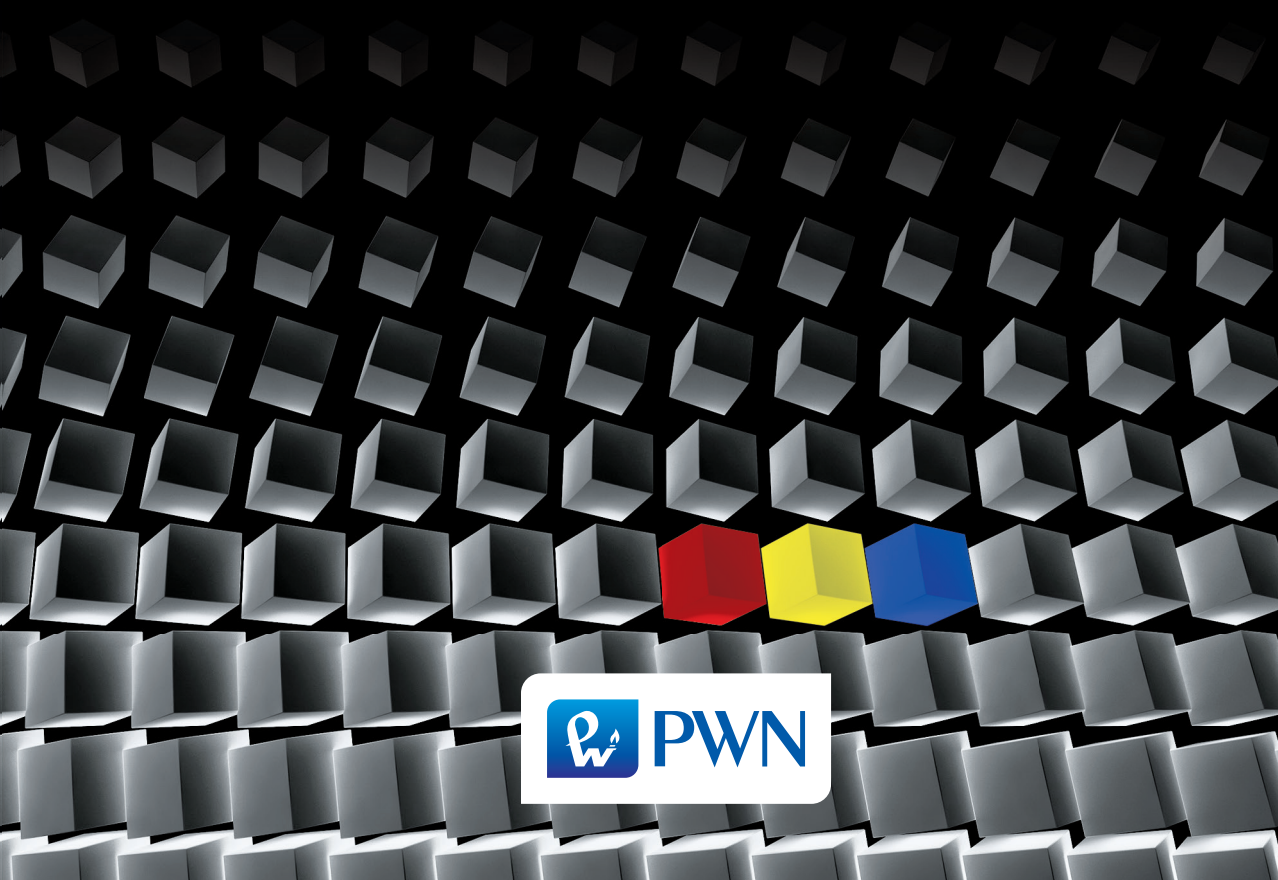
Bartosz W. Warzocha

PROGRAMOWANIE WIELOPLATFORMOWE z C++ i wxWidgets 3



PWN

Programowanie wieloplatformowe z C++ i wxWidgets 3

Bartosz W. Warzocha

Programowanie wieloplatformowe z C++ i wxWidgets 3

Projekt okładki i stron tytułowych **Hubert Zacharski**

Fotografia na okładce **sanchopancho/Adobe Stock**

Wydawca **Edyta Kawala**

Redaktor prowadzący **Jolanta Kowalczuk**

Redaktor **Irena Puchalska**

Koordynator produkcji **Anna Bączkowska**

Skład i łamanie **ALINEA**

ISBN 978-83-01-19899-2

Wydanie I
Warszawa 2018

Wydawnictwo Naukowe PWN SA
02-460 Warszawa, ul. Gottlieba Daimlera 2
tel. 22 69 54 321, faks 22 69 54 288
infolinia 801 33 33 88
e-mail: pwn@pwn.com.pl, reklama@pwn.pl
www.pwn.pl

Druk i oprawa:

Spis treści

Szczegółowy spis treści

Od autora

... dobrze jest koło sterowe mocniej ująć,
największej bowiem trzeba wytrawności
sterniczej i nie w mapę, lecz w trzewia
słoneczne patrzeć się godzi: nikt bowiem
nie przebył drogi tej tak samo dwa razy.

