# [KURS C#] Po co programujemy?



**UWAGA!**

**Nowa wersja kursu dostępna** jest na moim portalu z kursami. Nadal jest on **BEZPŁATNY**! A jednocześnie posiada więcej lekcji, wygodną nawigację i możliwość zapamiętania, w którym miejscu skończyłeś ostatnio!. Znajdziesz go [**POD TYM LINKIEM**](https://kurs-szarpania.pl/nauka/product/szkola-szarpania-intro)!

Kurs programowania zacznijmy od zastanowienia się i odpowiedzenia na pewne ważne pytanie.

Pytanie od którego powinien każdy zacząć i prawie nikt nie zaczyna: po co właściwie stukamy dzień w dzień w te klawisze?

Dlaczego firmy są gotowe płacić naprawdę duże pieniądze za pracę programistów? Czemu ten zawód tak dynamicznie się rozwija?

Nikt nie płaciłby za pracę, która nie jest potrzebna. Firma czy klient nie są organizacjami charytatywnymi, które chcą dawać programistom możliwość stukania palcami w firmową klawiaturę, żeby mogli na koniec dnia wrócić do domu zadowoleni po dobrej zabawie.

Zrozumienie dlaczego akurat Twoja praca jest potrzebna i za co tak naprawdę jesteś wynagradzany to klucz do zrozumienia istoty Twojej (przyszłej) pracy. Dzięki temu będziesz mógł skupić się na rozwijaniu tych elementów, które przyniosą Ci faktyczny profit i pozwolą w optymalny sposób osiągnąć cel.

# Nie uczą nas tego

W szkole, a potem na studiach często uczymy się dla uczenia się. Jeżeli sami nie drążymy mocno tematu ciągle zadając pytanie „po co?” to nikt nam nie mówi po co właściwie tej wiedzy potrzebujemy.

Ile to osób zastanawia się do czego im się przydadzą te wszystkie wzory z matematyki? Albo znajomość budowy ameby, o której słucha się na biologii? Doświadczenia na fizyce niby są ciekawe, ale czemu właściwie służą? Temu żebyśmy mieli co potem liczyć na sprawozdaniach z tych zadań?

A czy nauka nie byłaby dużo przyjemniejsza gdybyśmy od razu widzieli jej sens? Zadanie sobie pytania „dlaczego?” będzie służyło za drogowskaz przy poszerzaniu wiedzy i umiejętności.

Whether you are an entrepreneur, an employee, a leader of a team, or are looking to find clarity on your next move, your WHY is the one constant that will guide you toward fulfillment in your work and life.

źródło:[Start with Why](https://startwithwhy.com/find-your-why)

# Bo tak

Podobnego pytania „po co?” nie zadaje sobie bardzo duża grupa początkujących programistów. A i ci bardziej doświadczeni nie zawsze wiedzą jaka jest istotna programowania.  
Na pewno nie jest nią sam fakt pisania kodu. Nawet najwytrwalszy programista w końcu by miał dosyć gdyby jedynym efektem jego tygodni pracy była po prostu większa ilość kodu. Ot po prostu kilka kB więcej zapisanych na dysku.

Można by to porównać do przerzucania ziemi łopatą. Jak długo byś wytrzymał gdybyś przerzucał ziemię z jednego kupy na drugą nie wiedząc po co to robisz? Nie wydaje mi się żebyś potrafił tak pracować chociażby kilka miesięcy.

A właśnie w ten sposób najczęściej wyglądają kursy uczące podstaw jakiegoś języka programowania. Suche fakty i konstrukcje, które pisze się tylko po to żeby były napisane. Encyklopedyczne definicje dopełnione słowami „tak jest i koniec”. Bez wspomnienia o tym do czego akurat ten fragment wiedzy możemy użyć. Po co go poznajemy.

Również porady dotyczące rozwoju jako programista nieraz sprowadzają się do stwierdzenia „pisz więcej kodu„. Poza pojedynczymi wyjątkami nikt nie chce powiedzieć jak to potem wykorzystać. Co właściwie mamy pisać wykorzystując nabytą wiedzę. Co nam daje zastosowanie takiej, a nie innej metody czy struktury? A jest to moim zdaniem klucz do czerpania przyjemności z programowania i chęci ciągłego rozwoju – wiedza jak to wszystko wykorzystać w praktyce.

