Spis treści (skrócony)

Spis treści (z prawdziwego zdarzenia)

Dobrze zaprojektowane aplikacje są super

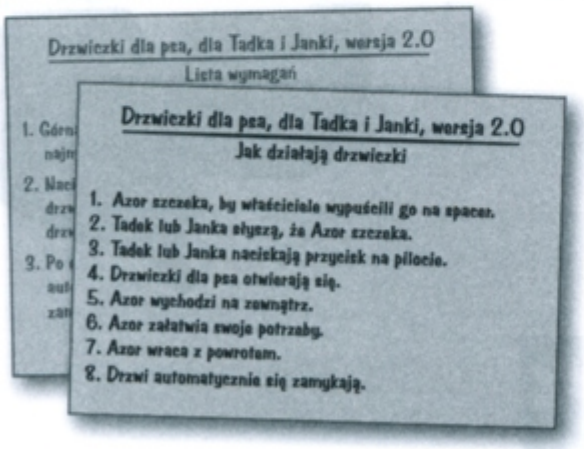
1 Tu zaczyna się wspaniałe oprogramowanie

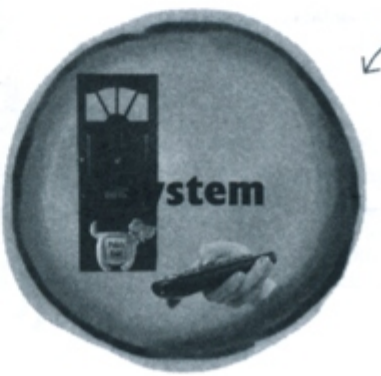
A zatem, w jaki sposób w praktyce pisze się wspaniałe oprogramowanie? Zawsze bardzo trudno jest określić, **od czego należy zacząć**. Czy aplikacja faktycznie **robi to, co powinna robić i czego od niej oczekujemy**? A co z takimi problemami jak powtarzający się kod — przecież to nie może być dobre ani właściwe rozwiązanie, prawda? Zazwyczaj trudno jest określić, które z wielu problemów należy rozwikłać w pierwszej kolejności, a jednocześnie mieć pewność, że podczas wprowadzania poprawek nie popsujemy innych fragmentów aplikacji. Bez obaw. Po zakończeniu lektury tego rozdziału będziesz już dokładnie **wiedział, jak pisać doskonałe oprogramowanie**, i pewnie podążał w kierunku trwałego poprawienia sposobu tworzenia programów. I w końcu zrozumiesz, dlaczego OOA&D to czteroliterowy skrót (pochodzący od angielskich słów: **Object-Oriented Analysis and Design**, analiza i projektowanie obiektowe), który Twoja matka **chciałaby**, byś poznał.

Gromadzenie wymagań

2 Daj im to, czego chcą

Każdy lubi zadowolonych klientów. Już wiesz, że pierwszy krok w pisaniu doskonałego oprogramowania polega na upewnieniu się, czego chce klient. Ale jak się dowiedzieć, **czego klient** oczekuje? Co więcej — skąd mieć pewność, że klient w ogóle **wie**, czego tak naprawdę chce? Właśnie wówczas na arenę wkraczają **„dobre wymagania"**. W tym rozdziale dowiesz się, w jaki sposób **zadowolić klientów**, upewniając się, że dostarczysz im właśnie to, czego chcą. Kiedy skończysz lekturę, wszystkie swoje projekty będziesz mógł opatrzyć etykietą „Satysfakcja gwarantowana" i posuniesz się o kolejny krok na drodze do tworzenia doskonałego oprogramowania... i to za każdym razem.





