

Wprowadzenie (15)

- Co odróżnia tę książkę od innych książek o C++? (15)
- Dla kogo jest przeznaczona ta książka? (17)
- Jak korzystać z tej książki? (17)
- Konwencje stosowane w książce (18)
- Dostęp do kodów źródłowych (19)

Część I Wprowadzenie do programowania w języku C++ (21)

Rozdział 1. Podejście zorientowane obiektowo - co je wyróżnia? (23)

- Źródła kryzysu oprogramowania (24)
- Rozwiązanie 1. - wyeliminowanie programistów (28)
- Rozwiązanie 2. - ulepszone techniki zarządzania (30)
 - Metoda wodospadu (31)
 - Szybkie tworzenie prototypu (32)
- Rozwiązanie 3. - projektowanie złożonego i rozwlekłego języka (33)
- Podejście zorientowane obiektowo - czy dostaniemy coś za nic? (34)
 - Na czym polega praca projektanta? (35)
 - Jakość projektu - spójność (37)
 - Jakość projektu - łączność (37)
 - Jakość projektu - wiązanie danych i funkcji (38)
 - Jakość projektu - ukrywanie informacji i kapsułkowanie (40)
 - Sprawa projektowania - konflikty nazewnictwa (41)
 - Sprawa projektowania - inicjalizacja obiektu (42)
 - Czym jest obiekt? (43)
 - Zalety stosowania obiektów (44)
- Charakterystyka języka C++ (45)
 - Cele języka C - wydajność, czytelność, piękno i przenośność (45)
 - Cele języka C++ - klasy ze wsteczną zgodnością z C (47)
- Podsumowanie (50)

Rozdział 2. Szybki start - krótki przegląd języka C++ (53)

- Podstawowa struktura programu (54)
- Dyrektywy preprocesora (56)
- Komentarze (60)
- Deklaracje i definicje (63)
- Instrukcje i wyrażenia (69)
- Funkcje i wywołania funkcji (77)
- Klasy (86)
- Praca z narzędziami programistycznymi (90)
- Podsumowanie (94)

Rozdział 3. Praca z danymi i wyrażeniami w C++ (95)

- Wartości i ich typy (96)
- Typy całkowitoliczbowe (98)
 - Kwalifikatory typu całkowitoliczbowego (100)
 - Znaki (104)

- Wartości logiczne (106)
- Typy liczb zmiennoprzecinkowych (107)
- Praca z wyrażeniami C++ (109)
 - Operatory o wysokim priorytecie (110)
 - Operatory arytmetyczne (111)
 - Operatory przesunięcia (114)
 - Bitowe operatory logiczne (115)
 - Operatory relacji i równości (118)
 - Operatory logiczne (120)
 - Operatory przypisania (122)
 - Operator warunkowy (123)
 - Operator przecinkowy (124)
- Wyrażenia mieszane - ukryte zagrożenia (125)
- Podsumowanie (131)

Rozdział 4. Sterowanie przebiegiem programu C++ (133)

- Instrukcje i wyrażenia (134)
- Instrukcje warunkowe (136)
 - Standardowe formy instrukcji warunkowych (136)
 - Częste błędy w instrukcjach warunkowych (140)
 - Zagnieżdżone instrukcje warunkowe i ich optymalizacja (152)
- Iteracje (158)
 - Zastosowanie pętli WHILE (159)
 - Iteracje w pętli DO-WHILE (167)
 - Iteracje w pętli FOR (170)
- Instrukcje skoków w C++ (173)
 - Instrukcja BREAK (174)
 - Instrukcja CONTINUE (177)
 - Instrukcja GOTO (178)
 - Instrukcje skoków RETURN i EXIT (179)
 - Instrukcja SWITCH (183)
- Podsumowanie (186)

Rozdział 5. Agregacja za pomocą typów danych zdefiniowanych przez programistę (187)

- Tablice jako agregaty homogeniczne (188)
 - Tablice jako wektory wartości (188)
 - Definiowanie tablic C++ (190)
 - Operacje na tablicach (193)
 - Sprawdzanie poprawności indeksów (194)
 - Tablice wielowymiarowe (197)
 - Definiowanie tablic znaków (200)
 - Operacje na tablicach znaków (202)
 - Funkcje łańcuchowe a błędy pamięci (204)
 - Dwuwymiarowe tablice znaków (208)
 - Wypełnienie tablic w algorytmach je wypełniających (210)
 - Definiowanie typów tablicowych (214)
- Struktury jako agregaty heterogeniczne (216)
 - Definiowanie struktur jako typów zdefiniowanych przez programistę (216)

