02\_Srodowiska\_pracy\_Srodowiska\_pracy\_Porownanie\_VS\_i\_VS\_Code\_instalacja\_NET\_SDK\_konfiguracja

Kasper Dobiech

# 1. Wprowadzenie

Celem lekcji jest zapoznanie ucznia ze środowiskiem IDE (Zintegrowane środowisko programistyczne) Visual Studio.

Uczeń pozna różnice pomiędzy dostępnymi IDE oraz będzie w stanie dobrać odpowiednie środowisko do pracy.

# 2. Teoria

Definicje

IDE- IDE to skrót od angielskiego Integrated Development Environment, co po polsku oznacza Zintegrowane Środowisko Programistyczne. Jest to zestaw narzędzi (program lub kilka programów), które programiści wykorzystują do tworzenia, modyfikowania, testowania i zarządzania oprogramowaniem.

## Visual Studio

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

Visual Studio to program stworzony przez firmę Microsoft, który służy do pisania, testowania i uruchamiania oprogramowania. Można powiedzieć, że to **„warsztat” dla programisty** – coś jak zestaw narzędzi dla mechanika, tylko do tworzenia aplikacji.

**Co można robić w Visual Studio?**

* **Pisać kod** w różnych językach, np. C#, C++, Python.
* **Tworzyć aplikacje na Windows, Android, iOS**, a nawet gry (np. w Unity).
* **Projektować interfejsy graficzne** metodą „przeciągnij i upuść”.
* **Sprawdzać błędy w programie** dzięki funkcji debugowania.
* **Współpracować z innymi** – program ma wbudowaną obsługę systemów kontroli wersji, np. Git.

## Visual Studio Code

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

Visual Studio Code (często nazywany **VS Code**) to **lżejsza wersja Visual Studio**, ale również bardzo potężna. Można powiedzieć:

**Visual Studio to kombajn dla dużych projektów, a Visual Studio Code to szybki i zwinny edytor dla każdego programisty.**

| **Cecha** | **Visual Studio** | **Visual Studio Code** |
| --- | --- | --- |
| Gotowe wsparcie dla C# | ✅ Tak (domyślnie) | ❌ Trzeba instalować rozszerzenia (C# + .NET SDK) |
| Tworzenie projektów Windows Forms / WPF / aplikacji konsolowych | ✅ Kreatory projektów z GUI | ❌ Trzeba pisać ręcznie w terminalu |
| Projektowanie interfejsu graficznego (przeciągnij-upuść) | ✅ Designer okienek | ❌ Brak |
| Debugowanie (śledzenie kodu krok po kroku) | ✅ Pełny debugger | ✅ Działa, ale mniej rozbudowany |
| Automatyczne podpowiedzi kodu (IntelliSense) | ✅ Bardzo rozbudowane | ✅ Dobre, ale zależne od wtyczek |

## Rider

Rider to trzecia popularna opcja obok Visual Studio i Visual Studio Code — często wybierana przez bardziej zaawansowanych programistów C#.

**Rider** to **płatne środowisko programistyczne (IDE) stworzone przez firmę JetBrains**, znaną m.in. z IntelliJ, PyCharm i ReSharpera (pluginu do Visual Studio). Rider łączy w sobie:

* **Silnik ReSharper** (bardzo inteligentne podpowiedzi, refaktoryzacje, analiza kodu),
* **Środowisko oparte na IntelliJ** (szybkie, modularne, multiplatformowe).

# 3. Przykłady

Poza **Visual Studio**, **VS Code** i **Rider**, istnieje kilka innych popularnych środowisk programistycznych (IDE i edytorów) używanych do nauki i pracy z różnymi językami.

* **Do C# i Windows:** Visual Studio
* **Do wielojęzycznego kodu i lekkiego edytora:** VS Code
* **Do profesjonalnej pracy w C#:** Rider
* **Do Pythona:** PyCharm lub VS Code
* **Do Javy:** IntelliJ IDEA lub Eclipse
* **Do mikrokontrolerów:** Arduino IDE
* **Do Androida:** Android Studio

# 4. Podsumowanie

**Dlaczego w naszej klasie będziemy używać Visual Studio?**

* Jest **najlepsze dla początkujących uczących się C#**.
* Ma **wszystko w jednym miejscu**: kod, debugowanie, projektant GUI.
* Ułatwia naukę podstaw programowania i tworzenia aplikacji.
* Pozwala od razu skupić się na **kodzie**, a nie na konfiguracji środowiska.