Wymagania ulegają zmianom

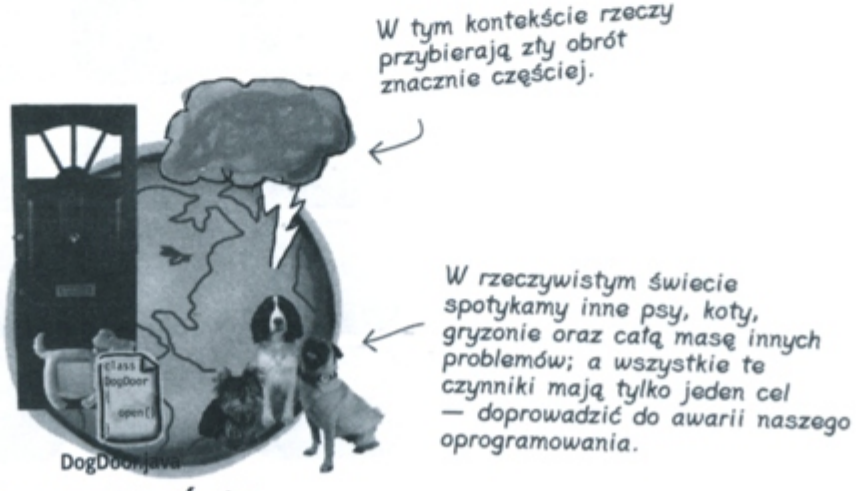
3 Kocham cię, jesteś doskonały... A teraz — zmień się

Sądzisz, że dowiedziałeś się już wszystkiego o tym, czego chciał klient? Nie tak szybko... A zatem przeprowadziłeś rozmowy z klientem, zgromadziłeś wymagania, napisałeś przypadki użycia, napisałeś i dostarczyłeś klientowi odlotową aplikację. W końcu nadszedł czas na miłego, relaksującego drinka, nieprawdaż? Pewnie... aż do momentu gdy klient uzna, że tak naprawdę chce **czegoś innego niż to, co Ci powiedział**. Bardzo podoba mu się to, co zrobiłeś — poważnie! — jednak obecnie **nie jest już w pełni usatysfakcjonowany**. W rzeczywistym świecie **wymagania zawsze się zmieniają**; to Ty musisz sobie z tymi zmianami poradzić i pomimo nich zadbać o zadowolenie klienta.

Analiza

4 Zaczynamy używać naszych aplikacji w rzeczywistym świecie

Czas zdać ostatnie egzaminy i zacząć stosować nasze aplikacje w rzeczywistym świecie. Twoje aplikacje muszą robić nieco więcej, niż jedynie działać prawidłowo na komputerze, którego używasz do ich tworzenia — komputerze o dużej mocy i doskonale skonfigurowanym; Twoje aplikacje muszą działać w takich warunkach, w jakich **rzeczywiści klienci będą ich używali**. W tym rozdziale zastanowimy się, jak zyskać pewność, że nasze aplikacje będą działać w **rzeczywistym kontekście**. Dowiesz się w nim, w jaki sposób analiza tekstowa może przekształcić stworzony wcześniej przypadek użycia w klasy i metody, które na pewno będą działać zgodnie z oczekiwaniami klienta. A kiedy skończysz lekturę tego rozdziału, także i Ty będziesz mógł powiedzieć: „Dokonałem tego! Moje oprogramowanie **jest gotowe do zastosowania w rzeczywistym świecie**!".



Rzeczywisty Świat

Dobry projekt = elastyczne oprogramowanie

5 (część 1.)

Nic nie pozostaje wiecznie takie samo

Zmiany są nieuniknione. Niezależnie od tego, jak bardzo podoba Ci się Twoje oprogramowanie w jego obecnej postaci, to najprawdopodobniej jutro zostanie ono **zmodyfikowane**. A im bardziej utrudnisz wprowadzanie modyfikacji w aplikacji, tym trudniej będzie Ci w przyszłości reagować na **zmiany potrzeb klienta**. W tym rozdziale mamy zamiar odwiedzić naszego starego znajomego oraz spróbować poprawić projekt istniejącego oprogramowania. Na tym przykładzie przekonamy się, jak **niewielkie zmiany mogą doprowadzić do poważnych problemów**. Prawdę mówiąc, jak się okaże, odkryte przez nas kłopoty będą tak poważne, że ich rozwiązanie będzie wymagało rozdziału składającego się aż z DWÓCH części!

