisz znak na ekran.
3. Dodaj do projektu plik **pierwiastki.c**, zdefiniuj w nim funkcję **main**, a następnie opracuj algorytm pozwalający rozwiązywać równanie kwadratowe $ax^2 + bx + c = 0$. Współczynniki równania mają być wprowadzane z klawiatury.
4. Dodaj do projektu plik **echo.c**, zdefiniuj w nim funkcję **main** i zaprezentuj kod wypisujący na ekran dowolny napis w taki sposób, aby każda litera pojawiała się **n** razy, gdzie **n** jest liczbą całkowitą większą od 0.

Uwagi końcowe:

1. W projekcie wymagane jest dołączenie odpowiednich plików nagłówkowych **<stdio.h>** oraz **<math.h>** za pomocą dyrektywy **#include**.
2. W projekcie może znajdować się wyłącznie jeden plik z funkcją główną **main**. W związku z tym nie będzie można dodać kolejnych funkcji głównych do istniejącego projektu. Problem można rozwiązać przez dodanie do już istniejącego rozwiązania nowego projektu (np. o nazwie **DrugiProjekt**) i umieszczenie w nim pliku **drugi.c** z nową funkcją **main**. Dzięki takiemu podejściu będzie można „przełączać się” pomiędzy projektami ustawiając jeden lub drugi jako początkowy – opcja **Ustaw jako projekt startowy**.