# Ulepszamy świat

Odpowiedź na zadane w tytule pytanie jest bardzo prosta – programujemy po to żeby ułatwiać życie użytkownikom. Ludziom. Tak zwyczajnie.

Jesteśmy po to żeby dostarczać wartość. Czy to będzie oszczędzony czas, czy możliwość zredukowania kosztów jakichś operacji, albo możliwość zmaksymalizowania przychodów, a może dostarczenie rozrywki. To wszystko są konkretne cele.

Programy zawsze pisze się po coś. Każdy z nich ma jakieś przeznaczenie. I w ten właśnie sposób powinniśmy patrzeć na to co robimy. Kiedy dowiemy się co piszemy to będziemy mogli łatwiej odpowiedzieć na pytanie jak to najlepiej zrobić. Zaczniemy patrzeć na projekty z innej perspektywy.

Jeżeli jesteś lub byłeś na studiach i miałeś tam programowanie to zastanów się, które projekty były najfajniejsze do wykonania? Te, których celem było napisanie programu, który „miał zawierać klasę, metodę i koniecznie mieć użyte dziedziczenie pomiędzy przynajmniej trzema klasami„? A może takie gdzie rozwiązywałeś problem typu „grupa naukowców musi szybko policzyć jaki wpływ na miasto będzie miała nadchodząca powódź. Pomóż im pisząc program, który…„?

Wysoka jakość kodu też nie jest tylko po to żeby była. Ma swój cel – ograniczenie kosztów utrzymania aplikacji i ułatwienie przyjmowania nowych członków zespołu. Dbanie o wysoką jakość tego co robisz wpłynie pozytywnie na zadowolenie końcowego odbiorcy w przyszłości. Nawet jeżeli nie wie, że tego potrzebuje albo chce. Bo jeśli będziesz pisał kod niechlujnie to zwiększasz szansę na wprowadzenie błędu, który dotknie użytkownika, a w dodatku spowodujesz, że czas naprawienia go będzie dużo dłuższy. Jeśli Twój kod będzie tak słaby, że nikt nie będzie chciał albo wręcz mógł z nim pracować to być może uśmiercisz przez to aplikację, która na co dzień pomaga tysiącom ludzi.

# Wyznaczajmy więc cel



Może jesteś zdziwiony, że na początku kursu jeszcze nie złapałeś za żadne narzędzia i nie ruszyłeś wystukiwać kolejnych liter na klawiaturze. Ani nawet nie usłyszałeś żadnego zdania dotyczącego języka C#. Jednak jak już wspomniałem, programy pisze się w jakimś celu, mają one jakieś zastosowanie. Dlatego zanim przejdziemy do zużywania naszych klawiatur zastanówmy się po co w ogóle będziemy to robić.

Jest to tak zwany etap zbierania wymagań. Nawet jeżeli na początku będą to tylko bardzo ogólne założenia to i tak nadadzą one kierunek dalej przygodzie. Nie chcesz przecież robić czegoś nieprzydatnego, prawda?

Ustalmy więc cel naszej pracy. Będzie to drogowskaz dla dalszych działań, dzięki któremu w każdej chwili będzie można ocenić co już jest gotowe, a czego jeszcze brakuje.

Dokładnie tak samo wygląda to w pracy. Nikt nie zaczyna pisania kodu bez sprecyzowania co właściwie ma ostatecznie powstać. A przecież jesteś tutaj żeby nauczyć się być programistą, który wie co robi. Zacznijmy więc od zebrania wymagań i zastanowienia się nad tym jaki cel chcemy osiągnąć w trakcie kursu.

**[KURS C#] Pomóżmy koledze, który otwiera bank!**



Z pierwszej części kursu już wiemy, że programujemy w jakimś celu. Dlatego zastanówmy się jaki problem rozwiążemy na początku naszej przygody z programowaniem? Robienie programu, który wyświetli prosty tekst na ekranie nie jest raczej zbyt fascynujące i nie pomaga zbyt wielu osobom. Co moglibyśmy w takim razie dać światu? hmm…

Na szczęście nie musimy długo się zastanawiać, bo z pomocą przychodzi nasz kolega, który otwiera swój mały bank i chce on od nas zamówić oprogramowanie dla tego banku!