5 (przerywnik)

Dobry projekt = elastyczne oprogramowanie

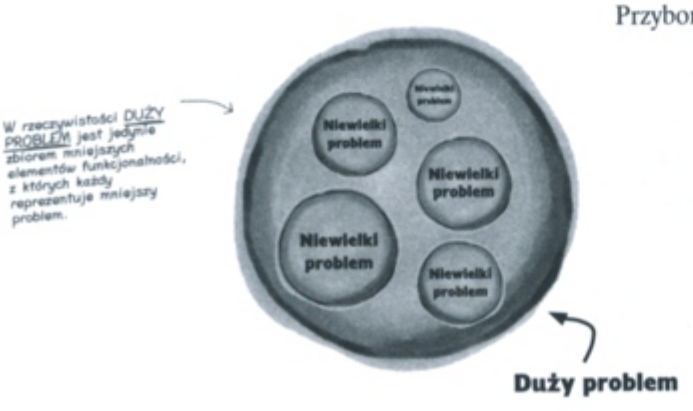
5 (część 2.) Zabierz swoje oprogramowanie na 30-minutowy trening

Czy kiedykolwiek marzyłeś o tym, by być nieco bardziej elastycznym w działaniu? Jeśli kiedykolwiek wpadłeś w kłopoty podczas prób wprowadzania zmian w aplikacji, to zazwyczaj oznacza to, że Twoje oprogramowanie powinno być nieco **bardziej elastyczne i odporne**. Aby pomóc swojej aplikacji, będziesz musiał przeprowadzić odpowiednią analizę, zastanowić się nad niezbędnymi zmianami w projekcie i dowiedzieć się, w jaki sposób **rozluźnić zależności pomiędzy jej elementami**. I w końcu, w wielkim finale, przekonasz się, że **większa spójność może pomóc w rozwiązaniu problemu powiązań**. Brzmi interesująco? A zatem przewróć kartkę — przystępujemy do poprawiania nieelastycznej aplikacji.

Rozwiązywanie naprawdę dużych problemów

6 „Nazywam się Art Vandelay... jestem Architektem"

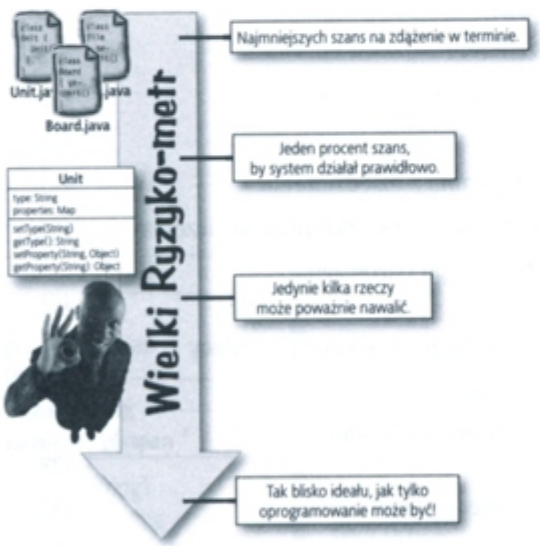
Nadszedł czas, by zbudować coś NAPRAWDĘ DUŻEGO. Czy jesteś gotów? Zdobyłeś już wiele narzędzi do swojego projektanckiego przybornika, jednak w jaki sposób z nich skorzystasz, kiedy będziesz musiał napisać coś **naprawdę dużego**? Cóż, może jeszcze nie zdajesz sobie z tego sprawy, ale **dysponujesz wszystkimi narzędziami, jakie mogą być potrzebne** do skutecznego rozwiązywania poważnych problemów. Niebawem poznasz kilka nowych narzędzi, takich jak **analiza dziedziny** oraz **diagramy przypadków użycia**, jednak nawet one bazują na wiadomościach, które już zdobyłeś, takich jak uważne słuchanie klienta oraz dokładne zrozumienie, co trzeba napisać, zanim jeszcze przystąpimy do faktycznego pisania kodu. Przygotuj się... nadszedł czas, byś sprawdził, jak sobie radzisz w roli architekta.



