

Podstawy programowania

mgr inż. Katarzyna Dadek

2024

Lista 2 IDE

Programowanie to proces tworzenia instrukcji, które komputer wykonuje, aby realizować określone zadania. W tym wprowadzeniu skoncentrujemy się na języku C#, który jest jednym z najpopularniejszych języków programowania, używanym do tworzenia aplikacji na platformę .NET. C# łączy w sobie cechy języków obiektowych oraz nowoczesne podejścia do programowania, co czyni go idealnym narzędziem zarówno dla początkujących, jak i doświadczonych programistów.

2.1 Środowisko Programistyczne

Aby rozpocząć programowanie w C#, konieczne jest skonfigurowanie odpowiedniego środowiska programistycznego. Najczęściej wykorzystywanym narzędziem jest Visual Studio lub JetBrains Rider, które oferują bogate funkcje, takie jak automatyczne uzupełnianie kodu, debugowanie i zarządzanie projektami. Wybór odpowiedniego IDE (Integrated Development Environment) pozwala na efektywną pracę i ułatwia proces tworzenia oprogramowania.

2.2 Struktura elementarnego programu

Elementarny program w C# składa się z kilku kluczowych elementów: deklaracji przestrzeni nazw, klasy oraz metody Main, która jest punktem startowym dla każdego programu. Programy w C# są organizowane w klasy, co pozwala na użycie programowania obiektowego i strukturalnego, umożliwiając tworzenie bardziej złożonych aplikacji.

2.3 Podstawowe typy

C# oferuje wiele podstawowych typów danych, które można wykorzystać do przechowywania wartości. Do najpopularniejszych z nich należą:

- *int* – typ całkowity, przechowujący liczby całkowite;
- *double* – typ zmiennoprzecinkowy, używany do reprezentacji liczb rzeczywistych;
- *char* – typ znakowy, przechowujący pojedyncze znaki;
- *string* – typ tekstowy, używany do przechowywania ciągów znaków;
- *bool* – typ logiczny, reprezentujący wartości prawda/fałsz;

2.4 Zmienne i instrukcje

Zmienne w C# to kontenery do przechowywania danych, którym przypisuje się określone typy. Każda zmienna musi być zadeklarowana przed użyciem, co zwiększa bezpieczeństwo kodu. Instrukcje, takie jak *if*, *for*, i *while*, pozwalają na wykonywanie operacji warunkowych i pętli, co czyni program bardziej dynamicznym i interaktywnym (omówione zostaną dokładniej na kolejnych laboratoriach). Dzięki tym podstawowym elementom programowania w C# można tworzyć różnorodne aplikacje i skrypty, realizujące zadania od prostych po bardzo złożone.

2.5 Linki

Poniżej podano linki zawierające dokładniejszy opis powyższych zagadnień wraz z przykładami.

- https://www.w3schools.com/cs/cs_intro.php
- https://www.w3schools.com/cs/cs_getstarted.php
- https://www.w3schools.com/cs/cs_syntax.php
- https://www.w3schools.com/cs/cs_output.php
- https://www.w3schools.com/cs/cs_comments.php
- https://www.w3schools.com/cs/cs_variables.php
- https://www.w3schools.com/cs/cs_data_types.php

Ważna informacja

Każde zadanie wymagające stworzenia kodu powinno być osobnym projektem w rozwiązaniu. Aby stworzyć nowy projekt należy prawym przyciskiem kliknąć na rozwiązaniu i wybrać *New project* (rys. 1).

