

Claude Code od zera – pierwszy projekt z agentem AI w terminalu - szkolenie

Numer usługi 2026/09/28/199788/3841534

1 845,00 PLN brutto
1 500,00 PLN netto
115,31 PLN brutto/h
93,75 PLN netto/h
157,50 PLN cena rynkowa ⓘ

EXPERT-SALES

SPÓŁKA Z

OGRANICZONĄ

ODPOWIEDZIALNOŚĆ

CIĄ

★★★★★ 4,9 / 5

25 ocen

🏠 Usługa szkoleniowa

📄 zdalna w czasie rzeczywistym

👥 Zajęcia grupowe

🕒 16:00 h

📅 28.11.2026 do 29.11.2026

Informacje podstawowe

Kategoria

Informatyka i telekomunikacja / Programowanie

Grupa docelowa usługi

Usługa szkoleniowa skierowana jest do osób zajmujących się wytwarzaniem oprogramowania, które chcą wdrożyć narzędzia generatywnej sztucznej inteligencji (asystenty w edytorze kodu, modele konwersacyjne oraz autonomiczne agenty programistyczne) do swojej codziennej pracy projektowej. Usługa jest dedykowana w szczególności:

- Programistom (Junior, Mid, Senior) chcącym zoptymalizować proces pisania, testowania i refaktoryzacji kodu,
- Specjalistom ds. zapewnienia jakości (QA / testerom automatyzującym) oraz inżynierom DevOps tworzącym skrypty i infrastrukturę jako kod,
- Liderom technicznym (Tech Leadom) oraz architektom oprogramowania planującym bezpieczne wdrożenie narzędzi AI w zespołach wytwórczych,
- Absolwentom kierunków informatycznych i bootcampów programistycznych wchodzącym na rynek pracy.

Minimalna liczba uczestników

3

Maksymalna liczba uczestników

25

Data zakończenia rekrutacji

27-11-2026

Forma prowadzenia usługi

zdalna w czasie rzeczywistym

Podstawa uzyskania wpisu do BUR

Certyfikat systemu zarządzania jakością wg. ISO 9001:2015 (PN-EN ISO 9001:2015) - w zakresie usług szkoleniowych

Cel

Cel edukacyjny

Usługa przygotowuje do samodzielnego realizowania projektów programistycznych z wykorzystaniem autonomicznego agenta kodującego Claude Code w pełnym cyklu twórczym, w tym do konfiguracji narzędzia, planowania architektury, implementacji modułów, generowania testów, zarządzania historią wersji w systemie Git oraz bezpiecznego wdrażania i publikacji oprogramowania.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Charakteryzuje architekturę i zasady działania agenta kodującego Claude Code	Omawia pętlę decyzyjną agenta (pomiędzy planem, użyciem narzędzi w terminalu a zgłoszeniem rezultatu).	Test teoretyczny
	Wyjaśnia rolę okna kontekstowego oraz mechanizmów zarządzania pamięcią sesji (komendy /clear, /compact).	Test teoretyczny
	Identyfikuje ryzyka techniczne związane z automatycznym modyfikowaniem plików oraz niekontrolowanym uruchamianiem poleceń systemowych.	Test teoretyczny
Konfiguruje środowisko pracy i reguły projektu dla narzędzia Claude Code	Przeprowadza instalację oraz inicjalizację agenta w lokalnym repozytorium kodu.	Test teoretyczny
	Opracowuje plik konfiguracyjny CLAUDE.md zawierający standardy architektoniczne, komendy budowania oraz reguły lintera.	Test teoretyczny
	Dobiera poziomy uprawnień systemowych przydzielanych agentowi w trakcie realizacji zadań w terminalu.	Test teoretyczny
Nadzoruje proces wytwarzania oprogramowania prowadzony przez agenta kodującego	Formułuje zadania implementacyjne z wykorzystaniem trybu planowania w małych, iteracyjnych krokach.	Test teoretyczny
	Kontroluje zmiany wprowadzane przez agenta w plikach projektu przed ich zatwierdzeniem do repozytorium Git.	Test teoretyczny
	Zleca agentowi tworzenie oraz uruchamianie testów automatycznych weryfikujących poprawność kodu.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Ocenia bezpieczeństwo i jakość rozwiązań programistycznych generowanych przez agenta AI	Weryfikuje proponowane przez agenta komendy terminala i zmiany w kodzie pod kątem bezpieczeństwa danych i stabilności systemu.	Test teoretyczny
	Rozstrzyga, które elementy logiki biznesowej powierzyć agentowi, a które zaimplementować ręcznie.	Test teoretyczny
	Uzasadnia konieczność wycofania lub przerwania błędnej pętli wykonawczej agenta na podstawie analizy diffu w systemie Git.	Test teoretyczny