Architektura

7 Porządkowanie chaosu

Gdzieś musisz zacząć, jednak uważaj, żeby wybrać właściwe „gdzieś"! Już wiesz, jak podzielić swoją aplikację na wiele małych problemów, jednak oznacza to tylko i wyłącznie tyle, iż obecnie nie masz jednego dużego, lecz **WIELE** małych problemów. W tym rozdziale spróbujemy pomóc Ci w określeniu, **gdzie należy zacząć**, i upewnimy się, że nie będziesz marnował czasu na zajmowanie się nie tym, co trzeba. Nadeszła pora, by pozbierać te wszystkie **drobne kawałki** na Twoim biurku i zastanowić się, jak można je przekształcić w **uporządkowaną i dobrze zaprojektowaną aplikację**. W tym czasie poznasz niesłychanie ważne „trzy P dotyczące architektury" i dowiesz się, że Risk to znacznie więcej niż jedynie słynna gra wojenna z lat 80.



Zasady projektowania

8 Oryginalność jest przereklamowana

Powielanie jest najlepszą formą unikania głupoty. Nic chyba nie daje większej satysfakcji niż opracowanie całkowicie nowego i oryginalnego rozwiązania problemu, który męczy nas od wielu dni... aż do czasu gdy okaże się, że ktoś **rozwikłał ten sam problem** już wcześniej, a co gorsza — zrobił to znacznie lepiej niż my. W tym rozdziale przyjrzymy się kilku **zasadom projektowania**, które udało się sformułować podczas tych wszystkich lat stosowania komputerów, i dowiemy się, w jaki sposób mogą one sprawić, że staniesz się lepszym programistą. Porzuć ambitne myśli o „zrobieniu tego lepiej" — lektura tego rozdziału udowodni Ci, jak pisać programy **sprytniej i szybciej**.