Stanisław Lem, *Cyberiada*

Pierwszy program komputerowy napisałem dość dawno temu. Była to bardzo prosta aplikacja graficzna wyświetlająca prostopadłościan i kilka dziwacznych figur na ekranie starego lampowego telewizora marki *Elektron*, do którego był podłączony mój pierwszy mikrokomputer – Sinclair ZX Spectrum. Do dziś pamiętam swoje zaskoczenie, gdy ów program zadziałał, szczeniacką satysfakcję i łzy ojca skupionego nad podręcznikiem BASIC'a, którego co rusz wypytywałem o sens i znaczenie różnych egzotycznie wyglądających komend. Była połowa lat osiemdziesiątych zeszłego wieku. Wszelka angielszczyzna kojarzyła się wówczas z „Kinem nocnym" i familijnymi serialami, którymi w niedzielne przedpołudnia zachwycał jeden z dwóch dostępnych wówczas kanałów telewizyjnych. Nikt nie przypuszczał, że techniki informatyczne tak się rozwiną – mimo że *Bajtek* wciąż donosił o coraz to bardziej fascynujących odkryciach. Ja też jeszcze nie wiedziałem, że to, czym wtedy żyłem, to początek wieloletniej pasji i wspaniałej przygody z programowaniem.

Prawdziwą rewolucją było dla mnie odkrywanie potęgi języka C++ oraz bibliotek, z którymi i z których mogłem wyczarować to, co chciałem. Z punktu widzenia programowania graficznego interfejsu użytkownika wspomaganego różnymi mechanizmami, jaki powinien mieć nowoczesny program komputerowy, najpotężniejszym z nich zawsze była dla mnie wxWidgets. Pracuję z nią od 2001 r. i po latach programowania stwierdzam, że nawet dzisiaj nie pomyślałbym, aby zwrócić się ku innej bibliotece służącej do wieloplatformowego programowania GUI. Teraz chcę podzielić się z Tobą moją wiedzą, doświadczeniem i pasją.

Na początek chcę Ci wyjaśnić, że to nie jest w pełni formalna, fachowa książka na temat wxWidgets. Takie publikacje już istnieją i temat w pewien sposób wyczerpują. Wszystkie one jednak traktują poszczególne elementy biblioteki w sposób niemal encyklopedyczny i niewiele można się z nich dowiedzieć o tym, jak w praktyce programować z wxWidgets. Ponadto najczęściej dotyczą wersji biblioteki, których świetność dawno już minęła i straciła swą aktualność. Czytelnik tych dzieł jest zmuszony do samodzielnej analizy, poszerzania i aktualizowania zgłębianego materiału, co często skutkuje pominięciem niektórych, bardzo istotnych zagadnień. Dlatego, jako prawdziwy fascynat wxWidgets, postanowiłem wypełnić tę lukę – stąd zrodziła się książka, którą właśnie trzymasz w rękach.

Jest ona rodzajem podręcznika, poradnika, przy lekturze którego – mam nadzieję – nabierzesz sporo nowych umiejętności i przekonasz się, jak łatwe i przyjemne jest programowanie z biblioteką wxWidgets, a także z innymi bibliotekami umożliwiającymi tworzenie profesjonalnych programów komputerowych dla systemów operacyjnych z rodziny MS Windows i Linux. Jeśli jesteś doświadczonym programistą, znajdziesz tu garść usystematyzowanej praktycznej i teoretycznej wiedzy, pogrupowanej według problematyki (choć systematyzowanie i swoista taksonomia wiadomości nie jest najważniejszym celem tej książki), do których poznania lub odświeżenia Cię zachęcam. Niemniej jednak ostrzegam Cię, że książka zawiera również mnóstwo opisów takich elementów programowania, które w oczach doświadczonego programisty mogą wydawać się elementarne. Nie chcę przy tym, abyś wydawał swoje pieniądze na wiedzę, którą już posiadasz, albo co do której masz inne poglądy. Chcę po prostu, aby książka była dostępna dla jak najszerszej grupy odbiorców, także tych, którzy w świecie wieloplatformowego programowania GUI stawiają pierwsze kroki.

Dlaczego akurat wxWidgets? Zdań na temat przydatności i możliwości tej biblioteki jest tyle, ilu jest programistów. Niewątpliwą i pierwszą ze wszystkich zaletą wxWidgets jest jej wieloplatformowość, za którą zmierza także niezwykła łatwość jej użycia. W rozpędzonym świecie coraz bardziej zróżnicowanych usług informatycznych, wykorzystujących różne platformy systemowe, są to cechy kluczowe. Poza tym biblioteka jest nieustannie rozwijana i wzbogacana o atrakcyjne i nowoczesne rozwiązania, które z pewnością spełnią oczekiwania nawet najbardziej wymagającego programisty. W końcu wxWidgets jest też biblioteką opartą na idei *open source*, a dzięki dość liberalnej licencji umożliwia dowolne jej wykorzystanie także w projektach komercyjnych (zob. *Dodatek C)*.

