

Kurs

C#

Cena szkolenia

Cena spotkania grupowego wynosi 100 zł netto za 60 min. Cena spotkania indywidualnego wynosi 120 zł netto za 60 min. Ilość godzin szkolenia zależy od Twoich postępów w nauce oraz ilości czasu, którą poświęcisz na realizację zadań i projektów zleconych w ramach szkolenia.

Opis kursu

Kurs przygotuje Cię do pracy na stanowisku C# Developer, .NET Developer lub każdym innym, gdzie wymagana jest znajomość języka C#. Po zakończeniu szkolenia będziesz w stanie tworzyć lub rozwijać nowoczesne aplikacje w języku C# oraz frameworkach, które z nim współpracują.

Naukę zaczniesz od zrozumienia podstaw języka C# i struktur programistycznych, które w nim występują. Nauczysz się zagadnień związanych z algorytmiką oraz zasad optymalizacji kodu i szybkości działania aplikacji. Wykorzystasz komunikację z bazami danych SQL, NoSQL oraz mechanizmy ORM. Zbudujesz nowoczesne aplikacje, które następnie skonteneryzujesz z wykorzystaniem platformy Docker. Programy przetestujesz z wykorzystaniem bibliotek dedykowanych do testów. To oczywiście nie wszystko. Poznasz wiele innych narzędzi wspomagających pracę z opisanymi wyżej technologiami (np. Git). Twoja wiedza będzie systematycznie poszerzana o dodatkowe materiały dydaktyczne, które znajdziesz na moich kanałach YouTube, TikTok lub otrzymasz w formie cotygodniowych newsletterów.

Opanowanie zagadnień, które pojawią się w szkoleniu, pozwoli Ci dalej rozwijać się w kierunku tworzenia aplikacji webowych, mobilnych oraz gier komputerowych.

W ramach szkolenia rozwiążesz ogromną ilość zadań teoretycznych i praktycznych, które skutecznie przygotują Cię do rozmowy kwalifikacyjnej. Twoja wiedza będzie systematycznie sprawdzana w trakcie naszych spotkań i organizowanych co pewien czas indywidualnych próbnych rozmów kwalifikacyjnych w języku polskim lub angielskim.

Po zakończeniu szkolenia Twoje portfolio powiększy się o kilka lub kilkanaście rozbudowanych projektów, które potwierdzą Twoje praktyczne umiejętności posługiwania się zagadnieniami, pojawiającymi się w kursie. Na każdym etapie szkolenia możesz liczyć na nieograniczoną pomoc mentora. Zagwarantuje to zgodność projektów z przyjętymi założeniami oraz dobrymi praktykami programistycznymi, jak również zmusi Cię do systematycznej pracy nad projektami. Dodatkowo zmierzysz się z zadaniami rozwijania lub modyfikowania istniejących już projektów oraz ich analizy pod kątem wyszukiwania błędów.

Potrzebne aplikacje i narzędzia

Do odbycia szkolenia potrzebujesz komputer lub laptop z zestawem narzędzi do budowania aplikacji pojawiających się w trakcie kursu. Spotkania możemy odbywać również na moim komputerze. Po spotkaniu zawsze otrzymasz wszystkie materiały, które pojawiły się w trakcie spotkania. Potrzebne oprogramowanie szczegółowo opiszę oraz pomogę Ci zainstalować na pierwszym darmowym spotkaniu testowym.

Informacje na temat aplikacji potrzebnych do nawiązania połączenia przekażę w odpowiednim momencie przed rozpoczęciem szkolenia. Dostaniesz wtedy szczegółowy wykaz kolejnych kroków, które należy wykonać w celu przygotowania się do rozpoczęcia kursu.

Umiejętności potrzebne przed rozpoczęciem kursu

Kurs przeznaczony jest dla osób o różnym stopniu zaawansowania. Możesz do niego przystąpić nawet jeżeli do tej pory nie miałeś nic wspólnego z programowaniem. Jeżeli już pracujesz na podobnym stanowisku, szkolenie również jest dla Ciebie. Celem kursu jest przygotowanie do pierwszej pracy programisty, ale również podnoszenie kwalifikacji osób już pracujących w branży IT. Szkolenie możesz rozpocząć od dowolnego punktu w planie szkolenia. Na początek odbędziemy szczegółową rozmowę kwalifikacyjną w języku polskim lub angielskim. Dzięki niej ocenimy Twój poziom z zakresu wiedzy przed wybranym punktem w planie szkolenia.