- Tworzenie i inicjalizowanie zmiennych strukturalnych (217)
 - Struktury hierarchiczne i ich komponenty (219)
 - Operacje na zmiennych strukturalnych (220)
 - Definiowanie struktur w programach złożonych z wielu plików (222)
- Unie, typy wyliczeniowe i pola bitowe (223)
 - Unie (223)
 - Typy wyliczeniowe (227)
 - Pola bitowe (229)
- Podsumowanie (233)

Rozdział 6. Zarządzanie pamięcią - stos i sverta (235)

- Zasięg nazw jako narzędzie współpracy (236)
 - Zasięgi leksykalne C++ (236)
 - Konflikty nazw w tym samym zasięgu (237)
 - Stosowanie takich samych nazw w zasięgach niezależnych (241)
 - Stosowanie takich samych nazw w zasięgach zagnieżdżonych (241)
 - Zasięg zmiennych pętli (246)
- Zarządzanie pamięcią - klasy pamięci (246)
 - Zmienne automatyczne (248)
 - Zmienne zewnętrzne (251)
 - Zmienne statyczne (257)
- Zarządzanie pamięcią - zastosowanie sterty (261)
 - Wskaźniki C++ jako zmienne o określonym typie (263)
 - Alokowanie pamięci na sterwie (268)
 - Tablice i wskaźniki (273)
 - Tablice dynamiczne (276)
 - Struktury dynamiczne (290)
- Operacje wejścia i wyjścia na plikach (300)
 - Zapisywanie do pliku (301)
 - Odczyt z pliku (304)
 - Plikowe obiekty wejścia-wyjścia (308)
- Podsumowanie (311)

Część II Programowanie obiektowe w C++ (313)

Rozdział 7. Programowanie w C++ z zastosowaniem funkcji (315)

- Funkcje w C++ jako narzędzie modularyzacji programu (317)
 - Deklaracje funkcji (318)
 - Definicje funkcji (319)
 - Wywołania funkcji (320)
- Promocja i konwersja argumentów funkcji (323)
- Przekazywanie parametrów do funkcji w C++ (326)
 - Przekazanie parametru przez wartość (326)
 - Przekazanie parametrów poprzez wskaźnik (328)
 - Przekazanie parametrów do funkcji charakterystycznej dla C++ - poprzez referencję (336)
 - Struktury (341)
 - Tablice (348)
 - Więcej o konwersjach typów (352)

- Zwracanie wartości z funkcji (355)
- Funkcje wplecione - inline (361)
- Parametry funkcji z wartościami domyślnymi (364)
- Przeciążanie nazw funkcji (370)
- Podsumowanie (377)

Rozdział 8. Programowanie obiektowe z zastosowaniem funkcji (381)

- Kohezja (385)
- Sprzęganie (386)
 - Niejawne sprzężenie (386)
 - Jawne sprzężenie (390)
 - Jak zredukować intensywność sprzęgania? (395)
- Hermetyzacja danych (400)
- Ukrywanie danych (407)
- Większy przykład hermetyzacji danych (413)
- Wady hermetyzacji danych przy użyciu funkcji (422)
- Podsumowanie (425)

Rozdział 9. Klasy w C++ jako jednostki modularyzacji (427)

- Podstawowa składnia definicji klasy (430)
 - Połączenie danych i operacji (430)
 - Eliminowanie konfliktów nazw (435)
 - Implementacja kodów metod poza definicją klasy (439)
 - Definiowanie obiektów przechowywanych w pamięci różnych kategorii (443)
- Kontrolowanie dostępu do komponentów klasy (444)
- Inicjowanie obiektów danej klasy (451)
 - Konstruktory - jako metody (452)
 - Konstruktory domyślne (455)
 - Konstruktory kopiujące (457)
 - Konstruktory konwersji (461)
 - Destruktry (463)
 - Co i kiedy, czyli co naprawdę robią konstruktory i destruktry (468)
 - Widoczność nazw w obrębie klasy i przesłanianie nazw przy zagnieżdżaniu (469)
 - Zarządzanie pamięcią za pomocą operatorów i wywołań funkcji (472)
- Zastosowanie w kodzie klienta obiektów zwracanych przez funkcje (476)
 - Zwrot wskaźników i referencji (476)
 - Zwrot obiektów z funkcji (479)
- Więcej o stosowaniu słowa kluczowego const (482)
- Statyczne komponenty klas (488)
 - Zastosowanie zmiennych globalnych jako charakterystyk klas (489)
 - Czwarte znaczenie słowa kluczowego static (491)
 - Inicjowanie statycznych pól danych (492)
 - Statyczne metody (493)
- Podsumowanie (497)