Książka, którą właśnie trzymasz w rękach, absolutnie nie wyczerpuje tematu, o którym ma ambicję traktować. Wymagałoby to stworzenia kilkutomowego dzieła, przy którym sienkiewiczowska *Trylogia* byłaby zaledwie broszurą. W związku z tym zmuszony jestem pokazać Ci tę część fascynującego świata wxWidgets, która pozwoli Ci pewnie i zdecydowanie poczuć się w jej meandrach, jednak poszukiwaniem niektórych szczegółów będziesz musiał zająć się sam (będę się starał zostawiać drogowskazy wszędzie tam, gdzie uznam to za konieczne). Twoją uwagę zechcę skierować na zagadnienia, które Ci pomogą poznać wxWidgets na tyle, abyś potrafił tworzyć za jej pomocą złożone aplikacje oraz wiedział, co zrobić dalej, gdy poczujesz potrzebę samodzielnego zgłębienia jej tajemnic.

Moim celem jest ukazanie Ci możliwości projektowania i tworzenia profesjonalnych, otwartych i komercyjnych programów komputerowych, wyłącznie przy użyciu ogólnie dostępnych i bezpłatnych narzędzi (najczęściej *otwartych*), a także pokazanie tego, w jaki sposób jedna wieloplatformowa biblioteka umożliwia tworzenie kodu współpracującego z różnymi systemami operacyjnymi i różnymi kompilatorami.

Jest nim też popularyzacja biblioteki wxWidgets wśród polskich programistów, mam bowiem wrażenie, że w polskich realiach została ona nieco zmarginalizowana, ale nie przez to, że ustępuje innym bibliotekom tego typu (bo takie stwierdzenie jest pochopne i zdecydowanie mija się z prawdą), a przez dotkliwy dla wszystkich brak jakiejkolwiek literatury w języku polskim, która byłaby jej poświęcona i traktowałaby temat na tyle wyczerpująco, aby po jej lekturze dysponować wachlarzem silnych argumentów w zestawianiu jej z innymi bibliotekami. Samodzielna analiza źródeł w języku angielskim może być zniechęcająca, a brak szybkich i widocznych efektów pracy z nowymi zagadnieniami może często prowadzić do zbyt wczesnego odstawienia narzędzi, które w rzeczywistości oferują potężne i ciekawe rozwiązania. Ta książka ma temu zapobiec.

Zapraszam Cię w podróż, z której nie wrócisz już taki sam...

Podziękowania

Nie pisałbym dzisiaj tych słów, gdybym nie spotkał kilku osób, którym zawdzięczam to, że moja fascynacja wxWidgets ma dziś taki właśnie kształt. Dziękuję mojemu ojcu, który sam będąc żądnym wiedzy o rozwijającym się świecie, w odpowiednim momencie spowodował, że w naszym domu pojawił się znienawidzony przez żeńską część rodziny (rozkochaną w serialach na *dwójce*) mikrokomputer, który stanowił iskrę zapalną dla wspaniałej pasji. Dziękuję też Piotrowi Cymbalukowi za to, że umożliwił mi wczesne wejście do krainy C++ i poznanie tego potężnego języka, a także motywował mnie do badań nad różnymi bibliotekami służącymi do kreowania wydajnych i efektywnych GUI aplikacji komputerowych. Dziękuję Marcinowi Łopacińskiemu, który służył mi niestrudzoną pomocą zawsze wtedy, gdy moja znajomość angielskiego okazywała się niewystarczająca do zgłębienia niektórych technicznych i zaawansowanych meandrów programowania, i z którym odbyłem wiele godzin fascynujących i kształcących programistycznych dyskusji. Szczególne podziękowania kieruję do Pani Edyty Kawali z Wydawnictwa Naukowego PWN, która doprowadziła do zmaterializowania niniejszej książki, a także do Pani Ireny Puchalskiej za wzbudzenie semantycznych wątpliwości, a także za to, że redagując kolejne rozdziały książki, niejednokrotnie dokonała cudu. Dziękuję również recenzentom, którzy dostarczyli mi cennych uwag, które niejednokrotnie miały wpływ na ostateczny kształt treści i przykładów zawartych na jej stronach. Przede wszystkim jednak dziękuję Elżbiecie, która dzielnie znosząc moje – chwilami zbyt wielkie – zaangażowanie w pisanie książki, nieustannie motywowała mnie do przelewania każdego słowa „na papier".

Ta książka nigdy nie powstałaby bez Was i Wam ją dedykuję.

Wprowadzenie

Zakres tematyczny książki

Biblioteka wxWidgets to temat, o którym można mówić bez końca. Niemniej jednak chęć zwięzłego i przejrzystego opowiadania o jej możliwościach zmusza do pewnych kompromisów. W związku z tym wywody, jakie znajdziesz w tej książce, są ograniczone do rozważań nad programowaniem dla platform MS Windows oraz Linux (np. Ubuntu oraz Mint[1]). Ponadto skupiłem się na tych elementach biblioteki, które prawdopodobnie bardzo często będziesz wykorzystywał w niemal wszystkich swoich programach, a także takich, których zrozumienie pomoże Ci na efektywne samodzielne korzystanie z biblioteki w przyszłości, czy też rozwijanie własnych, opartych na niej, komponentów programistycznych. Przykładowe programy oraz kody, jakie wspólnie napiszemy i stworzymy, nie tylko nauczą Cię praktycznego stosowania wxWidgets, ale przekonają Cię również o tym, że wxWidgets może z powodzeniem współistnieć i współpracować z innymi komponentami i bibliotekami w kodzie programu, a samo programowanie z nią jest łatwe, przyjemne i pozwala także na tworzenie czytelnego kodu dobrej jakości.

