

# Wprowadzenie

Programowanie, rozumiane jako umiejętność rozwiązywania problemów za pomocą komputera, wymaga znacznie więcej niż znajomości składni języka. Do utworzenia poprawnego kodu konieczna jest wiedza z zakresu algorytmów i struktur danych, technik projektowania, testowania i uruchamiania. Projekty informatyczne są zazwyczaj realizowane przez zespoły, a nie pojedyncze osoby, dlatego do realizacji tego typu przedsięwzięć potrzebna jest także umiejętność komunikacji w zespole. Nie można pominąć konieczności zdobycia wiedzy specyficznej dla dziedziny, dla której oprogramowanie jest tworzone, czy umiejętności zarządzania.

Książka opisuje sposoby poprawnego rozwiązywania typowych problemów, pojawiających się przy tworzeniu oprogramowania. Sposoby te, nazywane wzorcami projektowymi, rozszerzają zestaw akcesoriów dostarczanych przez język programowania, którymi możemy się posługiwać. Znajomość wzorców nie oznacza umiejętności programowania, ale dobry programista powinien mieć świadomość ich istnienia, aby unikać ponownego odkrywania znanych rozwiązań. Wiedza o przedstawionych technikach pozwala na lepszą komunikację w zespole, ponieważ można porozumiewać się na wyższym poziomie abstrakcji.

Programowanie jest umiejętnością praktyczną; aby nauczyć się programować, trzeba pisać programy. Warto także czytać kod innych, szczególnie ten dobrej jakości. Za taki uważa się kod poprawny, wydajny, przenośny i łatwy w utrzymaniu. Przykładem są biblioteki standardowe.

W tej pracy chcieliśmy połączyć podejście polegające na pokazaniu pewnych ogólnych technik oraz praktycznych przykładów. Naszym celem było napisa-

nie podręcznika do praktycznej nauki technik współcześnie wykorzystywanych w programowaniu. Wybraliśmy język C++, którym posługujemy się od kilkunastu lat. W tej pracy prezentujemy wzorce wykorzystując biblioteki, które ich używają.

Język C++ jest popularnym narzędziem służącym do tworzenia programów z wielu różnych dziedzin. Dostarcza on abstrakcji, pozwalając jednocześnie na tworzenie bardzo wydajnych rozwiązań. Wspiera wiele różnych metod tworzenia oprogramowania, m.in. podejście obiektowe i generyczne. Język ten można wygodnie rozszerzać, przykładem jest biblioteka standardowa oraz biblioteki `boost`. Biblioteki te zawierają wiele rozwiązań, które warto znać i stosować. Niektóre techniki, przedstawione w kolejnych rozdziałach, są specyficzne dla C++, inne (większość) mają swoje odpowiedniki w innych językach programowania.

Książka jest przeznaczona dla osób, które miały wcześniej styczność z programowaniem i językiem C++. Zawiera ona opis mechanizmów dostarczanych przez ten język, ale bazują one na znajomości podstawowych pojęć. Rozdział pierwszy można traktować jako przypomnienie i uzupełnienie wiadomości o C++.

Książka składa się z trzech części: przedstawienia mechanizmów języka (rozdział 1), opisu wzorców (rozdziały 2–5) oraz opisu zastosowań i bibliotek (rozdziały 6–9). Podział ten nie ścisły, czasami pokazujemy zastosowanie od razu przy opisie wprowadzanej techniki, czasami przy opisie zastosowania jest rozważana użyta tam technika.

Rozdziały 2–5 prezentują kolejne wzorce projektowe. Rozdział drugi poświęcony jest technikom wykorzystywanym w generycznym podejściu do tworzenia oprogramowania. Szablony pozwalają opisywać rozwiązania niezależne od konkretnego typu, są powszechnie stosowane w bibliotekach, dostarczając ogólnych rozwiązań. W rozdziale pokazano techniki ułatwiające korzystanie z szablonów oraz sposoby radzenia sobie z typowymi problemami tam występującymi, np. wykrywaniem błędów.