Rozdział 10. Funkcje operatorowe - jeszcze jeden dobry pomysł (499)

- Przeciążanie operatorów (501)
- Ograniczenia w przeciążaniu operatorów (510)
 - Które operatory nie mogą być poddane przeciążaniu? (510)
 - Ograniczenia typów wartości zwracanych przez funkcje operatorowe (512)
 - Ograniczenia liczby parametrów funkcji operatorowych (514)
 - Ograniczenia wynikające z priorytetu operatorów (515)
- Przeciążone operatory jako komponenty składowe klas (516)
 - Zastępowanie funkcji globalnej metodą należącą do klasy (516)
 - Zastosowanie komponentów klas w operacjach łańcuchowych (519)
 - Zastosowanie słowa kluczowego const (521)
- Analiza przedmiotowa - ułamki zwykłe (523)
- Mieszane typy danych jako parametry (533)
- Funkcje zaprzyjaźnione "friend" (541)
- Podsumowanie (556)

Rozdział 11. Konstruktory i destruktory - potencjalne problemy (557)

- Więcej o przekazywaniu obiektów poprzez wartość (559)
- Przeciążanie operatorów w klasach nie będących klasami numerycznymi (566)
 - Klasa String (567)
 - Dynamiczne zarządzanie pamięcią na stercie (569)
 - Ochrona danych na stercie należących do obiektu od strony kodu klienta (574)
 - Przeciążony operator konkatencji łańcuchów znakowych (574)
 - Zapobieganie wyciekom pamięci (577)
 - Ochrona integralności programu (578)
 - Jak "stać" przejść "tam"? (583)
- Więcej o konstruowaniu kopii obiektów (585)
 - Sposób na zachowanie integralności programu (585)
 - Semantyka referencji i semantyka wartości (590)
 - Konstruktor kopiujący definiowany przez programistę (592)
 - Zwrot poprzez wartość (597)
 - Ograniczenia skuteczności konstruktorów kopiujących (600)
- Przeciążenie operatora przypisania (601)
 - Problem z dodaną przez kompilator obsługą operatora przypisania (602)
 - Przeciążenie przypisania - wersja pierwsza (z wyciekami pamięci) (603)
 - Przeciążenie przypisania - wersja następna (samoprzypisanie) (604)
 - Przeciążenie przypisania - jeszcze jedna wersja (wyrażenia łańcuchowe) (605)
 - Pierwszy środek zapobiegawczy - więcej przeciążania (610)
 - Drugi środek zapobiegawczy - zwrot wartości poprzez referencję (611)
- Rozważania praktyczne - jak chcielibyśmy to zaimplementować? (612)
- Podsumowanie (616)

Część III Programowanie obiektowe przy wykorzystaniu agregacji oraz dziedziczenia (619)

Rozdział 12. Klasy złożone - pułapki i zalety (621)

- Wykorzystywanie obiektów jako danych składowych (623)
 - Składnia C++ dotycząca złożenia klas (625)
 - Dostęp do danych składowych komponentów klasy (626)
 - Dostęp do danych składowych parametrów metody (629)

- Inicjalizacja obiektów złożonych (630)
 - Wykorzystanie domyślnych konstruktorów komponentów (632)
 - Wykorzystanie listy inicjalizującej składowe (638)
- Dane składowe ze specjalnymi właściwościami (644)
 - Stałe dane składowe (645)
 - Dane składowe określone przez referencje (646)
 - Wykorzystywanie obiektów w charakterze danych składowych ich własnej klasy (649)
 - Wykorzystywanie statycznych danych składowych w charakterze składowych ich własnych klas (651)
- Klasy kontenerów (654)
 - Klasy zagnieżdżone (670)
 - Klasy zaprzyjaźnione (673)
- Podsumowanie (676)