Zasada otwarte-zamknięte

Zasada nie powtarzaj się

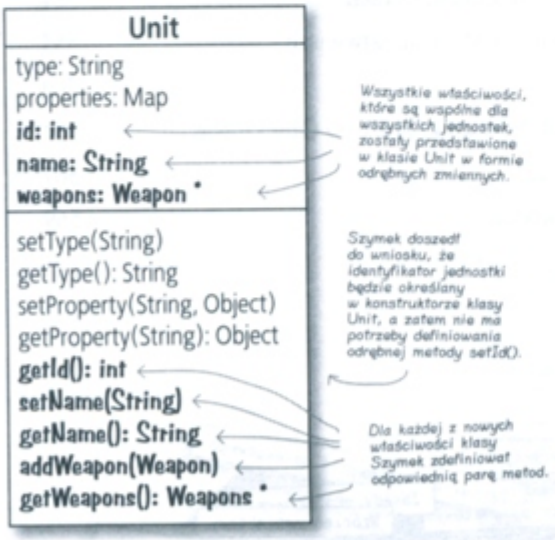
Zasada jednej odpowiedzialności

Zasada podstawienia Liskov

Powtarzanie i testowanie

9 Oprogramowanie jest wciąż przeznaczone dla klienta

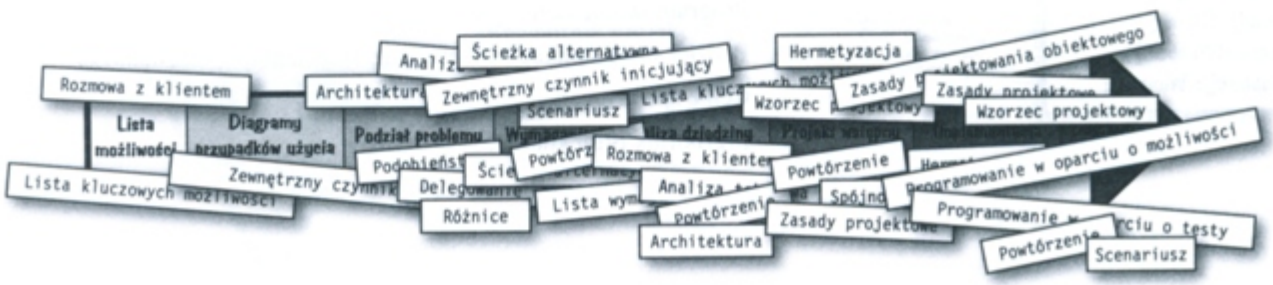
Czas pokazać klientowi, jak bardzo Ci na nim zależy. Nękają Cię szefowie? Klienci są zmartwieni? Udziałowcy wciąż zadają pytanie: „Czy wszystko będzie zrobione na czas?". Żadna ilość nawet wspaniale zaprojektowanego kodu nie zadowoli Twoich klientów; musisz **pokazać im coś działającego**. Teraz, kiedy dysponujesz już solidnym przybornikiem z narzędziami do programowania obiektowego, nadszedł czas, byś **udowodnił swoim klientom**, że pisane przez Ciebie oprogramowanie naprawdę działa. W tym rozdziale poznasz dwa sposoby pracy nad implementacją możliwości funkcjonalnych tworzonego oprogramowania — dzięki nim Twoi klienci poczują błogie ciepło, które sprawi, że powiedzą o Tobie: **„O tak, nie ma co do tego wątpliwości, jest właściwą osobą do napisania naszej aplikacji!"**.



Proces projektowania i analizy obiektowej

10 Scalając to wszystko w jedno

Czy dotarliśmy już do celu? Poświęciliśmy sporo czasu i wysiłku, by poznać wiele różnych sposobów pozwalających poprawić jakość tworzonego oprogramowania; teraz jednak nadeszła pora, by **połączyć i podsumować wszystkie zdobyte informacje**. Na to właśnie czekałeś: mamy zamiar zebrać **wszystko**, czego się nauczyłeś, i pokazać Ci, że wszystkie te informacje stanowią części jednego procesu, którego możesz wielokrotnie używać, by **tworzyć wspaniałe oprogramowanie**.



Pozostałości

A Dziesięć najważniejszych tematów (których nie poruszyliśmy)

Możesz nam wierzyć albo i nie, ale to jeszcze nie jest koniec. Owszem, wyobraź sobie, że nawet po przeczytaniu tych 600 stron wciąż możesz jeszcze znaleźć tematy, o których nawet nie wspomnieliśmy. Choć dziesięć zagadnień, jakie mamy zamiar przedstawić w tym dodatku, nie zasługuje na wiele więcej niż krótką wzmiankę, to jednak nie chcieliśmy, byś opuszczał Obiektów bez informacji na ich temat. Teraz będziesz miał nieco więcej tematów do rozmów podczas firmowej imprezy z okazji wygrania telewizyjnego quizu Obiektowa Katastrofa... poza tym któż, od czasu do czasu, nie kocha stymulujących rozmów o analizie i projektowaniu?

Kiedy już skończymy, pozostanie jeszcze... następny dodatek... no i oczywiście indeks, i może kilka reklam... ale później dotrzesz wreszcie do końca książki. Obiecujemy.

Antywzorce

Antywzorce są przeciwieństwem wzorców projektowych: stanowią one często stosowane ZŁE rozwiązania pewnych problemów. Powinniśmy być w stanie rozpoznawać te niebezpieczne pułapki i unikać ich.

Witamy w Obiektowie

Stosowanie języka obiektowego

Przygotuj się na zagraniczną wycieczkę. Czas odwiedzić Obiektów — miejsce, gdzie **obiekty robią to, co powinny**, aplikacje są **dobrze hermetyzowane** (już wkrótce dowiesz się, co to znaczy), a projekty oprogramowania pozwalają na ich **wielokrotne stosowanie i rozbudowę**. Musisz jeszcze poznać kilka dodatkowych zagadnień i poszerzyć swoje **umiejętności językowe**. Nie przejmuj się jednak, nie zajmie Ci to wiele czasu i zanim się obejrzysz, już będziesz rozmawiał w języku obiektowym, jakbyś mieszkał w Obiektowie od wielu lat.