Programowanie obiektowe C++ w środowisku Windows

Komunikacja z użytkownikiem, proste algorytmy obliczeniowe

Uwagi

- 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*.
- 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.

Komunikacja z użytkownikiem, proste algorytmy obliczeniowe

1. Przykład

Poniższy przykład pokazuje sposób obsługi formatowanego wejścia i wyjścia.

```
#include <stdlib.h>
#include <stdio.h>

int main()
{
  int wiek;
  char imie[80];

  printf_s("Podaj swoje imie: ");
  scanf_s("%s", imie, 80);

  printf_s("Podaj swoj wiek: ");
  scanf_s("%d", &wiek);

  printf_s("Czesc %s, tez mam %d lat", imie, wiek);

  return EXIT_SUCCESS;
}
```

- Przepisz, uruchom i przetestuj program.
- Uzupełnij program o dodatkowe instrukcje zapobiegające zamykaniu się okna konsoli po zakończeniu programu.
- Skompiluj projekt korzystając z predefiniowanych profili **Debug** i **Release** (dostępne na pasku narzędziowym), sprawdź wielkość powstałych programów (pliki **`*.exe`** znajdujące się w katalogu projektu, w podkatalogach **Debug** i **Realase**). Czym różnią się profile **Debug** i **Release**?
- Wykonaj 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]).

2. Przykład

Poniższy program oblicza pole trójkąta o bokach a, b, c z zależności:

$$P=\sqrt{s(s-a)(s-b)(s-c)},$$

gdzie $s=1/2(a+b+c)$. Dodatkowo program sprawdza czy istnieje trójkąt o podanych bokach i wyświetla stosowny komunikat.

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int main()
{
  float a, b, c, s, p;

  printf_s("Bok a = ");
  scanf_s("%f", &a);

  printf_s("Bok b = ");
  scanf_s("%f", &b);

  printf_s("Bok c = ");
  scanf_s("%f", &c);

  if(a >= b + c)
  {
    printf_s("Nie ma takiego trojkata\n");
    return EXIT_SUCCESS;
  }

  if(b >= a + c)
  {
    printf_s("Nie ma takiego trojkata\n");
    return EXIT_SUCCESS;
  }

  if(c >= a + b)
  {
    printf_s("Nie ma takiego trojkata\n");
    return EXIT_SUCCESS;
  }

  s = (a + b + c) / 2;
  p = sqrt(s * (s - a) * (s - b) * (s - c));

  printf_s("Pole trojkata = %f\n", p);

  return EXIT_SUCCESS;
}
```

- Przepisz, uruchom i przetestuj program.
- Zmodyfikuj program zastępując trzy instrukcję **`if()`** jedną z rozbudowanym wyrażeniem logicznym.

3. Zadanie

Wykorzystując zagnieżdżoną instrukcję warunkową napisz program informujący, jaka jest relacja między dwiema liczbami całkowitymi x i y wprowadzanymi do programu z klawiatury. Możliwe relacje to: $x=y$, $x>y$ i $x<y$.

4. Zadanie

Napisz program, który wykorzystując operator **%** (reszta z dzielenia) sprawdzający podzielność liczby **a** przez liczbę **b**.

5. Zadanie

Napisz program obliczający wartość funkcji:

$$f(x)=\begin{cases} x+2, dla\ x\le -2 \\ (x-2)^2, dla -2<x<0 \\ 4-x^2, dla\ x\ge 0 \end{cases}$$

6. Zadanie

Napisz program, który dla danych trzech liczb wyznacza ile z nich należy do przedziału $(0,100)$.

7. Zadanie

Rozwiąż ponownie poprzednie zadanie wykorzystując funkcję. Skorzystaj ze szkieletu aplikacji:

```
#include <stdlib.h>
#include <stdio.h>

int czyWPrzedziale(float, float, float); // prototyp funkcji

int main()
{
  // wpisz tresc programu

  return EXIT_SUCCESS;
}

int czyWPrzedziale(float x, float A, float B)
{
  if((x > A) && (x < B))
  {
    printf_s("Liczba %.2f zawiera sie w przedziale (%.2f, %.2f)\n", x, A, B);
    return 1;
  }
  else
  {
    printf_s("Liczba %.2f jest poza przedzialem (%.2f, %.2f)\n", x, A, B);
    return 0;
  }
}
```

Zadania dodatkowe

1. Napisz program wyznaczający obwód i pole koła dla zadanego promienia.

2. Napisz program wczytujący współrzędne trzech punktów i wyświetlający ile z nich leży w okręgu o danym promieniu. Opracuj funkcję do sprawdzenia czy punkt znajduje się wewnątrz okręgu.
3. Napisz program sprawdzający liczbę dni w roku. Wskazówka: *rok jest przestępny jeśli jest podzielny przez 4 z wyłączeniem przypadków gdy jest podzielny przez 100 i jednocześnie nie jest podzielny przez 400*.
4. Rozwiąż poprzednie zadanie wykorzystują funkcję **`int cztPrzestepny(int)`** zwracającą wartość **1** jeśli rok jest przestępny i **0** jeśli rok jest nieprzestępny.
5. Popraw funkcję z zadania 4 tak, żeby wykorzystywała operator wyrażenia warunkowego. Wskazówka: operator **`a ? b : c`** ocenia wartość logiczną wyrażenia **a**, jeśli jest ono prawdziwe, to zwracana jest wartość **b**, w przeciwnym wypadku zwracana jest wartość **c**.