Plan szkolenia

1. Zagadnienia podstawowe

- Konfiguracja i uruchomienie środowiska programistycznego
- Omówienie architektury .NET oraz .NET Core
- Wykonywanie kodu w środowisku zarządzanym oraz platforma CLI
- Zmienne
- Stałe
- Typy danych i kategorie typów
- Literały
- Zarządzanie znakami – typ char
- Napisy – przegląd metod typu string
- Interpolacja napisów
- Formatowanie napisów
- Typ `System.Text.StringBuilder`
- Wyrażenia regularne
- Pobieranie i prezentacja danych
- Formatowanie dwustronne
- Dyrektywa `using` oraz `using static`
- Wartości `null` oraz `void`
- Słowo kluczowe `var`
- Typy bezpośrednie i typy referencyjne
- Konwersje pomiędzy typami danych
- Operatory
- Instrukcje sterujące oraz pętle
- Mechanizm pattern matching
- Tablice
- Metody do zarządzania tablicami
- Typ tuple i dekonstrukcja
- Zagadnienia uzupełniające

2. Metody

- Definiowanie metod
- Stos i sarta
- Sposoby przekazywania parametrów do metod
- Pomijanie parametrów wejściowych - wildcards
- Parametry out variables
- Wartości zwracane przez metody
- Zwracanie przez referencję
- Metody z ciałem w postaci wyrażenia
- Metody o zmiennej ilości parametrów
- Parametry opcjonalne
- Przeciążanie metod
- Metody lokalne
- Rekurencja
- Zagadnienia uzupełniające

3. Programowanie obiektowe

- Definicja klasy i obiektu
- Referencja do obiektu klasy
- Omówienie założeń enkapsulacji
- Modyfikatory dostępu do składników klasy
- Pola instancji
- Pola tylko do odczytu - modyfikator readonly
- Metody instancji
- Referencja this
- Deklarowanie właściwości
- Właściwości przeznaczone tylko do odczytu lub tylko do zapisu
- Konstruktory
- Inicjatory obiektów
- Finalizatory
- Typy anonimowe

- Składniki statyczne
- Konstruktory statyczne
- Klasy statyczne
- Metody rozszerzające
- Klasy zagnieżdżone
- Klasy i metody częściowe
- Kompozycja
- Klasa `System.Object` i jej metody
- Zagadnienia uzupełniające

4. Dziedziczenie

- Omówienie założeń dziedziczenia
- Klasa bazowa i klasa pochodna
- Modyfikatory dostępu w dziedziczeniu
- Konstruktory w dziedziczeniu
- Przesłanianie składowych z klas bazowych
- Polimorfizm
- Metody rozszerzające w dziedziczeniu
- Składniki statyczne w dziedziczeniu
- Rzutowanie między typem bazowym a pochodnym
- Klasy zamknięte – modyfikator `sealed`
- Modyfikatory `virtual` oraz `new` w dziedziczeniu
- Odwołanie do składników klasy bazowej – słowo kluczowe `base`
- Klasy abstrakcyjne
- Zagadnienia uzupełniające

5. Interfejsy

- Deklaracja interfejsu
- Implementacja interfejsu
- Interfejsy a polimorfizm
- Implementacja z jawnie podawanym interfejsem
- Implementacja bez jawnie podawanego interfejsu
- Dziedziczenie interfejsów
- Metody rozszerzające i interfejsy
- Interfejsy znacznikowe
- Zagadnienia uzupełniające

6. Delegaty i zdarzenia

- Omówienie zasady działania delegatów
- Deklarowanie typu delegata
- Tworzenie instancji delegata
- Metody anonimowe
- Wyrażenia lambda
- Delegaty ogólnego przeznaczenia `System.Func` oraz `System.Action`
- Zarządzanie zmiennymi zewnętrznymi wyrażenia lambda
- Delegaty typu multicast i wzorzec obserwator
- Deklarowanie zdarzeń
- Typy generyczne a delegaty
- Zagadnienia uzupełniające

7. Algorytmika

- Algorytmy badające właściwości geometryczne
- Algorytm badające właściwości matematyczne
- Konwersje pomiędzy systemami liczbowymi
- Badanie ciągów danych pod kątem wybranych właściwości
- Sortowanie ciągów danych
- Zastosowanie metody dziel i zwyciężaj

- Wybrane metody numeryczne
- Programowanie zachłanne
- Algorytmy na tekstach
- Wybrane algorytmy kryptograficzne
- Rekurencja
- Przegląd wybranych struktur danych
- Zagadnienia uzupełniające