Ponadto celem książki nie jest wchodzenie w analityczne powtórzenia zagadnień, którymi zajęli się już autorzy innych opracowań traktujących o wxWidgets[2], ani takie, które bezpośrednio wynikają z dokumentacji bibliotek przedstawionych w książce, dlatego możesz przygotować się na odrobinę gawędziarstwa o elementach wxWidgets, ale przez pryzmat empiryzmu i ich przydatności w kontekście omawianych tematów.

Ważna uwaga o konstrukcji książki, czyli zanim kupisz

Podjęcie decyzji o konstrukcji książki opowiadającej o wxWidgets zajęło mi najwięcej czasu, jaki poświęciłem na jej przygotowywanie. Trudność tkwiła nie w doborze zakresu merytorycznego, z którym chcę Cię zapoznać, ale w identyfikacji najbardziej efektywnej oraz przystępnej formy i strategii poznawczej i dydaktycznej. Ostatecznie zdecydowałem się podjąć ryzyko i odejść od encyklopedycznej czy leksykonowej konwencji, jaką stosuje większość autorów, i wpleść poszczególne elementy wxWidgets w pewien ciąg przyczynowo-skutkowy, którego istotą jest wprowadzanie kolejnych zagadnień w miarę pisania i komplikowania kolejnych programów przykładowych. To właśnie praktyką i pisaniem programów będziemy zajmować się przede wszystkim na stronach niniejszej książki (stąd

[1] Wszelkie przykłady i metody zawarte w książce bez problemu działają również w dystrybucji Debian. Nie ukrywam, że zdecydowałem się na wykorzystanie dystrybucji Ubuntu oraz Mint ze względu na ich rosnącą popularność wśród zatwardziałych przeciwników linuksowego pingwina. Powinieneś również pamiętać, że próby uruchomienia przykładów w innych dystrybucjach Linuksa mogą zakończyć się niepowodzeniem, jednak zwykle będzie można to naprawić dzięki kilku prostym zabiegom.

[2] Wiele przydatnych informacji na temat podstaw programowania z biblioteką wxWidgets możesz znaleźć w książce Janusza Ganczarskiego i Mariusza Owczarka pod tytułem *C++. Wykorzystaj potęgę aplikacji graficznych*, wydanej przez wydawnictwo Helion w 2008 r. Autorzy omówili wiele ciekawych zagadnień, wśród których na szczególną uwagę zasługuje przystępne przedstawienie zdarzeń, przybliżenie wykorzystania standardowych okien dialogowych, przystępne i zwięzłe omówienie podstaw aplikacji sieciowych, obsługi grafiki czy elementów aplikacji wielowątkowych.

też jej „praktyczny" tytuł – *Programowanie...*). Wobec powyższego pozycja ta nie jest typową informatyczną monografią. Jest też zdecydowanie jedną z tych książek, które najlepiej jest przeczytać *od deski do deski*, mimo że poszczególne zagadnienia są łatwe do odszukania dzięki zawartemu na końcu książki indeksowi. Będziemy iść stopniowo od szczegółowej analizy wszystkich pojawiających się zagadnień w początkowych rozdziałach książki, ku konkretyzacji rozważań i skupieniu się wyłącznie na omawianych tematach w rozdziałach ostatnich. Cała książka stanowi jeden ciąg, jeśli zatem przeskoczysz dalej i coś wyda Ci się niezrozumiałe, jest wielce prawdopodobne, że odpowiedź na nurtujące Cię pytania znajdziesz w rozdziałach poprzednich.

Jeśli zdecydujesz się podjąć taką przygodę, jestem przekonany, że czytając ostatnie strony, będziesz umiał programować z wxWidgets. Jest to książka z rodzaju tych, z którymi trzeba po prostu się zaprzyjaźnić, i wierzę, że tak właśnie będzie w Twoim przypadku.

Aby zmierzyć się z niełatwą misją przekazania Ci moich doświadczeń i wiedzy, postanowiłem podzielić książkę na kilka części.

W pierwszej z nich przedstawię Ci bibliotekę wxWidgets 3, a także podpowiem, jak ją skompilować i jak przygotować swoje środowisko pracy do tworzenia nowych projektów. Pokażę Ci również, jak łatwa do opanowania i intuicyjna jest ogólna składnia wxWidgets, a także jak czytelna i przejrzysta jest „mechanika" aplikacji tworzonej z wxWidgets. Zobaczysz też, w jaki sposób zastosować w praktyce podstawowe mechanizmy wxWidgets, w tym – niepotrzebnie budzącą lęk (zwłaszcza dzięki początkującym programistom) obsługę zdarzeń.