Kwalifikacje i kompetencje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem zawierają opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji i zgodnie z zaplanowanymi metodami walidacji?

TAK

Pytanie 3. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

- Pierwsza sesja z agentem:** Architektura agenta kodującego vs tradycyjne czaty AI i asystenci w edytorze; instalacja CLI Claude Code, uwierzytelnianie, podstawowe komendy terminala; inspekcja plików i struktury projektu przez agenta; polityka uprawnień, zatwierdzanie modyfikacji i przerywanie procesów; ćwiczenia z budowy prostego skryptu pomocniczego.
- Planowanie projektu i konfiguracja reguł:** Definiowanie zakresu architektury aplikacji (MVP); tryb planowania zadań; strukturyzacja pliku instrukcji CLAUDE.md (stos technologiczny, lintery, polecenia uruchomieniowe); workflow w systemie kontroli wersji Git (strategia małych commitów); ćwiczenia: generowanie planu oraz szkieletu projektu.
- Przyrostowa implementacja aplikacji:** Praca w małych pętlach deweloperskich; odwoływanie się do plików kontekstowych oraz komend powłoki systemowej; cofanie błędnych iteracji (skrót sesji, cofanie zmian w Git); analiza i inspekcja jakości kodu generowanego automatycznie; ćwiczenia: implementacja bazowej logiki biznesowej.
- Testowanie, diagnostyka i stabilizacja:** Generowanie testów jednostkowych i integracyjnych przez agenta; raportowanie błędów modelowi (kroki reprodukcji, logi, oczekiwany rezultat); debugowanie i analiza stack trace; kontrola diffu zmian przed commitem;

- ćwiczenia: pokrycie kodu testami i usunięcie usterek.
5. **Zarządzanie kontekstem sesji i bezpieczeństwo:** Zarządzanie pamięcią podręczną modelu (komendy /clear, /compact, wznawianie przerwanych sesji); parametryzacja poziomu namysłu modelu (thinking budget); konfiguracja reguł bezpieczeństwa (uprawnienia CLI, ochrona kluczy API, .env, .gitignore); ćwiczenia: bezpieczna rozbudowa modułu w nowo otwartej sesji.
6. **Publikacja, zaawansowane mechanizmy i walidacja (3 h):** Generowanie dokumentacji technicznej i pliku README.md; publikacja repozytorium w serwisie GitHub; przegląd zaawansowanych możliwości (subagenci, protokół MCP, hooki); warsztat samodzielnego rozwoju projektu pod okiem trenera; omówienie rozwiązań, sesja Q&A oraz przeprowadzenie testu końcowego.

Warunki organizacyjne dla przeprowadzenia usługi:

- **Forma i liczebność grupy:** Szkolenie realizowane w kameralnej grupie liczącej od 3 do maksymalnie 25 uczestników, co gwarantuje możliwość bieżącej konsultacji kodu z trenerem.
- **Samodzielne stanowisko komputerowe:** Każdy uczestnik pracuje przy indywidualnym stanowisku komputerowym (laptop/PC)
- **Dostęp i narzędzia:** Uczestnik posiada aktywne konto z dostępem do Claude Code (subskrypcja Anthropic Claude Pro/Team lub konto Anthropic Console z doładowanym kredytem API) oraz darmowe konto w serwisie GitHub.