**Startujący biznes**



Tak więc mamy pierwszego klienta – kolegę Jacka, który zakłada swój pierwszy, mały bank.

Jako, że obecnie żadna instytucja tego typu nie może funkcjonować bez oprogramowania, które ułatwi, albo wręcz umożliwi wykonywanie pewnych operacji to Jacek zwrócił się do nas o pomoc.

Poprosił nas – młodych i szalenie utalentowanych programistów – abyśmy mu przygotowali taką aplikację. Nie wypada mu przecież odmówić! A jako, że jesteśmy profesjonalistami to na początek chcemy dowiedzieć się co projekt powinien obejmować.

**Niezbędne minimum**

Bank kolegi będzie na początku pozwalał na podstawowe operacje pozwalające na prowadzenie konta. Jednak Jacek chce je mieć wykonywane na komputerze. Dzięki temu będzie wiedział, że wszystko ma pod kontrolą i uniknie błędu i stosów walających się po biurze papierów.

Jacek w swoim biznesplanie założył, że klienci będą mogli założyć u niego konto rozliczeniowe i oszczędnościowe, będą mogli wypłacać i wpłacać pieniądze, a na koniec miesiąca pieniądze na ich kontach oszczędnościowych będą się pomnażały o pewien procent. Rozsądny zestaw dostępnych możliwości.

**A dokładniej?**

Wiemy już, że Jacek ma pewien plan i my możemy mu pomóc ten plan zrealizować. Jednak kolega całą wizję opisał nam w bardzo ogólny sposób. Niewystarczający aby wiedzieć co przygotować. Warto więc dopytać o szczegóły, żeby uniknąć ewentualnych nieporozumień.

Nieporozumienia w projektach programistycznych to duży problem, który niezauważony zawczasu doprowadza to opóźnień, nerwów i konieczności wyrzucania części pracy do kosza.  
W tym momencie nie chodzi o to, żeby poznać każdy szczegół. Ważne żeby dowiedzieć się jak to ma mniej więcej ze sobą działać.

Nieporozumienia zdarzają się zawsze, ale umiejętność ich maksymalnego ograniczenia to cenna cecha. Jeżeli w jakimś miejscu musisz się czegoś domyślać to znak, że w tym miejscu powinieneś zadać pytanie o szczegóły lub doprecyzowanie idei.

**Konto dla każdego**

Wiemy, że użytkownicy mają mieć możliwość zakładania **kont rozliczeniowych i oszczędnościowych**.

Jacek, zachęcony do ujawnienia szczegółów, od razu mówi co musi mu przekazać klient aby mógł takie **konto** otworzyć. Są to:  
**– Imię i nazwisko**  
**– Pesel**

A co z numerem konta? Okazuje się, że w banku Jacka numery kont powinny być kolejnymi liczbami naturalnymi. Jednak każde konto musi mieć **numer składający się z 10 cyfr**. Jak liczba jest krótsza to **z przodu dodane będą zera**.

Dodatkowo przepisy wymagają od niego żeby numer konta poprzedzał **dwucyfrowy identyfikator banku**. W przypadku tego banku jest to **94**.

Tak więc przykładowy numer konta będzie wyglądał w ten sposób: **940000005321**.

Świetnie. Już coś mamy. Wiemy co jest potrzebne żeby konto mogło zostać otworzone. Wiemy również w jaki sposób zostanie nadany mu indywidualny numer. Przejdźmy więc do wpłat i wypłat.

**Pieniądz tu, pieniądz tam…**



Czy wpłaty i wypłaty można robić na dowolne konto i na dowolną kwotę? Tak. Aktualnie weryfikacją tego czy pieniądze mogą zostać wpłacone bądź wypłacone zajmować się będzie obsługa banku.

Do wykonania wpłaty bądź wypłaty wymagany jest numer konta i kwota, to oczywiste.

Stwierdzenie „to oczywiste” jest najczęstszym źródłem problemów. Dla kogoś kto siedzi w „biznesie” pewne rzeczy są naturalne i nie wyobraża sobie, że ktoś mógł powiedzieć inaczej.