Rozdział trzeci opisuje techniki związane z pośrednim odwoływaniem się do obiektów. Przechowywanie uchwytów pozwala wykonywać pewne czynności związane z zarządzaniem obiektem: sprytnie wskaźniki usuwają zarządzany obiekt, jeżeli nie jest on wykorzystywany, techniki związane z leniwym (opóźnionym) tworzeniem lub kopiowaniem zarządzają operacją powoływania nowego obiektu. Obiekty przechowujące uchwyt mogą uprościć przeglądanie elementów w kolekcji, przechowywać odniesienie do funkcji lub metody albo dostosowywać interfejs zarządzanego obiektu do aktualnych wymagań.

Rozdział czwarty poświęcony jest sposobom tworzenia obiektów. Mechanizmy języka mają tutaj spore ograniczenia – można utworzyć tylko typ znany w momencie kompilacji, zaś identyfikator typu musi być dostarczony w formie zrozumiałej dla kompilatora. Opisane w rozdziale wzorce pozwalają tworzyć obiekty

na podstawie identyfikatora dostarczanego w czasie działania, dbają o utworzenie tylko jednej instancji obiektu danego typu, albo pozwalają dostarczać pewną grupę typów jednocześnie. Wspomniano o technikach zarządzania pamięcią oraz o technice odwoływania się do typów, które nie są znane w momencie kompilacji.

Tematem kolejnego rozdziału są wzorce związane z komunikacją pomiędzy obiektami. Omawiane są techniki rozszerzające mechanizm funkcji wirtualnych oraz techniki bardziej elastycznej niżwołanie metody, organizacji współpracy. Rozważania uzupełnia dyskusja na temat stałości.

Rozdział szósty zawiera sposoby organizowania obiektów w celu tworzenia bardziej złożonych bytów. Rozdział ten opisuje udogodnienia, których dostarcza biblioteka standardowa oraz biblioteki `boost`. Omówiono kolekcje jednowymiarowe, tablice wielowymiarowe i grafy.

Kolejny rozdział jest poświęcony mechanizmom dostarczonym przez system operacyjny, pozwalającym wykorzystywać zasoby komputera. Jednym z głównych tematów tego rozdziału jest przetwarzanie równoległe (współbieżność). Opisano biblioteki pozwalające wykorzystywać wiele strumieni instrukcji w aplikacji oraz podstawowe problemy i wzorce projektowe związane z tego rodzaju programami. Rozdział zawiera opis bibliotek obsługujących wejście i wyjście reprezentowane przez strumienie znaków lub bajtów, oraz opis techniki i bibliotek umożliwiających asynchroniczną obsługę urządzeń, co pozwala wydajnie tworzyć aplikacje – szczególnie te, komunikujące się przez sieć.

Rozdział ósmy pokazuje wykorzystanie języka C++ do przetwarzania tekstu. Omawiane są algorytmy działające na napisach, sposoby uwzględniania standardów narodowych przy operacjach na tekście, biblioteki pozwalające wykorzystywać wyrażenia regularne i gramatyki bezkontekstowe.

Ostatni rozdział zawiera opis mechanizmów i bibliotek pozwalających łączyć moduły utworzone w C++ z modułami utworzonymi w innych językach programowania.

Zagadnienia przedstawione w książce nie są omówione wyczerpująco. Opisując daną technikę, skupiamy się na jednym lub kilku wariantach, nie omawiając ich wszystkich. Musieliśmy zrezygnować z przedstawienia wielu interesujących szczegółów. Dzięki tym zabiegom udało nam się zachować początkowo założoną objętość tekstu. Bibliografia obejmuje pozycje monograficzne, które zdobyły sobie naszą sympatię i to one zawierają szczegółowe omówienie prezentowanych tematów. Tam też należy szukać odnośników do źródeł prezentowanych technik.

Prezentowany materiał powstał na bazie notatek do wykładów prowadzonych na Wydziale Elektroniki i Technik Informatycznych. Autorem pierwszej części podręcznika (rozdział 1) jest Andrzej Pająk, część drugą i trzecią (rozdziały 2–9) opracował Robert Nowak. Chcielibyśmy podziękować naszym studen-

tom oraz kolegom, którzy testowali zamieszczone tutaj programy i pracowali z wczesnymi wersjami przedstawionego dalej tekstu. Dziękujemy w szczególności Tomkowi Janczakowi, Adamowi Dzedzicowi, Szymonowi Janikowskiemu, Piotrowi Jasiunowi, Michałowi Maruszakowi, Jackowi Suchowskiemu, Sebastianowi Łuczakowi, ich uwagi przyczyniły się do usunięcia wielu błędów i niedociągnięć.