Úvod do Vibe codingu pro neprogramátory i programátory

Kód kurzu: VIBECD1

Praktický kurz pro každého, kdo si chce s pomocí AI vytvořit vlastní aplikaci, a to i bez předchozích zkušeností s programováním. Účastníci nejprve pochopí základní stavební prvky programu a vyzkoušejí si jednoduchý algoritmus vytvořený pomocí AI. Poté přejdou k agentnímu vývoji, při kterém AI pracuje přímo s projektem a pomáhá aplikaci postupně vytvářet a rozšiřovat.Praktická část kurzu je vedena v Claude Code, probírané principy a pracovní postupy jsou však obecné a využitelné i při práci s dalšími AI coding agenty.Důraz je kladen na to, aby výsledkem nebyl jednorázový pokus, ale projekt, který lze bezpečně upravovat a dále rozvíjet: testování, Git jako bezpečná historie změn, systematický debugging, práce s limity kontextu napříč sessions, projektové instrukce pro AI agenta, strukturovaná dokumentace a plánování větších změn před implementací.Účastníci si odnesou nejen funkční aplikaci, ale především praktické návyky a postupy, které mohou využít i při dalších a podstatně větších projektech a při práci s různými AI coding agenty. Pro praktická cvičení musí mít každý účastník vlastní účet s přístupem ke Claude Code.

Pobočka	Dnů	Cena kurzu	ITB
Praha	2	11 800 Kč	20
Brno	2	11 800 Kč	20
Bratislava	2	500 €	20

Uvedené ceny jsou bez DPH.

Termíny kurzu

Datum	Dnů	Cena kurzu	Typ výuky	Jazyk výuky	Lokalita

Uvedené ceny jsou bez DPH.

Pro koho je kurz určen

Kurz je určen pro kohokoli, kdo chce s pomocí AI vytvářet vlastní aplikace – pro začátečníky bez zkušeností s programováním, managery, obchodníky, specialisty z provozu, marketingu, financí nebo HR, analytiky, product owners, testery i vývojáře, kteří chtějí do své práce systematicky zapojit AI coding agenty.
Praktická cvičení probíhají v Claude Code, získané principy jsou však využitelné i v dalších agentních vývojových nástrojích.

Co Vás naučíme

- porozumět základním stavebním prvkům programu
- vytvořit jednoduchý algoritmus pomocí AI
- ověřovat správnost kódu pomocí testů a unit testů
- používat Git jako bezpečnou historii projektu
- vytvořit a postupně rozšiřovat aplikaci pomocí AI coding agenta
- pokračovat v práci na existujícím projektu napříč sessions
- systematicky hledat a opravovat chyby
- pracovat s limity kontextu AI agenta
- řídit práci AI pomocí projektových instrukcí a strukturované dokumentace
- plánovat větší změny před jejich implementací
- připravit projekt tak, aby na něj mohl navázat nový AI agent, nová session nebo jiný vývojář

Požadované vstupní znalosti

- Základní práce s počítačem
- Zkušenost s programováním ani s AI nástroji není nutná
- Vlastní účet s přístupem ke Claude Code

GOPAS Praha
Na Strži 2097/63
140 00 Praha 4 - Krč
Tel.: +420 226 201 390
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 530 513 590
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 902 903 132
info@gopas.sk

Copyright © 2026 GOPAS, a.s.,
All rights reserved

- (placené předplatné Claude Pro, Max, Team nebo Enterprise, případně účet v Claude Console s API kreditem)

Osnova kurzu

Základní stavební prvky programu

Jednoduchý algoritmus vytvořený pomocí AI

- Zadání úlohy AI chatu
- Přenesení kódu do programu a spuštění
- Úprava výsledku

Testování a unit testy

- Proč testovat kód vytvořený AI
- Ruční test a unit test
- Spouštění testů

Git jako bezpečná historie projektu

- Diff a commit
- Historie změn
- Návrat k předchozí verzi

První jednoduchá aplikace pomocí AI coding agenta

- Spuštění Claude Code v projektu
- Zadání a vytvoření aplikace
- Kontrola a schválení změn
- Obecný princip práce agenta přímo se soubory projektu

Práce napříč sessions a pokračování v existujícím projektu

- Named sessions v Claude Code
- Návrat k rozpracované práci
- Co musí agent při pokračování v projektu znovu zjistit

Debugging

- Reprodukce chyby
- Hledání příčiny
- Oprava
- Ověření výsledku

Aplikace s více soubory a persistentními daty

- Rozdělení aplikace do více souborů
- Ukládání a načítání dat

Postupné rozšiřování aplikace pomocí AI coding agenta

- Malé změny místo rozsáhlých zásahů
- Průběžné ověřování
- Commit po každé funkční změně

Limity kontextu a práce napříč sessions

- Co je kontext AI agenta a proč je omezený
- Co se může při dlouhé práci nebo nové session ztratit
- Jak zajistit, aby agent znal důležité informace o projektu

Projektové instrukce pro AI agenta

- Proč používat trvalé projektové instrukce
- CLAUDE.md

GOPAS Praha
Na Strži 2097/63
140 00 Praha 4 - Krč
Tel.: +420 226 201 390
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 530 513 590
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 902 903 132
info@gopas.sk

Copyright © 2026 GOPAS, a.s.,
All rights reserved

- jako jejich praktická implementace v Claude Code
- Vytvoření a údržba projektových instrukcí
- Přenos stejného principu na jiné AI coding agenty

Pravidla projektu a způsob práce s projektem

- Architektonická pravidla
- Konvence projektu
- Příkazy pro spuštění aplikace
- Příkazy a pravidla pro testování

Strukturovaná projektová dokumentace

- Rozdělení informací do
- requirements.md
- ,
- architecture.md
- a
- testing.md
- Oddělení požadavků, technického návrhu a pravidel testování
- Odkazování na dokumentaci z projektových instrukcí

Planning před implementací větších změn

- Návrh plánu před změnou kódu
- Kontrola navrženého postupu
- Implementace podle odsouhlaseného plánu

Předání a pokračování projektu

- README.md
- a způsob spuštění projektu
- Známá omezení
- Předání projektu nové session, jinému AI agentovi nebo jinému vývojáři

GOPAS Praha
Na Strži 2097/63
140 00 Praha 4 - Krč
Tel.: +420 226 201 390
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 530 513 590
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 902 903 132
info@gopas.sk

Copyright © 2026 GOPAS, a.s.,
All rights reserved