Rozdział 13. Klasy podobne - jak je traktować? (677)

- Traktowanie podobnych klas (679)
 - Łączenie cech podklas w jednej klasie (681)
 - Przekazywanie odpowiedzialności za integralność programu do klasy serwera (683)
 - Oddzielenie klas dla każdego rodzaju obiektu serwera (688)
 - Wykorzystywanie dziedziczenia w języku C++ w celu łączenia powiązanych klas (691)
- Składnia dziedziczenia w języku C++ (694)
 - Różne tryby dziedziczenia z klasy bazowej (695)
 - Definiowanie oraz wykorzystywanie obiektów klas bazowych oraz klas pochodnych (699)
- Dostęp do usług klasy bazowej oraz pochodnej (701)
- Dostęp do komponentów bazowych w obiektach klasy pochodnej (706)
 - Dziedziczenie publiczne (706)
 - Dziedziczenie chronione (711)
 - Dziedziczenie prywatne (716)
 - Zwiększanie dostępu do składowych bazowych w klasie pochodnej (718)
 - Domyślny tryb dziedziczenia (720)
- Reguły zakresu widoczności oraz rozwiązywanie nazw przy stosowaniu dziedziczenia (722)
 - Przeciążanie oraz ukrywanie nazw (725)
 - Wywoływanie metody klasy bazowej ukrytej przez klasę pochodną (729)
 - Wykorzystanie dziedziczenia w celu rozwoju programu (733)
- Konstruktory oraz destruktory klas pochodnych (736)
 - Wykorzystanie list inicjalizujących w konstruktorach klas pochodnych (740)
 - Destruktry w przypadku dziedziczenia (743)
- Podsumowanie (745)

Rozdział 14. Wybór pomiędzy dziedziczeniem a złożeniem (747)

- Wybór techniki wielokrotnego wykorzystywania kodu (749)
 - Przykład relacji typu klient-serwer pomiędzy klasami (749)

- Ponowne wykorzystanie kodu poprzez ludzką inteligencję - po prostu zrób to jeszcze raz (753)
- Ponowne użycie kodu poprzez kupowanie usług (755)
- Ponowne wykorzystanie kodu poprzez dziedziczenie (759)
- Dziedziczenie wraz z ponownym zdefiniowaniem funkcji (764)
- Plusy i minusy dziedziczenia oraz złożenia (766)
- Język UML (768)
 - Cele stosowania języka UML (768)
 - Podstawy UML - Notacja klas (772)
 - Podstawy UML - notacja relacji (773)
 - Podstawy UML - notacja dla agregacji oraz uogólnienia (774)
 - Podstawy UML - notacja krotności (776)
- Studium przypadku - wypożyczalnia filmów (778)
 - Klasy oraz ich skojarzenia (779)
- Widoczność klasy oraz podział odpowiedzialności (796)
 - Widoczność klas oraz relacje pomiędzy klasami (797)
 - Przekazywanie odpowiedzialności do klas serwera (799)
 - Stosowanie dziedziczenia (801)
- Podsumowanie (804)

Część IV Zaawansowane wykorzystanie języka C++ (805)

Rozdział 15. Funkcje wirtualne oraz inne zaawansowane sposoby wykorzystania dziedziczenia (807)

- Konwersje pomiędzy klasami niepowiązanymi (809)
 - Ścisła oraz słaba kontrola typów (812)
 - Konstruktory konwertujące (813)
 - Rzutowania pomiędzy wskaźnikami (lub referencjami) (815)
 - Operatory konwersji (816)
- Konwersje pomiędzy klasami powiązanymi poprzez dziedziczenie (817)
 - Konwersje bezpieczne oraz niebezpieczne (818)
 - Konwersje wskaźników oraz referencji do obiektów (824)
 - Konwersje wskaźników oraz referencji występujących w charakterze argumentów (833)
- Funkcje wirtualne - kolejny nowy pomysł (840)
 - Wiązanie dynamiczne - podejście tradycyjne (843)
 - Wiązanie dynamiczne - podejście obiektowe (852)
 - Wiązanie dynamiczne - funkcje wirtualne (861)
 - Wiązanie dynamiczne oraz statyczne (865)
 - Funkcje czysto wirtualne (869)
 - Funkcje wirtualne - destruktory (873)
- Wielodziedziczenie - kilka klas bazowych (875)
 - Wielodziedziczenie - reguły dostępu (877)
 - Konwersje pomiędzy klasami (878)
 - Wielodziedziczenie - konstruktory oraz destruktory (880)
 - Wielodziedziczenie - niejednoznaczności (881)
 - Wielodziedziczenie - grafy skierowane (884)
 - Czy wielodziedziczenie jest przydatne? (885)
- Podsumowanie (886)