Sposób organizacji walidacji:

- **Forma walidacji:** Samodzielny, elektroniczny test teoretyczny (quiz jednokrotnego i wielokrotnego wyboru) przeprowadzany w ramach ostatniego modułu (Moduł 6).
- **Zakres i struktura:** 10 pytań sprawdzających wiedzę i zrozumienie procedur w zakresie: architektury agenta, składni i roli pliku CLAUDE.md, komend zarządzania kontekstem sesji, bezpieczeństwa uprawnień w CLI oraz kontroli wersji w systemie Git.
- **Kryteria oceny:** Warunkiem uzyskania pozytywnego wyniku walidacji jest udzielenie minimum 60% poprawnych odpowiedzi (6 na 10 punktów). Wynik walidacji generowany jest natychmiastowo po zatwierdzeniu formularza testowego.

Harmonogram

Liczba pozycji harmonogramu: 16

Przedmiot / temat	Typ aktywności	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
1 z 16 B1.1 Agent kodujący a czat AI, instalacja Claude Code, terminal - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	28-11-2026	08:00	09:00	01:00
2 z 16 B1.2 Pierwsze zadanie z agentem, zgody na zmiany, przerywanie pracy - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	28-11-2026	09:00	10:00	01:00

Przedmiot / temat	Typ aktywności	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
3 z 16 B2.1 Pomysł i zakres projektu, tryb planowania - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	28-11-2026	10:00	11:00	01:00
4 z 16 B2.2 CLAUDE.md, repozytorium git i commit po każdym kroku - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	28-11-2026	11:00	12:00	01:00
5 z 16 -	Przerwa	-	28-11-2026	12:00	13:00	01:00
6 z 16 B3.1 Budowa małymi krokami, wskazywanie plików i poleceń powłoki - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	28-11-2026	13:00	14:00	01:00
7 z 16 B3.2 Cofanie zmian, czytanie kodu napisanego przez agenta - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	28-11-2026	14:00	15:00	01:00
8 z 16 B4.1 Testy automatyczne pisane przez agenta - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	28-11-2026	15:00	16:00	01:00

Przedmiot / temat	Typ aktywności	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
9 z 16 B4.2 Opisywanie błędów, debugowanie z agentem, przegląd zmian - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	29-11-2026	08:00	09:00	01:00
10 z 16 B5.1 Okno kontekstu, /clear i /compact, wznawianie rozmowy, dobór modelu - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	29-11-2026	09:00	10:00	01:00
11 z 16 B5.2 Tryby zgód, uprawnienia, hasła i dane w projekcie - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	29-11-2026	10:00	11:00	01:00
12 z 16 B6.1 Publikacja na GitHubie, README, przegląd dalszych możliwości - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	29-11-2026	11:00	12:00	01:00
13 z 16 -	Przerwa	-	29-11-2026	12:00	13:00	01:00
14 z 16 B6.2 Warsztat na własnym projekcie, konsultacje indywidualne - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	29-11-2026	13:00	14:00	01:00

Przedmiot / temat	Typ aktywności	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
15 z 16 B6.3 Prezentacja projektów, omówienie, Q&A - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	29-11-2026	14:00	15:00	01:00
16 z 16 -	Walidacja	-	29-11-2026	15:00	16:00	01:00

Podsumowanie

Rodzaj godzin	Liczba godzin
Suma godzin zegarowych usługi	16:00
w tym suma godzin zajęć	13:00
w tym suma godzin walidacji	01:00
w tym suma przerw	02:00
Suma godzin dydaktycznych bez przerw	18:30

