



Claude Code – praktyczny przewodnik programisty - szkolenie

Numer usługi 2026/09/29/199788/3844302

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

📅 05.12.2026 do 06.12.2026

Informacje podstawowe

Kategoria

Informatyka i telekomunikacja / Programowanie

Grupa docelowa usługi

Szkolenie jest skierowane do osób, które zawodowo piszą lub utrzymują kod i chcą pracować z agentem AI Claude Code w terminalu. W szczególności do:

- programistów aplikacji webowych, desktopowych, mobilnych i usług backendowych
- testerów automatyzujących i inżynierów QA piszących testy
- inżynierów DevOps utrzymujących skrypty i infrastrukturę jako kod
- liderów technicznych i architektów wprowadzających narzędzia AI w zespole
- studentów i absolwentów kierunków informatycznych, którzy już programują

Minimalna liczba uczestników

3

Maksymalna liczba uczestników

25

Data zakończenia rekrutacji

04-12-2026

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

Celem szkolenia jest nabycie przez uczestnika praktycznych umiejętności pracy z agentem kodującym Claude Code w codziennym programowaniu: prowadzenia sesji, zarządzania pamięcią projektu i oknem kontekstu, bezpiecznej konfiguracji uprawnień, automatyzacji pracy hookami i skillami oraz rozszerzania agenta o subagentów i serwery MCP.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Charakteryzuje architekturę operacyjną i mechanizmy pamięci agenta Claude Code	Omawia pętlę narzędziową agenta (tool use loop) oraz proces inspekcji i modyfikacji drzewa plików.	Test teoretyczny
	Wyjaśnia hierarchię wczytywania plików pamięci CLAUDE.md oraz zasady zarządzania oknem kontekstowym komendami /clear i /compact.	Test teoretyczny
	Rozróżnia mechanizmy rozszerzania możliwości agenta: zdarzenia cyklu życia (hooki), pliki umiejętności (skille), subagenci oraz serwery protokołu MCP.	Test teoretyczny
Konfiguruje środowisko uruchomieniowe i politykę bezpieczeństwa agenta kodującego	Wskazuje procedury instalacji narzędzia oraz uruchamiania go w trybie interaktywnym i wsadowym (-p).	Test teoretyczny
	Opracowuje plik konfiguracyjny settings.json z regułami uprawnień dla narzędzi (allow, ask, deny).	Test teoretyczny
	Definiuje reguły izolacji środowiska (sandbox) zapobiegające wykonywaniu niebezpiecznych poleceń systemowych i wyciekowi danych.	Test teoretyczny
Projektuje strukturę instrukcji projektowych i rozszerzeń funkcjonalnych agenta	Opracowuje plik CLAUDE.md definiujący wytyczne architektoniczne, styl kodowania oraz polecenia budowania projektu.	Test teoretyczny
	Projektuje skrypty hooków PreToolUse oraz PostToolUse w celu automatycznej walidacji akcji agenta.	Test teoretyczny
	Tworzy definicję modułu umiejętności w formacie SKILL.md oraz schemat integracji z zewnętrznym serwerem MCP.	Test teoretyczny

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Nadzoruje wieloetapowy proces wytwarzania oprogramowania prowadzony przez agenta	Dobiera parametry sesji, w tym model docelowy, budżet namysłu (thinking) oraz procedury wznawiania przerwanej pracy.	Test teoretyczny
	Przeprowadza procedurę testowania i diagnostyki błędów w oparciu o informacje zwrotne z terminala i logów.	Test teoretyczny
	Organizuje równoległą pracę agentów nad osobnymi zadaniami z wykorzystaniem mechanizmu git worktree.	Test teoretyczny
Weryfikuje jakość rozwiązań oraz bezpieczeństwo integracji agenta w zespole programistycznym	Dokonuje krytycznej oceny poprawności i bezpieczeństwa wygenerowanego kodu przed zatwierdzeniem zmian w repozytorium.	Test teoretyczny
	Rozstrzyga, które etapy zadania powierzyć agentowi autonomicznemu, a które wymagają bezpośredniej implementacji manualnej. Rekomenduje zespołowi ustandaryzowane pliki konfiguracyjne, reguły uprawnień i szablony umiejętności w ramach wspólnego repozytorium.	Test teoretyczny Test teoretyczny

Kwalifikacje i kompetencje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

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

TAK

Pytanie 2. Czy dokument lub wyrażnie 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 wyrażnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

Program

1. Start i sterowanie sesją

- Instalacja i pierwsze uruchomienie w repozytorium
- Tryb -p: Claude Code w potoku powłoki i w skryptach
- Wznawianie rozmowy (-c, -r, /resume), dobór modelu i jego zmiana w trakcie sesji
- Polecenia powłoki przez !, wskazywanie plików przez @
- Przerwywanie pracy agenta (Esc) i cofanie zmian w kodzie (podwójny Esc)
- Ćwiczenia: pierwsze zadania w repozytorium ćwiczeniowym

2. Pamięć projektu i okno kontekstu

- CLAUDE.md, hierarchia plików pamięci i importy @plik
- Notatki przechodzące między sesjami
- /clear i /compact – kiedy które
- Prompt caching i co go zeruje
- Poziomy namysłu (/effort) i ich koszt
- Ćwiczenia: CLAUDE.md dla własnego projektu