8. Obsługa wyjątków

- Omówienie hierarchii wyjątków
- Sposoby zgłaszania wyjątków
- Sposoby przechwytywania wyjątków
- Ponowne zgłaszanie przechwyconego wyjątku
- Wyjątki w wyrażeniach
- Implementacja własnych klas do obsługi wyjątków
- Dziedziczenie klas wyjątków
- Zwalnianie zasobów z wykorzystaniem instrukcji using
- Dobre praktyki zgłaszania i obsługi wyjątków
- Zagadnienia uzupełniające

9. Wzorce projektowe

- Klasyfikacja wzorców projektowych
- Implementacja wzorców kreacyjnych
- Implementacja wzorców strukturalnych
- Implementacja wzorców czynnościowych
- Zagadnienia uzupełniające

10. Kolekcje i technologia LINQ

- Omówienie hierarchii kolekcji
- Inicjatory kolekcji
- Interfejs IEnumerable<T> oraz IQueryable<T>
- Standardowe operatory kwerend

- Wyrażenia z kwerendami
- Przegląd wszystkich metod technologii LINQ
- Wzorzec Lazy Evaluation w pracy z kwerendami
- Klauzule let, into, from
- Wzorzec projektowy iterator
- Podstawowe interfejsy w kolekcjach: `ICollection<T>`, `IList<T>`, `IDictionary<T>`
- Podstawowe klasy w kolekcjach: `List<T>`, `Dictionary<K, V>`, `SortedDictionary<K, V>`, `SortedList<T>`, `Stack<T>`, `Queue<T>`, `LinkedList<T>`
- Indeksery
- Zagadnienia uzupełniające

11. Typy generyczne

- Definicja klasy generycznej
- Generyczne interfejsy i struktury
- Reprezentacja typów generycznych w CIL
- Typy Pair oraz Tuple
- Zagnieżdżone typy generyczne
- Ograniczenia typów generycznych
- Metody generyczne
- Kowariancja i kontrawariancja w typach generycznych
- Przechwytywanie informacji o typie w runtime
- Zagadnienia uzupełniające

12. Wielowątkowość oraz asynchroniczność

- Tworzenie, uruchamianie i zarządzanie wątkami
- Blokowanie i usypianie wątków
- Pule wątków
- Wątki aktywne i działające w tle
- Priorytety wątków
- Wyjątki w wielowątkowości

- Współdzielenie zasobów
- Synchronizacja
- Klasa Monitor
- Słowo kluczowe lock
- Pola jako zmienne volatile
- Klasa System.Threading.Interlocked
- Inne typy związane z synchronizacją
- Zadania asynchroniczne
- Wzorzec obsługi asynchroniczności za pomocą zadań
- Instrukcje async oraz await
- Asynchroniczne wyrażenia lambda
- Pętle równoległe
- Klasy kolekcji przetwarzanych równoległe
- Równoległe wykonywanie kwerend LINQ
- Zegary
- Zagadnienia uzupełniające

13. Entity Framework

- Omówienie architektury frameworka Entity Framework
- Zarządzanie bazą danych – podejście code-first
- Zarządzanie bazą danych – podejście database-first
- Budowanie modelu bazy danych – mapowanie tabel
- Praca z migracjami
- Dodawanie nowej tabeli
- Modyfikowanie istniejącej tabeli
- Usuwanie istniejącej tabeli
- Tworzenie relacji pomiędzy tabelami
- Wypełnianie tabel przykładowymi danymi
- Konfiguracja tabel i wprowadzanie ograniczeń z wykorzystaniem DataAnnotations
- Konfiguracja tabel i wprowadzanie ograniczeń z wykorzystaniem FluentAPI

- Pobieranie i przetwarzanie danych z wykorzystaniem technologii LINQ
- Lazy Loading
- Problem Select N + 1
- Eager Loading
- Explicit Loading
- Wykorzystanie wzorców projektowych i typów generycznych w zarządzaniu danymi
- Narzędzia wspomagające pracę z Entity Framework
- Architektura wielowarstwowa
- Implementacja kompletnej aplikacji zarządzającej komunikacją bazodanową z wykorzystaniem Entity Framework
- Zagadnienia uzupełniające

14. Zagadnienia dodatkowe

- Dyrektywy preprocesora
- Przestrzenie nazw
- Obsługa zasobów zewnętrznych
- Operator nameof
- Porównanie działania operatorów is oraz as
- Struktury
- Wyliczenia
- Przeciążanie operatorów
- Operatory konwersji
- Implementacja bibliotek
- Refleksje
- Atrybuty

15. Zarządzanie projektami – GIT

- Omówienie architektury GIT
- Przegląd instrukcji GIT
- Rozgałęzianie i scalanie w GIT
- Aplikacje i wtyczki do pracy z GIT
- Praca ze zdalnym repozytorium
- GitHub Pages
- Zagadnienia uzupełniające