Druga część książki pomoże Ci wejść łagodnie w niezwykły świat kontrolek GUI, a także różnych przydatnych klas oraz mechanizmów wxWidgets, z którymi będziesz mógł tworzyć swoje pierwsze aplikacje. Napiszemy w niej razem kilka programów, które pomogą Ci zrozumieć, w jaki sposób działają i współpracują ze sobą różne elementy biblioteki. W tej części książki nie omówiłem szczegółowo wszystkich dostępnych elementów wxWidgets, jednak jej lektura spowoduje, że będziesz czuć się swobodnie, stosując składnię stosowaną w bibliotece, a także będziesz dobrze się orientować w stylu programowania i możliwościach wxWidgets. Druga część jest również w pewien sposób wyjątkowa, gdyż wprowadza kolejne elementy wxWidgets w miarę tworzenia programów przykładowych, których pisaniem zajmiemy się przede wszystkim. Oznacza to, że kolejne tematy pojawiają się w niej w porządku wyznaczonym przez konkretne programistyczne potrzeby, a nie według jakiejś z góry przyjętej, swoistej „taksonomii" i systematyki.

Trzecia część książki wprowadza pewne uszczegółowienie i traktuje o bardzo ważnym zagadnieniu, jakim jest wykorzystanie standardowych okien dialogowych i tworzenie własnych. Okna dialogowe są bowiem podstawą efektywnej interakcji między programem komputerowym a jego użytkownikiem. Nie muszę Cię chyba przekonywać, że jest to lektura obowiązkowa. Ponadto znajdziesz w niej całą masę informacji o technikach kontroli danych wejściowych, jakie za pomocą okien dialogowych trafiają do programu.

W czwartej części książki wejdziemy w zaawansowaną analizę niektórych zagadnień wxWidgets. Między innymi omówimy najważniejsze klasy wxWidgets z punktu widzenia przetwarzania danych. Ponadto zdradzę Ci kilka przydatnych rozwiązań, których nie znajdziesz w dokumentacji biblioteki wxWidgets, a które zrodziły się podczas mojego wieloletniego obcowania z jej subtelnymi zawiłościami. Wierzę, że mogą one wzbogacić Twój warsztat i sprawić, że Twoje programy będą bardziej atrakcyjne i różnorodne. Zmierzymy się także z meandrami konwersji typów wxWidgets, a także z zasadami samodzielnego konstruowania i programowania interfejsu użytkownika. Zdradzę Ci tajniki projektowania i tworzenia własnych kontrolek, a także pokażę Ci wiele innych zaawansowanych technik, dzięki którym będziesz mógł w pełni wykorzystać możliwości wxWidgets i realizować swoje programistyczne pomysły. Między wierszami przemyciłem także trochę informacji o narzędziach, które być może ułatwią Ci pracę.

Na ostatnich stronach książki znajdziesz kilka przydatnych dodatków. Dodatek A zawiera krótki leksykon większości standardowych elementów graficznego interfejsu użytkownika, z jakich możesz korzystać, programując z wxWidgets. Oprócz ogólnych informacji o każdej z kontrolek GUI zawiera także przykładowe kody oraz informacje o najważniejszych metodach poszczególnych klas biblioteki. Dodatek B został poświęcony zwięzłemu przedstawieniu najważniejszych (w mojej subiektywnej

ocenie) rozszerzeń biblioteki, jakie są tworzone przez jej sympatyków i jakie znacznie podnoszą jej wszechstronność. Dodatek C zawiera teksty kluczowych licencji związanych z wxWidgets.

Gdybyś potrzebował dotrzeć do konkretnego tematu, możesz zawsze skorzystać z indeksu, który znajduje się na końcu książki.

Dla kogo jest ta książka?

Odpowiedź na to pytanie jest prosta – dla wszystkich, którzy chcą zapoznać się z wxWidgets lub chcą poszerzyć posiadaną już na jej temat wiedzę. Ponadto książka jest dla tych, których intrygują potężne możliwości wykorzystania darmowych narzędzi do tworzenia nowoczesnych aplikacji komputerowych, bez specjalnych ograniczeń co do platformy systemowej.

Niemniej jednak, mimo pełnej otwartości, zakres tematyczny książki narzuca pewne wymagania wobec jej czytelników. Abyś mógł w pełni wykorzystać treści zawarte w książce, powinieneś zatem:

- znać język programowania C++ (w tym umieć praktycznie stosować elementy biblioteki standardowej STL oraz znać zagadnienia programowania zorientowanego obiektowo, a także orientować się w nowościach wprowadzonych wraz z tzw. 11 wersją języka);
- umieć praktycznie stosować środowiska programistyczne Visual C++ dla systemów z rodziny MS Windows i/lub Code::Blocks dla systemów Linux (lub inne, lecz na stronach niniejszej książki skupimy się jedynie na pracy z wymienionymi narzędziami);
- znać język angielski w stopniu umożliwiającym lekturę dokumentacji bibliotek (biblioteki omówione w książce nie mają zbyt wielu opracowań lub nie mają ich w ogóle, jednak wszystkie mają bardzo dobrze opracowaną dokumentację w języku angielskim) lub posiadać umiejętność szybkiego korzystania z tłumaczy internetowych;
- mieć chęć i cierpliwość do systematycznego zgłębiania nowej wiedzy, jaką jest biblioteka wxWidgets, oraz stosowania jej w praktyce.

Aby tworzyć wspaniałe aplikacje okienkowe z użyciem wxWidgets, nie musisz mieć żadnego doświadczenia w kreowaniu tego typu programów z innymi bibliotekami C++ (Windows API, NET. Framework, GTK+, Qt i tak dalej). Swoją przygodę z wxWidgets możesz zacząć „od zera”.