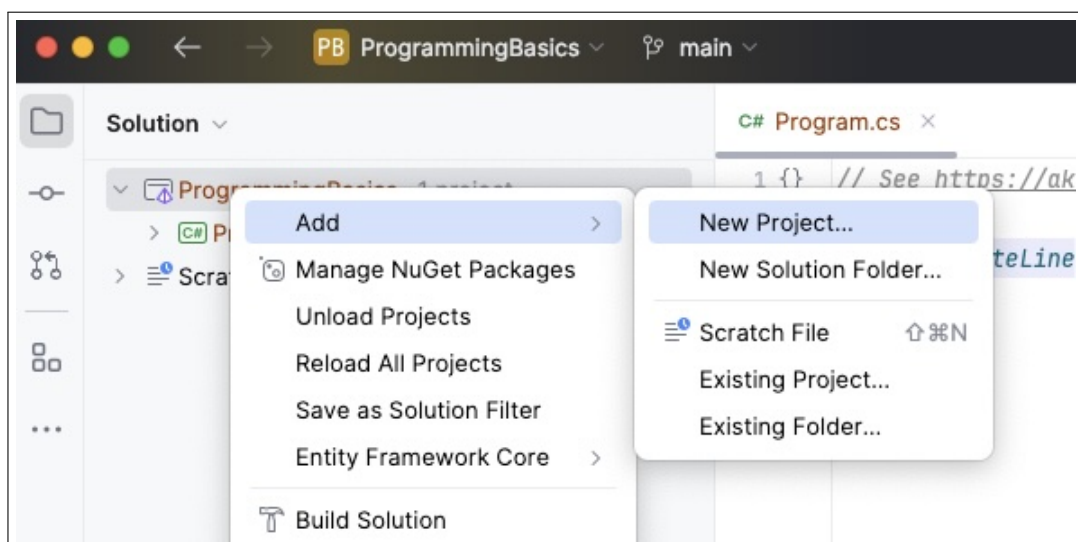
Następnie skonfiguruj projekt analogicznie jak do konfiguracji rozwiązania - ważne jest by wybrać właściwą ścieżkę dla folderu (tak jak na rys. 2 - jest to dokładnie ten sam folder w którym umieszczona jest rozwiązanie). Pamiętaj że należy stosować adekwatne nazewnictwo projektów (zadań) np. *Task__2__2*.

Po kliknięciu *Create* projekt zostanie dodany, jeżeli pojawi się monit o dostęp do folderu spoza projektu to należy wybrać na *zezwolenie dostępu dla całego folderu* oraz kliknąć *OK* (rys. 3).

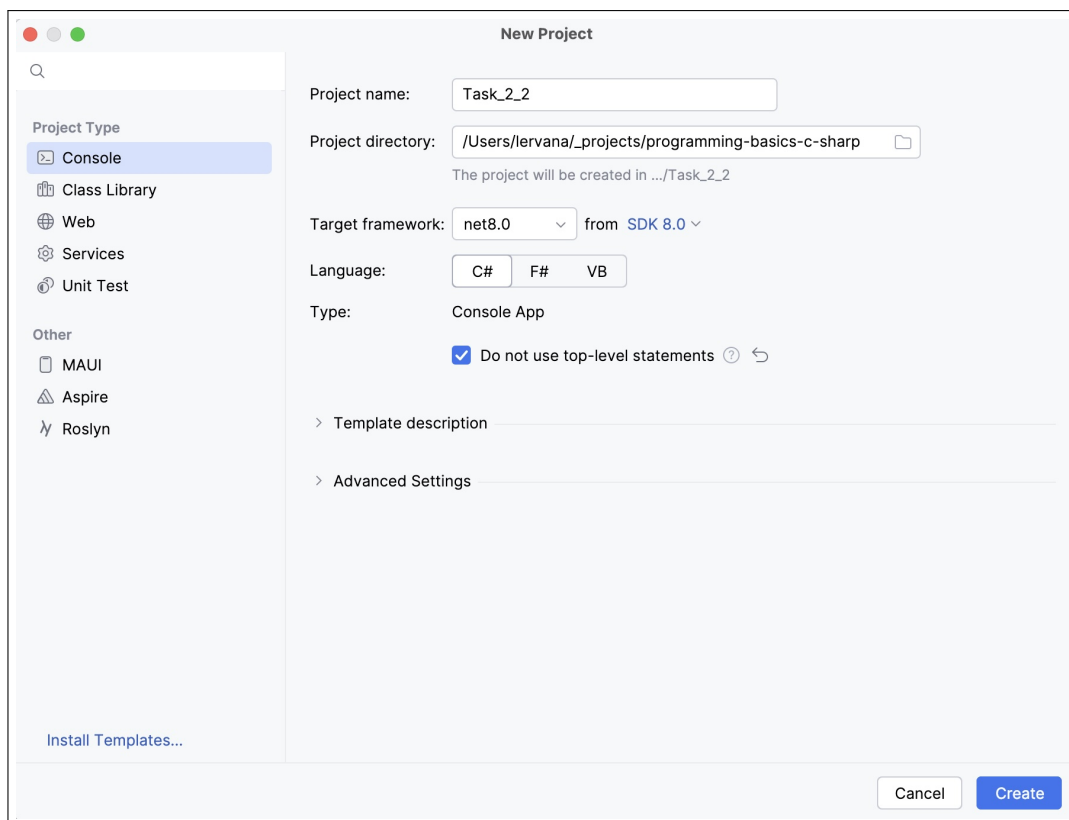
W stworzonym projekcie zmodyfikuj wyświetlany tekst tak by brzmiał *Task__2__2*. Powyższe operacje spowodują dodanie nowego projektu, obejmującego dane zadanie, do rozwiązania (rys. 4).