W przedstawionym przypadku informacja, że operacja wymaga podania numeru konta i kwoty tylko teoretycznie jest nadmiarowa. A co jeżeli wymagane byłyby dodatkowo dane osoby, która zleca tą operację? Albo zamiast numeru konta podawałoby się dane jego właściciela i robiło wpłatę „na osobę”?

**Mały procent**

Bank Jacka ma **stałe oprocentowanie kont oszczędnościowych** wynoszące 4%. Jacek chce **ręcznie uruchamiać** operację dodania tych procentów do pieniędzy klientów.  
Oszczędności zwiększają się co miesiąc po tym jak kolega wyda takie polecenie.

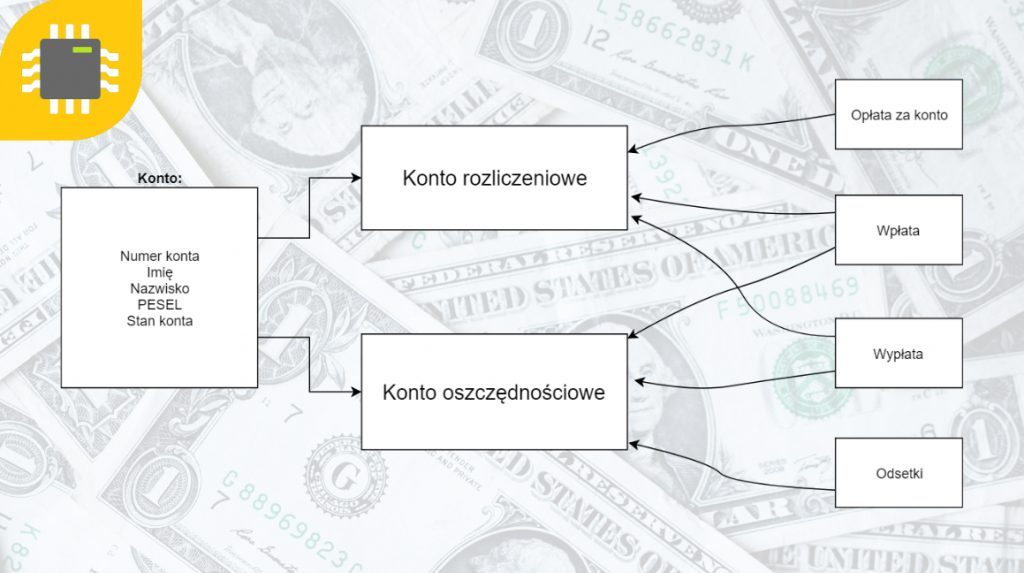
Ale nie ma tak dobrze, że klienci tylko dostają od Jacka pieniądze. Co prawda konta oszczędnościowe zyskują, ale za to konta rozliczeniowe nie są darmowe. Jacek uznał, że skoro dopiero zaczyna to nie może sobie pozwolić na takie gratisy.

Dlatego w tym samym momencie kiedy naliczany jest procent na koncie oszczędnościowym to naliczana jest również **opłata za prowadzenie konta rozliczeniowego**. Jest stała i wynosi **5zł** miesięcznie. Dzięki temu ten biznes będzie się jakoś kręcił. Tak przynajmniej koledze wyszło z obliczeń. To już nie nasza działka.

**Jest plan**

Mamy więc już plan tego co chcemy wykonać. W bardziej skondensowany sposób prezentuje to poniższa grafika, którą polecam gdzieś sobie zapisać albo po prostu mieć otwartą w zakładach przeglądarki. Będziemy do niej wracać w kolejnych lekcjach.

W końcu programujemy coś konkretnego i dobrze będzie się trzymać tego planu. Nie jest to żaden oficjalny sposób opisywania projektu, ale uważam, że w takiej formie będzie pomocny i przejrzysty.

[](https://zajacmarek.com/wp-content/uploads/2018/12/bank-diagram.jpg)

Projekt powierzony przez Jacka będzie naszym pierwszym poważnym zleceniem. Postarajmy się żeby był wykonany jak najlepiej. Dzięki temu kolega rozpowie o nas swoim znajomym.

Skoro mamy już cel jaki chcemy zrealizować i wiemy po co to wszystko robimy to zostało nam tylko jedno – nauczyć się programować w języku **C#**!