

Comarch Bootcamp

Junior C# Developer

Cele szkolenia

Kurs ma na celu przygotowanie do roli Młodsze Programisty C# oraz podjęcia pracy w charakterze programisty. Podczas kursu uczestnik buduje pierwsze programy komputerowe, korzysta w stopniu zaawansowanym z narzędzi niezbędnych w codziennej pracy programisty, wybiera zaawansowane funkcje języka C# usprawniające produkowanie programów oraz korzysta z wielu technologii pozwalających na tworzenie profesjonalnych projektów komercyjnych.

Efektom kursu będzie przygotowanie samodzielne projektu o charakterze zaawansowanym.

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Korzystać z systemu kontroli wersji git
- Tworzyć aplikacje w języku C#
- Korzystać z wbudowanych bibliotek BCL, FCL
- Stosować się do zasad programowania obiektowego
- Używać narzędzi nuget, msbuild, nswag, swagger do zarządzania projektem, kompilacji oraz generowania dokumentacji
- Tworzyć aplikacje webowe przy użyciu asp.net core
- Wykorzystywać wzorce projektowe
- Tworzyć i posługiwać się relacyjnymi bazami danych
- Używać Entity Framework do operacji bazodanowych
- Tworzyć aplikacje okienkowe
- Tworzyć testy automatyczne aplikacji
- Zarządzać swoją pracą w metodyce Agile
- Tworzyć podstawowe diagramy UML

Profil uczestników

Bootcamp dedykowany jest wszystkim, którzy chcą podjąć pracę jako programista C#.

Szczegółowy program szkolenia

Wstęp do .net i języka C#

- Historia .Net i .Net Core

- Rodzaje aplikacji .NET
- Wersje .NET
- Wersje C#
- Deasemblacja kodu

Narzędzia developerskie

- GIT
 - Instalacja
 - Klonowanie repozytorium
 - Tworzenie repozytorium
 - Tworzenie branch'a
 - Merge, rozwiązywanie konfliktów
 - Umieszczanie zmian w lokalnym repozytorium
 - Umieszczanie zmian w zdalnym repozytorium
 - Tworzenie pull requestów
 - popularne usługi github, bitbucket
- .Net Core Framework
 - Instalacja
- Visual Studio
 - Instalacja
 - Rodzaje projektów
 - Logiczna i fizyczna struktura projektu
 - Najważniejsze widoki, komendy oraz skróty klawiszowe w Visual Studio
 - Uruchomienie oraz debugowanie aplikacji
 - Aplikacja „Hello World”

C# podstawy

- Komentarze, regiony
- Typy podstawowe
- Operatory
- Metody
- Tablice
- Pętle
- Instrukcje warunkowe
- Typy danych
- Typy generyczne
- Wnioskowanie typu (var)
- Typy wyliczeniowe

Wstęp do obiektowości

- Klasy
- Struktury
- Konstruktory
- Pola
- Właściwości
- Zdarzenia
- Polimorfizm
- Delegaty
- Metody anonimowe
- Indeksery

- Interfejsy

LINQ

- Metody rozszerzające
- Wyrażenia lambda
- LINQ to Objects
- LINQ to XML

Nowoczesny C#

- Nullable typy referencyjne
- Krotki
- Rekordy
- Dopasowanie do wzorca
- Wybrane elementy programowania funkcyjnego

Architektura, wzorce, czysty kod, testy

- Wybrane wzorce "Gang of Four"
- Wzorce projektowe
 - Service Locator
 - Dependency Injection
 - Event Broker
- Programowanie aspektowe
- Testy jednostkowe

Relacyjne bazy danych

- Relacyjne bazy danych - teoria
- Podstawy SQL
- ADO.net
- Entity Framework Core
 - Źródła danych stosowane w ASP.NET Core
 - Opisywanie modelu za pomocą encji POCO
 - CRUD Tworzenie relacyjnej bazy danych z modelu, pobieranie oraz modyfikacja

Wybrane elementy platformy .net

- Zasoby, ustawienia, lokalizowanie aplikacji
- Obsługa IO (system plików, sieć)
- Serializacja (binarna, XML, JSON)
- Refleksja
- Programowanie współbieżne (wątki, pule wątków, TPL, PLINQ)
- Programowanie asynchroniczne (APM, EAP, TAP)

Zaawansowane elementy platformy .net

- Aplikacje okienkowe (WPF):
 - XAML,
 - najważniejsze kontrolki
 - data binding
 - MVVM
 - WPF VS Windows Forms
- Aplikacje rozproszone (WCF, Web RTC)
- Aplikacje webowe:

#ASP.NET Core Podstawy

- Hosting (kestrel, konfiguracja)
- Dependency Injection,

- Wzorzec Repository
- Middleware
- Omówienie wbudowanych Middleware'ów (logowanie, obsługa błędów, CORS, serwowanie plików statycznych)
- Zasady działania mechanizmów routing'u
- Obszary stosowania mechanizmów routing'u (Areas)

#ASP.NET Core Web API

- Przetwarzanie żądania HTTP
- Budowa żądania HTTP
- Architektura REST
- Filtry (opis istniejących, tworzenie własnych)

#ASP.NET Core MVC

- Architektura MVC
- Podstawowe mechanizmy służące do budowy kontrolerów w architekturze MVC
- Klasa ActionResult i jej zastosowanie w kontrolerach
- Asynchroniczne operacje kontrolera z wykorzystaniem typów Task
- Wykorzystanie klas ViewData oraz TempData, w celu usprawnienia kontrolerów
- Widok
- Sposoby definiowania widoków
- Definiowanie układu strony
- Składnia Razor
- Typowane i nietypowane widoki
- Metody pomocnicze HTML (tworzenie własnych)
- Szablony (tworzenie własnych)
- Mechanizm metadanych
- Sekcje Mechanizmy partialviews oraz viewcomponents

Podstawy front-end'u

- JavaScript/TypeScript
- HTML
- CSS
- JQuery
- AJAX

Metody wytwarzania oprogramowania

- Agile (Scrum, Kanban)
- Dokumentacja
UML/4C

Projekt końcowy:

to w pełni działająca aplikacja tworzona w metodyce agile. Wymagane jest przygotowanie dokumentacji technicznej aplikacji oraz wykorzystanie jak największej ilości technologii.

Projekt musi spełniać trzy główne założenia – **gromadzenie danych, przetwarzanie danych oraz możliwość manipulacji danymi**. Dzięki takiemu podejściu należy wykazać się umiejętnościami pisania przemyślanego kodu back-end'owego, front-end'owego oraz projektowania i implementowania prostych algorytmów.

Skończony projekt jest również dobrym **elementem składowym do portfolio przyszłego programisty**. Temat projektu można wybrać z puli przygotowanej przez trenerów lub po konsultacji i zaakceptowaniu napisać wymyśloną samodzielnie aplikację.

Podczas wykonywania zadań oraz projektu kursanci mogą liczyć na pomoc mentorów i konsultować napotkane problemy.

Liczba dni, liczba godzin szkoleniowych

26 dni

+ 4h konsultacji HR

W cenę kursu (dla uczestników Kierunek Kariera Zawodowa) wliczony jest EGZAMIN: