

Comarch SA

★★★★★ 4,5 / 5

1 003 oceny

Comarch Bootcamp – Junior C# Developer

Numer usługi 2025/11/05/7733/3129813

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 212 h

📅 10.01.2026 do 31.05.2026

12 054,00 PLN brutto

9 800,00 PLN netto

56,86 PLN brutto/h

46,23 PLN netto/h

Informacje podstawowe

Kategoria

Informatyka i telekomunikacja / Administracja IT i systemy komputerowe

Identyfikatory projektów

Małopolski Pociąg do kariery

Grupa docelowa usługi

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

Szkolenie skierowane jest do osób chcących dopiero rozpocząć przygodę z programowaniem od zera. Na szkoleniu wymagana jest biegła obsługa komputera, a predyspozycje algorytmiczno-matematyczne będą dodatkowym atutem.

Wybierz nasz kurs, jeżeli:

- Pracujesz w innym obszarze specjalności i chcesz się przekwalifikować
- Chcesz zdobyć kompetencje dostosowane do oczekiwań pracodawców względem kandydatów stanowiska C# Developer
- Masz już za sobą pierwsze kroki w programowaniu (np. inne języki) i chcesz wziąć udział w intensywnym kursie uczącym programować w C#

„Usługa również adresowana dla uczestników projektu Kierunek Kariera Zawodowa” i „Małopolskie Bony Rozwojowe Plus”

Minimalna liczba uczestników

5

Maksymalna liczba uczestników

14

Data zakończenia rekrutacji

02-01-2026

Forma prowadzenia usługi

zdalna w czasie rzeczywistym

Liczba godzin usługi

212

Cel

Cel edukacyjny

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. Efektem kursu będzie przygotowanie samodzielnie projektu o char

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Korzysta z systemu kontroli wersji Git do zarządzania kodem i współpracy z zespołem.	Uczestnik skutecznie zarządza repozytoriami Git, tworzy gałęzie, łączy je, rozwiązuje konflikty oraz wykonuje operacje na kodzie w sposób efektywny i zgodny z najlepszymi praktykami w zakresie kontroli wersji.	Test teoretyczny
Tworzy aplikacje w języku C#, wykorzystując pełny potencjał tego języka.	Uczestnik tworzy aplikacje, wykorzystując zaawansowane funkcje języka C#, takie jak delegaty, zdarzenia, LINQ, asynchroniczność, generics, co pozwala na optymalne wykorzystanie możliwości tego języka.	Test teoretyczny
Korzysta z wbudowanych bibliotek BCL i FCL, przyspieszając rozwój aplikacji.	Uczestnik wykorzystuje odpowiednie klasy i narzędzia z bibliotek BCL (Base Class Library) i FCL (Framework Class Library), przyspieszając rozwój aplikacji oraz poprawiając jakość kodu poprzez ponowne wykorzystanie sprawdzonych komponentów.	Test teoretyczny
Stosuje zasady programowania obiektowego, co pozwala na tworzenie przejrzystego i łatwego w utrzymaniu kodu.	Uczestnik tworzy przejrzysty, dobrze zorganizowany kod, przestrzegając zasad programowania obiektowego (SOLID, wzorce projektowe), co pozwala na łatwe rozbudowywanie i utrzymanie aplikacji.	Test teoretyczny
Używa narzędzi takich jak NuGet, MSBuild, NSwag i Swagger do zarządzania projektem, kompilacji oraz generowania dokumentacji.	Uczestnik efektywnie wykorzystuje narzędzia do zarządzania zależnościami (NuGet), budowania projektu (MSBuild), oraz generowania dokumentacji i specyfikacji API (NSwag, Swagger), zapewniając sprawny proces tworzenia i dokumentowania aplikacji.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Tworzy aplikacje webowe przy użyciu ASP.NET Core, implementując nowoczesne rozwiązania webowe.	Uczestnik tworzy aplikacje webowe oparte na ASP.NET Core, implementując mechanizmy routingu, middleware, autentykacji i autoryzacji oraz zarządzania sesjami, stosując najlepsze praktyki w budowie aplikacji webowych.	Test teoretyczny
Wykorzystuje wzorce projektowe w celu budowania skalowalnych i elastycznych aplikacji.	Uczestnik stosuje odpowiednie wzorce projektowe (np. MVC, Singleton, Repository, Dependency Injection), co umożliwia budowanie aplikacji o elastycznej i łatwej do rozbudowy strukturze.	Test teoretyczny
Tworzy i zarządza relacyjnymi bazami danych, zapewniając sprawną obsługę danych.	Uczestnik projektuje struktury baz danych, tworzy tabele, relacje, indeksy oraz dba o optymalizację zapytań, co zapewnia sprawną i efektywną obsługę danych w aplikacjach.	Test teoretyczny
Używa Entity Framework do efektywnego wykonywania operacji na bazach danych.	Uczestnik wykorzystuje Entity Framework do mapowania obiektowo-relacyjnego i efektywnie wykonuje operacje na bazach danych, w tym zapytania, migracje oraz operacje CRUD.	Test teoretyczny
Tworzy aplikacje okienkowe, dostosowane do różnych systemów operacyjnych.	Uczestnik tworzy aplikacje desktopowe przy użyciu odpowiednich technologii (WPF, Windows Forms), zapewniając ich kompatybilność z różnymi wersjami systemów operacyjnych i dostosowując interfejsy użytkownika do wymagań platformy.	Test teoretyczny
Tworzy testy automatyczne, zapewniając jakość i niezawodność aplikacji.	Uczestnik tworzy testy jednostkowe i integracyjne, automatyzując proces testowania oraz zapewniając wysoką jakość i niezawodność aplikacji. Testy są przeprowadzane za pomocą narzędzi takich jak NUnit, MSTest czy xUnit.	Test teoretyczny
Zarządza swoją pracą w metodyce Agile, skutecznie realizując projekty w iteracjach.	Uczestnik skutecznie realizuje projekty w metodyce Agile, aktywnie uczestnicząc w spotkaniach, planowaniu sprintów oraz retrospektywach, przyczyniając się do realizacji projektów zgodnie z harmonogramem i wymaganiami zespołu.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Tworzy podstawowe diagramy UML, które pomagają w modelowaniu i dokumentowaniu systemów informatycznych.	Uczestnik tworzy diagramy UML (np. diagramy klas, przypadków użycia, sekwencji), które skutecznie pomagają w modelowaniu, dokumentowaniu oraz wizualizowaniu struktur i procesów systemu informatycznego. Diagramy są zgodne z zasadami notacji UML i dostosowane do wymagań projektowych.	Test teoretyczny

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji zawiera opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji?

TAK

Pytanie 3. Czy dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

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

Harmonogram

Liczba przedmiotów/zajęć: 0

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
Brak wyników.					

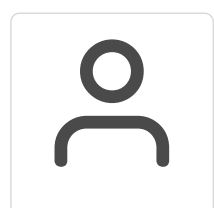
Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	12 054,00 PLN
Koszt przypadający na 1 uczestnika netto	9 800,00 PLN
Koszt osobogodziny brutto	56,86 PLN
Koszt osobogodziny netto	46,23 PLN

Prowadzący

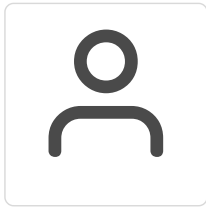
Liczba prowadzących: 2



1 z 2

Weronika Wasieczko

Specjalista z zakresu Human Resources, specjalizująca się w rekrutacji na stanowiska w branży IT; HR Business Partner w Comarch; wykształcenie wyższe; trenerka Centrum Szkoleniowego Comarch od 2018 roku, certyfikaty i szkolenia: Insightful Profiler (IP121) Professional Certificate, Extended DISC Certification Training; w ramach IT Camp prowadzi moduł HR – 4 godziny



2 z 2

Leszek Lewandowski

Mam 8-letnie doświadczenie zawodowe w programowaniu, które rozpocząłem jeszcze w liceum, tworząc proste aplikacje na potrzeby szkoły. Od tego czasu rozwijałem swoje umiejętności, pracując nad różnorodnymi projektami, które wymagają zarówno solidnej wiedzy technologicznej, jak i kreatywności w rozwiązywaniu problemów. W swojej karierze brałem udział w tworzeniu systemów rekrutacyjnych, systemów obsługi placówek medycznych, a obecnie specjalizuję się w systemach informacji pasażerskiej, dystrybucji biletów oraz obsługi dworców i terminali kolejowych.

Dodatkowo pełnię rolę trenera, prowadząc szkolenia z zakresu technologii .NET i Oracle, dzieląc się swoją wiedzą z innymi. Regularnie uczestniczę w konferencjach i spotkaniach branżowych, aby na bieżąco śledzić rozwój technologii i wymieniać doświadczenia z innymi profesjonalistami. Ostatnio brałem udział w .NET Developers Day, gdzie miałem okazję poszerzyć swoje umiejętności i nawiązać nowe kontakty zawodowe.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Uczestnicy otrzymują materiały szkoleniowe w wersji elektronicznej, nagrania oraz linki do stron www.

Warunki uczestnictwa

W ramach kursu będą wyznaczane zadania domowe oraz jeden większy projekt (ok 50 roboczo godzin) wykorzystujący umiejętności zdobyte na całym szkoleniu.

Prace domowe pozwolą na obycie się z kodem, środowiskiem i zdobycie doświadczenia w pisaniu i projektowaniu oprogramowania. Pomogą przyswoić dobre praktyki programowania i zrozumieć problemy wynikające z niestarannie pisanego kodu lub nieprzemyślanego projektu.

Część zadań będzie polegała na **samodzielnym zgłębieniu konkretnego tematu i omawianiu go na zajęciach**

Informacje dodatkowe

Zawarto umowę z WUP Kraków na rozliczanie Usług z wykorzystaniem elektronicznych bonów szkoleniowych w ramach projektu „Małopolski Pociąg do kariery”.

Warunki techniczne

Wymagania techniczne:

- Komputer / laptop ze stałym dostępem do Internetu (Szybkość pobierania/przesyłania: minimalna 2 Mb/s / 128 kb/s; zalecana 4 Mb/s / 512 kb/s)
- przeglądarka internetowa – zalecane: Google Chrome, Mozilla Firefox, Microsoft Edge
- słuchawki lub dobrej jakości głośniki
- mikrofon

Zalecane

- dodatkowy monitor
- kamera (w przypadku komputerów stacjonarnych)
- spokojne miejsce, odizolowane od zewnętrznych czynników rozpraszających

- podstawowa znajomość języka angielskiego (do sprawnego poruszania się po platformie zdalnej)

Szkolenie Zdalne prowadzone jest w czasie rzeczywistymi i transmitowane za pomocą kanału internetowego z wykorzystaniem systemu ZOOM lub Webex, który umożliwia komunikację głosową oraz wideo z Uczestnikami przebywających w dowolnym miejscu ze sprawnie działającym stałym łączem internetowym. Każdy z uczestników szkolenia otrzymuje przed szkoleniem link dostarczony w wiadomości mailowej z informacjami dotyczącymi szkolenia zdalnego.

Link umożliwiający uczestnictwo w szkoleniu jest aktywny w godzinach wskazanych na karcie usługi.

Kontakt



Aneta Lewkowska

E-mail aneta.lewkowska@comarch.pl

Telefon (+48) 126 877 811