3. Uprawnienia i bezpieczna praca

- Tryby zgód (Shift+Tab) i plan mode
- Dlaczego --dangerously-skip-permissions nie jest skrótem do celu
- Reguły allow / ask / deny w pliku settings.json
- Dostęp do katalogów spoza repozytorium, zawężanie zestawu narzędzi
- Sandbox dla poleceń powłoki
- Ćwiczenia: konfiguracja uprawnień projektu

4. Automatyzacja: hooki, własne komendy i skille

- Hook PostToolUse: automatyczne formatowanie pliku po edycji
- Hook PreToolUse: zatrzymanie niebezpiecznego polecenia (rm -rf) przed wykonaniem
- Własna komenda i skill z jednego pliku SKILL.md
- Skille ładowane na żądanie; kto uruchamia skill: człowiek czy model
- Ćwiczenia: hook i skill dla repozytorium ćwiczeniowego

5. Agenci, MCP i wtyczki

- Subagenci ze świeżym kontekstem i fork rozmowy
- Podłączenie serwera MCP; zakresy local / project / user i plik .mcp.json w repozytorium
- Marketplace wtyczek i koszt kontekstu
- Złożenie poznanych narzędzi we własny workflow na projekcie uczestnika

6. Agent w codziennym cyklu pracy i projekt własny

- Praca od testu: agent implementuje, dopóki testy nie przejdą
- Równoległe sesje w osobnych git worktree
- Przegląd zmian przez agenta przed commitem, opis commita i pull requesta
- Warsztat na własnym projekcie: zadanie z własnej listy przeprowadzone od planu (plan mode) do zatwierdzonej zmiany
- Konsultacje indywidualne z prowadzącym w trakcie warsztatu
- Omówienie wybranych rozwiązań uczestników, Q&A

Harmonogram

Przedmiot / temat	Typ aktywności	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
1 z 16 B1.1 Start: instalacja, pierwsze uruchomienie w repozytorium, tryb -p w skryptach - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	05-12-2026	08:00	09:00	01:00
2 z 16 B1.2 Sterowanie sesją: wznawianie rozmowy, dobór modelu, polecenia ! i @, cofanie zmian - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	05-12-2026	09:00	10:00	01:00
3 z 16 B2.1 Pamięć projektu: CLAUDE.md, hierarchia plików pamięci, importy - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	05-12-2026	10:00	11:00	01:00
4 z 16 B2.2 Okno kontekstu: /clear i /compact, prompt caching, poziomy namysł - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	05-12-2026	11:00	12:00	01:00
5 z 16 -	Przerwa	-	05-12-2026	12:00	13:00	01:00

Przedmiot / temat	Typ aktywności	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
6 z 16 B3.1 Uprawnienia: tryby zgód, plan mode, reguły allow/ask/deny - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	05-12-2026	13:00	14:00	01:00
7 z 16 B3.2 Bezpieczna praca: zawężanie narzędzi, katalogi spoza repozytorium, sandbox - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	05-12-2026	14:00	15:00	01:00
8 z 16 B4.1 Hooki: PostToolUse (formatowanie po edycji), PreToolUse (blokada rm - rf) - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	05-12-2026	15:00	16:00	01:00
9 z 16 B4.2 Własne komendy i skille (SKILL.md) - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	06-12-2026	08:00	09:00	01:00
10 z 16 B5.1 Subagenci i fork rozmowy - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	06-12-2026	09:00	10:00	01:00
11 z 16 B5.2 MCP, wtyczki, własny workflow - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	06-12-2026	10:00	11:00	01:00

Przedmiot / temat	Typ aktywności	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
12 z 16 B6.1 Agent w cyklu pracy: praca od testu, równoległe sesje w git worktree, przegląd zmian i opis commita - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	06-12-2026	11:00	12:00	01:00
13 z 16 -	Przerwa	-	06-12-2026	12:00	13:00	01:00
14 z 16 B6.2 Warsztat na własnym projekcie: zadanie od planu do zatwierdzonej zmiany, konsultacje indywidualne - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	06-12-2026	13:00	14:00	01:00
15 z 16 B6.3 Warsztat na własnym projekcie: dokończenie zadania, omówienie rozwiązań uczestników, Q&A - współdzielenie ekranu	Zajęcia	PAWEŁ HALLADIN	06-12-2026	14:00	15:00	01:00
16 z 16 -	Walidacja	-	06-12-2026	15:00	16:00	01:00

Podsumowanie

Rodzaj godzin	Liczba godzin
Suma godzin zegarowych usługi	16:00
w tym suma godzin zajęć	13:00

Rodzaj godzin	Liczba godzin
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
Koszt osobogodziny netto	93,75 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:

- repozytorium ćwiczeniowe z zadaniami do każdego bloku
- przykładowe pliki konfiguracyjne: CLAUDE.md, settings.json, hooki, skill, .mcp.json
- seria nagrań prowadzącego o Claude Code na YouTube (30+ odcinków) jako materiał uzupełniający

Warunki techniczne

- komputer stacjonarny lub laptop (Windows 10/11, macOS lub Linux) z działającą kamerą i mikrofonem
- stabilne łącze internetowe o parametrach minimum 10 Mbps download / 5 Mbps upload
- uprawnienia do instalowania oprogramowania na komputerze (instalacja Claude Code)
- zainstalowany git i edytor kodu (np. Visual Studio Code)
- konto GitHub (darmowe) – dostęp do repozytorium ćwiczeniowego
- 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