Cennik

Jeżeli korzystasz z dofinansowania i usługa stanowi usługę kształcenia zawodowego lub przekwalifikowania zawodowego wraz z usługą lub dostawą towarów ściśle związaną z usługami kształcenia zawodowego lub przekwalifikowania zawodowego to możesz mieć możliwość skorzystania ze zwolnienia z podatku VAT na podstawie art. 43 ust. 1 pkt 29 lit. c ustawy z dnia 11 marca 2004 r. o podatku od towarów i usług, jeśli usługa w całości jest finansowana ze środków publicznych lub § 3 ust. 1 pkt 14 rozporządzenia Ministra Finansów z dnia 20 grudnia 2013 r. w sprawie zwolnień od podatku od towarów i usług oraz warunków stosowania tych zwolnień w przypadku, gdy usługa jest finansowana w co najmniej 70% ze środków publicznych.

Cennik

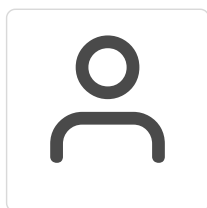
Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	1 845,00 PLN
Koszt przypadający na 1 uczestnika netto	1 500,00 PLN
Koszt osobogodziny brutto	115,31 PLN

Liczba godzin usługi

Rodzaj godzin	Liczba godzin
Liczba godzin zegarowych usługi	16:00

Prowadzący

Liczba prowadzących: 1



1 z 1

PAWEŁ HALLADIN

Paweł Halladin – programista i trener IT z ponad 20-letnim doświadczeniem zawodowym. Od 2004 roku projektuje i tworzy aplikacje desktopowe, webowe, mobilne i embedded dla klientów indywidualnych oraz firm. Od 5 lat zawodowo zajmuje się również przekazywaniem wiedzy jako korepetytor i trener IT, prowadząc zajęcia indywidualne oraz szkolenia grupowe dla osób dorosłych, w tym usługi realizowane w ramach Bazy Usług Rozwojowych.

Specjalizuje się w C#/.NET, Java/Spring, C/C++, Python, JavaScript/TypeScript, bazach danych SQL oraz wykorzystaniu AI w pracy programisty. Prowadzi również szkolenia z zakresu aplikacji webowych, embedded/IoT oraz przygotowania do egzaminów zawodowych INF.03 i INF.04.

W latach 2025–2026 zrealizował 195 godzin szkoleń grupowych, w tym 55 godzin szkolenia „Java i Spring w praktyce: projektowanie, testowanie i wdrażanie aplikacji” w ramach BUR oraz 140 godzin szkoleń z C++, .NET i programowania obiektowego. Od 2021 roku prowadzi także indywidualne zajęcia z programowania dla uczniów i studentów.

W pracy szkoleniowej stawia na praktykę – uczestnicy od początku zajęć tworzą działający kod i rozwijają projekty. Złożone zagadnienia wyjaśnia krok po kroku na konkretnych przykładach, dostosowując tempo do poziomu grupy. Posiada swobodę pracy z anglojęzyczną dokumentacją techniczną.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Materiały:

- lista pomysłów na projekt ćwiczeniowy z opisem zakresu
- przykładowy plik CLAUDE.md i ściągawka poleceń Claude Code
- seria nagrań prowadzącego o Claude Code na YouTube (30+ odcinków) jako materiał uzupełniający

Warunki techniczne

- Procesor min. 4-rdzeniowy, pamięć RAM min. 16 GB, system operacyjny macOS, Linux lub Windows z obsługą WSL2;
 - Zainstalowane środowisko Node.js (LTS), środowisko terminala (Bash/Zsh/PowerShell) oraz narzędzie kontroli wersji Git;
 - Dostęp do edytora kodu (np. VS Code) oraz aktywne uprawnienia administratora na stacji roboczej (niezbędne do globalnej instalacji pakietu npm Claude Code);
 - Stabilne łącze internetowe o przepustowości min. 30/10 Mbps.
- własny dostęp do Claude Code: aktywna subskrypcja Claude Pro lub wyższa albo klucz API z doładowanymi środkami (zapewnia uczestnik)

Kontakt



KRZYSZTOF LISZKA

E-mail liszka.k@gmail.com

Telefon (+48) 509 308 253