Teraz, aby uruchomić nowy projekt należy wybrać listę po lewej stronie od symbolu *Play* (rys. 5) i zamiast uruchomić *ProgrammingBasics* należy wybrać przycisk *Play* przy nazwie nowego projektu (np. *Task__2__2* - rys. 6).

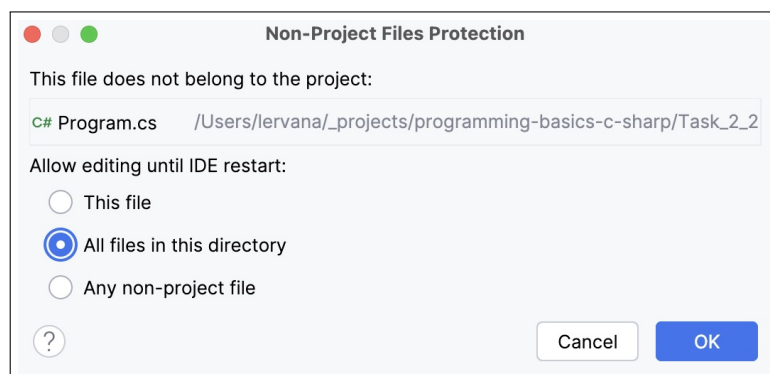
W konsoli powinien wyświetlić się napis z nowego projektu, wraz z informacją o ścieżce (rys. 7).