Konwencje przyjęte w książce

Jeżeli jesteś już przyzwyczajony do literatury informatycznej, nie znajdziesz w tej książce żadnych innowacji dotyczących sposobu prezentowania treści.

Fragmenty kodu i wszelkie listingi są złożone czcionką stałej szerokości. Mimo potrzeby częstego, nienaturalnego łamania wierszy, taki sposób zapisu wydaje mi się bardziej dostępny i intuicyjnie łatwiej rozpoznawalny przez programistów. Nie mogę oprzeć się wrażeniu, że samo spojrzenie na tekst pisany czcionką stałej szerokości przełącza umysł programisty na odpowiedni tryb pracy i analizy, co ułatwia przyswajanie i syntezę elementów prezentowanego kodu. Nawet ojciec języka C++, Bjarne Stroustrup, mimo swoich pierwotnych przekonań o wyższości zapisu kodu za pomocą „zwykłych” krojów czcionek, w końcu przekonał się do monotypowej prezentacji kodu w swoich dziełach[3].

Odrębnym zagadnieniem jest konwencja i stylistyka kodu przyjęta na potrzeby programowania z wxWidgets. Poświęcam temu tematowi trochę uwagi w dalszej części książki (zobacz podrozdz. 1.6).

Ponadto wszędzie tam, gdzie moim zamiarem jest przeprowadzenie szczegółowej analizy elementów programu, stosuję numerację wierszy kodu.

Jako że osobiście brzydzę się wszelkimi przejawami notacji węgierskiej (przepraszam wszystkich, którzy są do niej przyzwyczajeni), nie znajdziesz w tej książce nic, co mogłoby mieć z nią cokolwiek wspólnego, chyba że będzie to konieczne wraz z cytowaniem fragmentów kodu biblioteki.

---

[3] Porównaj Bjarne Stroustrup, *Język C++*, wyd. VI, WNT, Warszawa 2002, oraz Bjarne Stroustrup, *Język C++. Kompendium wiedzy*, wyd. IV, Helion, Gliwice 2013.

Wszystkie zagadnienia, którym chciałem nadać większą wagę, w tym również ważne fragmenty kodu, zostały wyróżnione czcionką pogrubioną. Niektóre nazwy, cytaty, słowa kluczowe i inne elementy, w zależności od kontekstu, oznaczyłem kursywą.

Wszystkie nazwy klas, a także schematy i przytoczone deklaracje funkcji, które pojawiają się po raz pierwszy w tekście, wyróżniłem pogrubioną kursywą. W przypadku funkcji typ zwracanej wartości jest jedynie pogrubiony, przy czym zarówno listy parametrów funkcji, jak i informacja o typie zwracanym mogą być całkiem pominięte, zwłaszcza jeśli nie mają znaczenia dla aktualnego wywodu (możesz zawsze je sprawdzić w dokumentacji bilioteki). Jeśli nazwa funkcji jest przywoływana po raz kolejny, używam wobec niej jedynie kursywy, nieraz rezygnując z informacji o typie zwracanych oraz jej liście parametrów.

Choć tym samym stawiam wyzwanie wielu wyznawcom formalnych i nieformalnych konwencji nazewniczych w obrębie programowania w C++, z troską o stylistykę i jasność wypowiedzi w odniesieniu do funkcji globalnych używam terminu *funkcja globalna* lub po prostu *funkcja*, w odniesieniu zaś do funkcji składowych klas – zamiennie terminów *funkcja*, *funkcja składowa* oraz *metoda*, starając się jednak do maksimum ograniczyć ryzyko semantycznych niejasności. Ponadto, gdy długie listy argumentów zacierają czytelność przykładów i schematów kodu skoncentrowanych na konkretnych zagadnieniach, miejscami stosuję wielokropek w obrębie list argumentów funkcji, co jest po prostu symbolicznym skrótem i nie należy tego mylić ze zmienną listą argumentów.

Wszelkie odniesienia do rozdziałów lub ich części, jakie znajdują się w tekście, opierają się na ich numeracji, zgodnej ze spisem treści – na przykład „ zobacz 1.2.2.1”.

Odniesienia do dzieł innych autorów przywołuję głównie w przypisach lub bezpośrednio w treści książki, z uwagi zaś na ogrom zagadnienia, jakim jest programowanie w języku C++, zrezygnowałem z przedstawienia zbiorczej bibliografii przedmiotu, gdyż zakres informacji na ten temat, jakie zawiera niniejsze opracowanie, nie odbiega od zakresu przedstawionego w ogólnodostępnej literaturze dotyczącej C++, która od lat, w przyzwoitej ilości i różnorodności, dostępna jest dla polskiego programisty.

Na koniec dodam, że w trakcie lektury książki możesz spotkać różne ramki, którym nadałem takie oto znaczenie:

Przydatne informacje

W takiej ramce znajdziesz ciekawostki, uwagi oraz wszystkie inne rzeczy, na które warto spojrzeć i które mogą być lub są istotne dla omawianego właśnie tematu.