Rozdział 16. Zaawansowane wykorzystanie przeciążania operatorów (889)

- Przeciążanie operatorów - krótki wstęp (890)
- Operatory jednoargumentowe (898)
 - Operatory inkrementacji oraz dekrementacji (899)
 - Przyrostkowe operatory przeciążone (907)
 - Operatory konwersji (910)
- Operatory indeksowania oraz wywołania funkcji (918)
 - Operator indeksowania (918)
 - Operator wywołania funkcji (927)
- Operatory wejścia-wyjścia (933)
 - Przeciążanie operatora >> (933)
 - Przeciążanie operatora << (937)
- Podsumowanie (939)

Rozdział 17. Szablony - jeszcze jedno narzędzie projektowania (941)

- Prosty przykład projektowania klas przeznaczonych do wielokrotnego wykorzystania (942)
- Składnia definicji klasy szablonu (951)
 - Specyfikacja klasy szablonu (952)
 - Konkretyzacja szablonu (953)
 - Implementacja funkcji szablonu (955)
 - Szablony zagnieżdżone (962)
- Klasy szablonów z wieloma parametrami (963)
 - Kilka parametrów określających typ (963)
 - Szablony z parametrami określonymi za pomocą stałego wyrażenia (967)
- Związki pomiędzy konkretyzacjami klas szablonów (970)
 - Zaprzyjaźnione klasy szablonów (970)
 - Zagnieżdżone klasy szablonów (974)
 - Szablony ze składowymi statycznymi (977)
- Specjalizacje szablonów (979)
- Funkcje szablonowe (983)
- Podsumowanie (985)

Rozdział 18. Programowanie przy użyciu wyjątków (987)

- Prosty przykład obsługi wyjątków (988)
- Składnia wyjątków w języku C++ (995)
 - Generowanie wyjątków (997)
 - Przechwytywanie wyjątków (998)
 - Deklaracja wyjątków (1005)
 - Przekazywanie wyjątków (1007)
- Wykorzystywanie wyjątków z obiektami (1011)
 - Składnia generowania, deklaracji oraz przechwytywania obiektów (1011)
 - Wykorzystywanie dziedziczenia podczas stosowania wyjątków (1015)
 - Wyjątki zdefiniowane w bibliotece standardowej (1020)
- Operatory rzutowania (1021)
 - Operator `static_cast` (1022)
 - Operator `reinterpret_cast` (1026)

- Operator `const_cast` (1026)
- Operator `dynamic_cast` (1029)
- Operator `typeid` (1032)
- Podsumowanie (1033)

Rozdział 19. Czego nauczyłeś się dotychczas? (1035)

- C++ jako tradycyjny język programowania (1036)
 - Wbudowane typy danych języka C++ (1036)
 - Wyrażenia języka C++ (1038)
 - Przepływ kontroli w programie C++ (1040)
- C++ jako język modułowy (1041)
 - Typy agregacyjne języka C++ - tablice (1042)
 - Typy agregacyjne języka C++ - struktury, unie, wyliczenia (1043)
 - Funkcje C++ jako narzędzia modularyzacji (1044)
 - Funkcje C++ - przekazywanie parametrów (1046)
 - Zakres widoczności oraz klasy pamięci w języku C++ (1048)
- C++ jako język obiektowy (1049)
 - Klasy języka C++ (1050)
 - Konstruktory, destruktory oraz operatory przeciążone (1051)
 - Składanie klas oraz dziedziczenie (1052)
 - Funkcje wirtualne oraz klasy abstrakcyjne (1054)
 - Szablony (1055)
 - Wyjątki (1056)
- Język C++ a konkurencja (1058)
 - Język C++ a starsze języki programowania (1058)
 - Język C++ a Visual Basic (1058)
 - Język C++ a C (1059)
 - Język C++ a Java (1060)
- Podsumowanie (1062)

Dodatki (1063)

Skorowidz (1065)