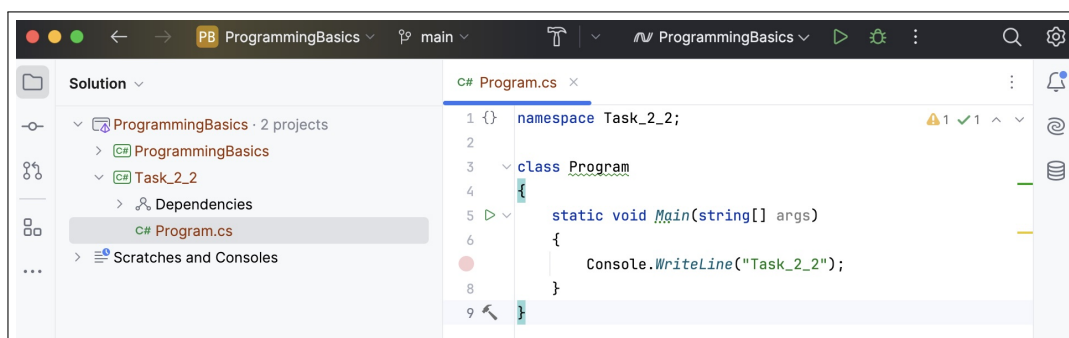
Rysunek 1. Tworzenie nowego projektu



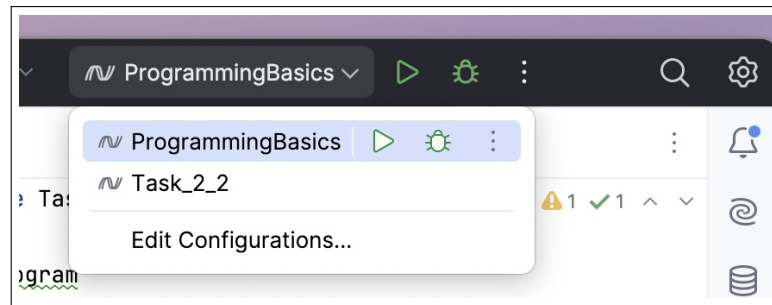
Rysunek 2. Konfiguracja nowego projektu



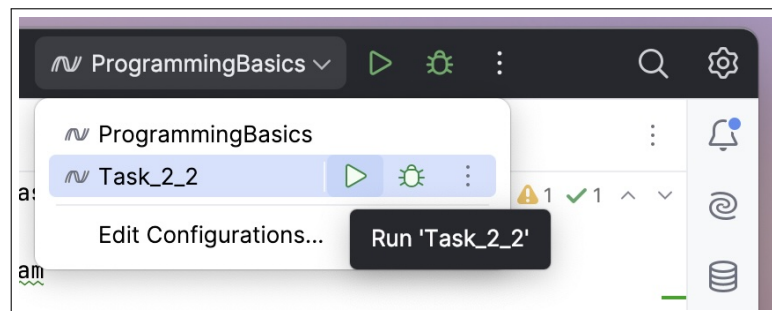
Rysunek 3. Potwierdzenie dostępu



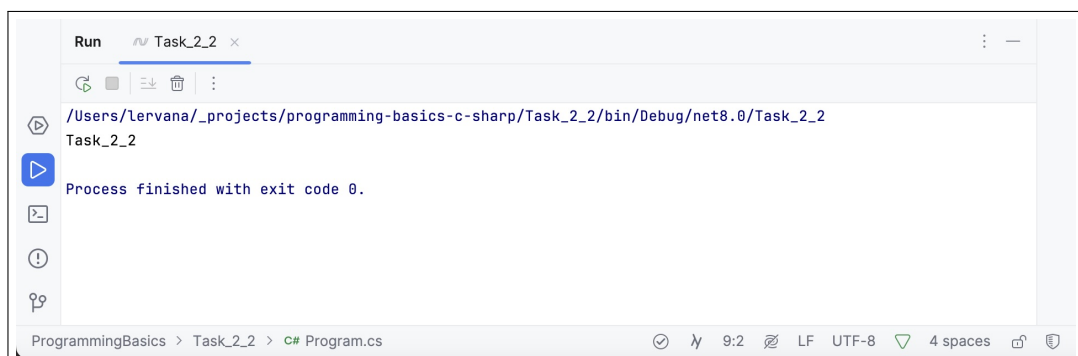
Rysunek 4. Widok solucji



Rysunek 5. Uruchomienie nowego projektu (1)



Rysunek 6. Uruchomienie nowego projektu (2)



Rysunek 7. Uruchomienie nowego projektu (3)

2.6 Zadania

Z2.1. Zapoznaj się z materiałami powyżej - wstępem teoretycznym oraz linkami.

Z2.2. Napisz program, który wyświetli napis "1234". Dodaj do programu komentarze: pojedyncze linie i komentarz wieloliniowy. W głównej funkcji Main() dodaj więcej wywołań Console.WriteLine(), aby zapoznać się ze strukturą programu i przećwiczyć sposób działania programu liniowo.

Z2.3. Utwórz zmienne o typach *int*, *double*, *string*, *bool*. Przypisz im odpowiednie wartości, a następnie wyświetl te wartości na ekranie za pomocą *Console.WriteLine()*. Jakie wartości mogą przechowywać te typy?

- Z2.4. Zadeklaruj dwie zmienne *a* i *b* typu *int* i wykonaj na nich operacje: dodawania, odejmowania, mnożenia i dzielenia. Wyświetl wyniki każdej operacji.
- Z2.5. Utwórz dwie zmienne *firstName* i *lastName* typu *string*. Połącz je w jedną zmienną *fullName* i wyświetl wynik.
- Z2.6. Napisz program, który wyliczy pole prostokąta o wymiarach przechowywanych w zmiennych *x* oraz *y*. Wynik ma być najpierw zapisany do zmiennej *z*, a następnie wyświetlony.
- Z2.7. Napisz program, który przeliczy wartość z PLN na USD oraz EUR. Zadeklaruj następujące zmienne:

- *pln* – przechowującą wartość w PLN;
- *usdRate* – przechowującą kurs dolara;
- *eurRate* – przechowującą kurs euro;

Oblicz wartości przeliczone:

- *usdValue* – wartość w dolarach, obliczoną na podstawie *pln* i *usdRate*;
- *eurValue* – wartość w euro, obliczoną na podstawie *pln* i *eurRate*;

Wyświetl wyniki w konsoli za pomocą *Console.WriteLine()*.