Uwagi o programowaniu w MS Windows

W takiej ramce będę chciał zwrócić Twoją uwagę na te kwestie programowania z wxWidgets, które są specyficzne dla systemów operacyjnych z rodziny MS Windows.

Uwagi o programowaniu w Linux

W ramce oznakowanej ikoną pingwinka GNU/Linux zamieściłem informacje, które są związane ze specyfiką programowania w systemach Linux.

Zadanie, problem do przemyślenia

Taka ramka zawiera zadania lub problemy do samodzielnego przemyślenia i rozwiązania.

Pomysł

W ramce tego typu zechcę zasugerować Ci pomysły, jakie wydają mi się ciekawe i dotyczą omawianych zagadnień.

Warsztat

Aby zrozumieć i samodzielnie wypróbować wszystko to, z czym zapoznasz się w trakcie lektury, a także byśmy mogli razem stworzyć przykładowe programy, będziesz potrzebować odpowiednio przygotowanego warsztatu. W zależności od systemu operacyjnego, którego używasz, będą to specyficzne dla tych systemów narzędzia. Wykorzystamy również programy dostępne dla obu omawianych platform systemowych.

MS Windows

Microsoft Visual C++

Wśród różnych środowisk programistycznych C++ (IDE) dostępnych dla platformy systemowej MS Windows moim zdecydowanym faworytem jest Visual C++. Oczywiście nie mam zamiaru przekonywać Cię do zmiany Twojego ulubionego IDE, lecz musisz pamiętać, że wszystkie przykłady zawarte w książce zostały przygotowane tak, aby bez problemu można je było kompilować z użyciem środowiska Microsoft Visual C++ pochodzącego z pakietu Microsoft Visual Studio Community lub Microsoft Visual Studio Express for Windows Desktop w wersji 2015, choć nie powinieneś mieć najmniejszych problemów z uruchomieniem ich na nowszych wersjach środowiska. To, z której wersji środowiska będziesz korzystać, jest zatem tylko kwestią Twoich upodobań i nie narzucam tu żadnych ograniczeń.

Visual Studio Community oraz Visual Studio Express for Windows Desktop są darmowymi wersjami środowiska umożliwiającymi (mimo drobnych różnic licencyjnych) wykorzystanie ich zarówno w projektach otwartych, jak i komercyjnych. Ważne jest, że w przypadku tej pierwszej wsparcie narzędzi służących do programowania C++ jest opcjonalne, dlatego trzeba się upewnić, że zostały one poprawnie zainstalowane ze środowiskiem (w rzeczywistości polega to na doinstalowaniu do środowiska narzędzi Visual C++ for Windows Desktop). Niedostępność opcji właściwych dla pełnych i profesjonalnych wersji oprogramowania nie przeszkadza w pełnym wykorzystaniu możliwości biblioteki wxWidgets.

Wybraną wersję środowiska Visual Studio możesz pobrać na stronie internetowej firmy Microsoft.

Wspomniane wyżej środowisko nie jest jedynym dostępnym narzędziem. Inne, jakie są przeznaczone dla systemów operacyjnych z rodziny MS Windows, to Code::Blocks będący doskonałym środowiskiem do programowania z biblioteką wxWidgets, a także Dev-C++[4], który korzysta m.in. z kompilatora MinGW GCC. Wszędzie tam, gdzie jest to konieczne, na wybrane aspekty pracy z kompilatorem MinGW GCC zwracam uwagę przy okazji omawiania programowania w systemach linuksowych Ubuntu i Mint oraz śrdowisku Code::Blocks IDE.

[4] Program ma również swój klon, który jest dedykowany pracy z biblioteką wxWidgets. Jest on upowszechniany pod nazwą wxDevC++. Program, oprócz przygotowanych wstępnie projektów wxWidgets, ma również narzędzie umożliwiające graficzne tworzenie interfejsu użytkownika z użyciem kontrolek dostępnych w wxWidgets.

Linux (Ubuntu i Mint)

Code::Blocks

O systemach operacyjnych Linux można powiedzieć, że są rajem dla programistów, a wszystko to dzięki ogromnej liczbie dostępnych narzędzi i wbudowanych bibliotek. Nie bez znaczenia pozostaje również otwarty kod tych narzędzi i całego systemu operacyjnego. Takie okoliczności aż proszą o niezliczone eksperymenty i testy.

Oprogramowanie Code::Blocks jest dość potężnym kombajnem programistycznym dostępnym zarówno dla systemów operacyjnych Linux, jak i MS Windows, dostarczającym profesjonalnych narzędzi ułatwiających tworzenie i kompilowanie programów pisanych w C++. Będąc użytkownikiem Linuksa, zapewne już dawno odkryłeś swoje ulubione narzędzia, jednak na potrzeby książki wszystkie przykłady i specyficzne problemy, jakie są związane z programowaniem w systemach linuksowych Ubuntu oraz Mint, będę prezentować na podstawie pracy ze środowiskiem Code::Blocks.

Code::Blocks IDE możesz pobrać ze strony projektu (*https://www.codeblocks.org*) lub zainstalować wraz ze wszystkimi wymaganymi komponentami za pomocą standardowego instalatora oprogramowania, w jaki jest wyposażona Twoja edycja Linuksa.