- Z2.8. Napisz program, który oblicza pole prostokąta. Wartości boków *a* i *b* mają być "losowe". Zmienne reprezentujące wprowadzane wartości powinny być typu *double*. Wynik należy wyświetlić z dokładnością dwóch miejsc po przecinku.
- Z2.9. Napisz program, który sprawdza, czy podana liczba całkowita jest parzysta czy nieparzysta, i wyświetla odpowiedni komunikat.
- Z2.10. Stwórz program, który sprawdza, czy podany rok jest rokiem przestępnym i informuje użytkownika o wyniku.
- Z2.11. Napisz program, który konwertuje temperaturę z Celsjusza na Fahrenheit lub odwrotnie, w zależności od ustawienia flagi typu *bool*.
- Z2.12. Zadeklaruj kilka zmiennych. Wykonaj na nich operacje dzielenia modulo. Wyniki wyświetl na ekranie.
- Z2.13. Przeanalizuj poniższy kod. Nie uruchamiaj programu, ale zastanów się, jaki będzie wyświetlony ciąg liczb. Następnie po uruchomieniu kodu sprawdź swoje przypuszczenia.

```

1  class Program
2  {
3      static void Main(string[] args)
4      {
5          int x = 1, y;
6
7          Console.WriteLine(++x);
8          Console.WriteLine(x++);
9          Console.WriteLine(x);
10
11         y = x++;
12         Console.WriteLine(y);
13
14         y = ++x;
15         Console.WriteLine(++y);
16     }
17 }

```

- Z2.14. Zmień kod z powyższego zadania tak, aby operator ++ został zastąpiony operatorem --. Następnie przeanalizuj działanie powstałego programu i sprawdź, czy otrzymany wynik jest taki sam jak na ekranie po uruchomieniu kodu.
- Z2.15. Wykonaj dzielenie zmiennych typu całkowitego. Sprawdź rezultaty w sytuacji, gdy rzeczywisty wynik jest ułamkiem.
- Z2.16. Zadeklaruj zmienną typu sbyte. Przypisz jej wartość 128. Spróbuj wykonać kompilację otrzymanego kodu. Jaki jest wynik?

```

1  class Program
2  {
3      static void Main(string[] args)
4      {
5          sbyte zmienna;
6          zmienna = 128;
7          Console.WriteLine("zmienna = " + zmienna);
8      }
9  }

```

- Z2.17. Zadeklaruj zmienną typu sbyte i przypisz jej wartość 127. Następnie wykonaj operację arytmetyczną zwiększającą wartość zmiennej, a zatem powodującą przekroczenie dopuszczalnej wartości. Wyświetl wynik na ekranie. Jaki jest wynik tych operacji (odpowiedź uzasadnij)?

```

1  class Program
2  {
3      static void Main(string[] args)
4      {
5          sbyte zmienna;
6          zmienna = 127;
7          zmienna++;
8          Console.WriteLine("zmienna = " + zmienna);
9      }
10 }

```

Z2.18. Wykorzystaj operator warunkowy do zmodyfikowania wartości dowolnej zmiennej typu całkowitego (int). Operator warunkowy ma składnię następującą: *warunek ? wartość1 : wartość2*;

Z2.19. Użyj komentarza składającego się z dwóch ukośników do opisania poniższego fragmentu kodu programu.

```

1  class Program
2  {
3      static void Main(string[] args)
4      {
5          int x = 1, y;
6          y = (x == 1)? 10 : 20;
7          Console.WriteLine("y = " + y);
8      }
9  }

```

Z2.20. Użyj komentarza blokowego w kodzie programu z powyższego zadania.

Z2.21. Napisz program, który wyznaczy wartość funkcji kwadratowej $y = ax^2 + bx + c$ w zadanym punkcie. Wartości współczynników a , b , c oraz punkt x mają być podane przez użytkownika w trakcie działania programu (wczytane z konsoli). Upewnij się, że program ma przejrzysty (w miarę możliwości) interfejs.

Z2.22. Dodaj komentarze do kodu z powyższego zadania opisujące działanie każdej z instrukcji.

Z2.23. Napisz program, który poprosi użytkownika o podanie liczby (niekoniecznie całkowitej) i wypisze jej część całkowitą.