Code::Blocks IDE jest również dostępny w wersji dla systemów operacyjnych z rodziny MS Windows, może być zatem dobrym wyborem, jeżeli zechcesz mieć jak największą gwarancję łatwej przenośności swojego kodu między platformami systemowymi.

Inne

Poniższe narzędzia są dostępne dla wszystkich interesujących nas platform systemowych.

PoEdit

Biblioteka wxWidgets udostępnia czytelny i wydajny system internacjonalizacji aplikacji, dzięki któremu w łatwy sposób możesz dodać do swojego programu tłumaczenia na inne języki. PoEdit jest narzędziem służącym do wygodnego tworzenia i edycji plików tłumaczeń. Jeżeli chcesz programować aplikacje wielojęzyczne z wxWidgets, PoEdit powinien koniecznie znaleźć się w Twoim programistycznym warsztacie.

GIMP

GIMP-a nie trzeba przedstawiać nikomu, kto choć przez chwilę dotknął ogromnego świata wspaniałych narzędzi *open source*. Ten rozbudowany program graficzny wykorzystamy do ubrania naszego programu w piękne ikonki i obrazki. Oczywiście możesz używać swojego ulubionego programu graficznego, jednak przykłady zawarte w książce, a także dołączone do niej materiały dodatkowe, zawierają grafiki utworzone za pomocą programu GIMP.

Materiały dodatkowe do książki

Materiały dodatkowe oraz kody źródłowe programów przedstawionych w książce można pobrać w formie spakowanego archiwum ze strony internetowej:

```
http://it.pwn.pl/Artykuly/Programowanie/
   Programowanie-wieloplatformowe-z-C-i-wxWidgets-3
```

po wpisaniu hasła:

```
w8x18WxWidgetS2e
```

Wszystkie zaprezentowane w książce przykłady, jakie znalazły odzwierciedlenie w materiałach dodatkowych do książki, zostały przygotowane do pracy ze środowiskami programistycznymi MS Visual Studio 2015 (Visual C++) oraz Code::Blocks 16.01. Były one uruchamiane i testowane w systemach operacyjnych MS Windows 7, 8.1 oraz 10, a także w systemie operacyjnym Ubuntu Linux w wersjach 16.04, 16.10 oraz 17.04 i systemie operacyjnym Mint Linux w wersjach 17.3 i 18.1.

Głównymi wersjami biblioteki, użytymi podczas kreowania programów przykładowych, były wxWidgets 3.1.0 oraz 3.1.1, choć w większości przypadków możliwe jest również kompilowanie ich z biblioteką w wersjach 3.0.x.

Jeśli podczas lektury książki i testowania przykładów zawartych w materiałach dodatkowych napotkasz wątpliwości i problemy wymagające bardziej szczegółowych objaśnień, możesz napisać do mnie kilka słów na adres *bartosz.warzocha@gmail.com*. Tak jak każdy w dzisiejszym rozpędzonym świecie, ja również często bywam zajęty, niemniej jednak dołożę wszelkich starań, aby szybko odpowiedzieć na Twoją wiadomość.

Errata i aktualizacje

Kondycja człowieka bezlitośnie kształtuje naturę jego czynów, a te czasami bywają skażone drobnymi pomyłkami. Jako że nie jestem żadnym wyjątkiem od tej reguły, podczas lektury książki możesz sporadycznie spotkać drobne pomyłki literowe, choć bezgranicznie wierzę, że tak nie będzie. Jeśli jednak mimo wszystko tak się stanie, wybacz mi proszę i jeśli zechcesz, poświęć chwilę na napisanie do mnie, aby poinformować mnie o odkrytym błędzie. Wszelkie aktualizacje kodu źródłowego materiałów dodatkowych do książki oraz aktualną erratę możesz znaleźć na stronie internetowej Wydawnictwa Naukowego PWN.

Notka prawna

W pracy wykorzystano zmodyfikowaną ikonę Tux, zmodyfikowaną ikonę żarówki oraz zmodyfikowaną ikonę znaku zapytania, których pierwowzory znajdują się w domenie publicznej. Zostały one wykorzystane jako ikony wybranych ramek informacyjnych umieszczonych w tekście (zob. *Konwencje przyjęte w książce*).

Użyta w książce nazwa Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i innych krajach, natomiast nazwy Microsoft, Windows, Word, Excel, Visual Studio, Visual C++ są zastrzeżonymi znakami towarowymi firmy Microsoft Corporation w Stanach Zjednoczonych i innych krajach. Prawa autorskie dotyczące biblioteki wxWidgets i jej dokumentacji należą do Juliana Smarta, Vadima Zeitlina, Stefana Csomora, Roberta Roeblinga, a także innych członków zespołu wxWidgets (wymienionych w punkcie *Copyrights and Licenses* dokumentacji biblioteki).

Wszystkie materiały dodatkowe do książki (z wyjątkiem komponentów trzecich lub jeśli nie zastrzeżono inaczej) są udostępnione na zasadach licencji wxWidgets (zajrzyj do *Dodatku C*). Oznacza to, że możesz je swobodnie modyfikować, rozpowszechniać i zmieniać na warunkach tej licencji. Przede wszystkim bez ograniczeń możesz je wykorzystać we własnych projektach, dostosowując je do własnych potrzeb.