

Spis treści

O autorach	xi
O recenzencie technicznym	xii
Wprowadzenie	xiii
 Rozdział 1. Początki C++	1
Przepis 1-1. Znajdowanie edytora tekstu	3
Przepis 1-2. Instalowanie Clang w Ubuntu	5
Przepis 1-3. Instalowanie Clang w Windows	7
Przepis 1-4. Instalowanie Clang w OS X	9
Przepis 1-5. Kompilowanie pierwszego programu C++	10
Przepis 1-6. Debugowanie programów C++ przy użyciu GDB w środowisku Cygwin lub Linux	12
Przepis 1-7. Debugowanie programów C++ w OS X	16
Przepis 1-8. Przełączanie trybów kompilacji C++	19
Przepis 1-9. Budowanie z użyciem biblioteki Boost	20
Przepis 1-10. Instalowanie Microsoft Visual Studio	22
 Rozdział 2. Nowoczesny C++	25
Przepis 2-1. Inicjowanie zmiennych	26
Przepis 2-2. Inicjowanie obiektów przy użyciu list inicjalizacyjnych	30
Przepis 2-3. Używanie dedukowania typu	33
Przepis 2-4. Używanie auto w funkcjach	37
Przepis 2-5. Używanie stałych czasu kompilacji	40
Przepis 2-6. Używanie wyrażeń lambda	45
Przepis 2-7. Praca z czasem	56

Przepis 2-8. Istota referencji poprzez l-wartość i r-wartość	61
Przepis 2-9. Używanie zarządzanych wskaźników	71
Rozdział 3. Praca z tekstem	81
Przepis 3-1. Reprezentowanie ciągów przy użyciu literałów	82
Przepis 3-2. Lokalizowanie tekstu widocznego dla użytkownika	88
Przepis 3-3. Wczytywanie ciągów znaków z pliku	97
Przepis 3-4. Wczytywanie danych z pliku XML	101
Przepis 3-5. Wstawianie danych czasu wykonania do ciągów	109
Rozdział 4. Praca z liczbami	113
Przepis 4-1. Korzystanie z typów całkowitoliczbowych w C++	113
Przepis 4-2. Podejmowanie decyzji za pomocą operatorów relacji	119
Przepis 4-3. Łączenie decyzji za pomocą operatorów logicznych	124
Przepis 4-4. Posługiwanie się wartościami szesnastkowymi	128
Przepis 4-5. Manipulowanie bitami przy użyciu operatorów binarnych	131
Przepis 4-6. „Spaceship”, czyli trójdrożny operator porównania	141
Rozdział 5. Klasy	145
Przepis 5-1. Definiowanie klasy	145
Przepis 5-2. Dodawanie danych do klasy	147
Przepis 5-3. Dodawanie metod	149
Przepis 5-4. Korzystanie z modyfikatorów dostępu	152
Przepis 5-5. Inicjalizowanie zmiennych członkowskich klasy	156
Przepis 5-6. Czyszczenie klas	162
Przepis 5-7. Kopiowanie klas	166
Przepis 5-8. Optymalizowanie kodu przy użyciu semantyki przenoszenia	178
Przepis 5-9. Implementowanie funkcji wirtualnych	184
Rozdział 6. Dziedziczenie	187
Przepis 6-1. Dziedziczenie z klasy	187
Przepis 6-2. Kontrolowanie dostępu do członkowskich zmiennych i metod w klasach potomnych	190

Przepis 6-3. Ukrywanie metod w klasach potomnych	196
Przepis 6-4. Używanie polimorficznych klas bazowych	199
Przepis 6-5. Powstrzymywanie przestaniania metod	203
Przepis 6-6. Tworzenie interfejsów	206
Przepis 6-7. Dziedziczenie wielokrotne	209
 Rozdział 7. Kontenery STL	213
Przepis 7-1. Przechowywanie ustalonej liczby obiektów	213
Przepis 7-2. Przechowywanie rosnącej liczby obiektów	216
Przepis 7-3. Przechowywanie zbioru elementów, które są stale modyfikowane	226
Przepis 7-4. Przechowywanie posortowanych obiektów w kontenerze umożliwiającym szybkie wyszukiwanie	230
Przepis 7-5. Przechowywanie nieuporządkowanych elementów w kontenerze zapewniającym bardzo szybkie wyszukiwanie	241
Przepis 7-6. Używanie desygnowanej inicjalizacji w C++20	246
 Rozdział 8. Algorytmy STL	249
Przepis 8-1. Używanie iteratora do definiowania sekwencji wewnątrz kontenera	249
Przepis 8-2. Wywoływanie funkcji dla każdego elementu kontenera	256
Przepis 8-3. Wyszukiwanie maksymalnej i minimalnej wartości w kontenerze	258
Przepis 8-4. Zliczanie wystąpień wartości w sekwencji	265
Przepis 8-5. Wyszukiwanie wartości w sekwencji	269
Przepis 8-6. Sortowanie elementów w sekwencji	270
Przepis 8-7. Wyszukiwanie wartości w zbiorze	273
 Rozdział 9. Szablony	275
Przepis 9-1. Tworzenie szablonu funkcji	275
Przepis 9-2. Częściowe konkretyzowanie szablonu	280
Przepis 9-3. Tworzenie szablonów klas	287
Przepis 9-4. Tworzenie singletonów	290
Przepis 9-5. Obliczanie wartości podczas kompilacji	293
Przepis 9-6. Koncepty w C++20	297

Rozdział 10. Pamięć	299
Przepis 10-1. Używanie pamięci statycznej	299
Przepis 10-2. Używanie pamięci stosu	302
Przepis 10-3. Używanie pamięci sterty	308
Przepis 10-4. Używanie automatycznej pamięci współdzielonej	312
Przepis 10-5. Tworzenie obiektów dynamicznych pojedynczego wystąpienia	316
Przepis 10-6. Tworzenie inteligentnych wskaźników	321
Przepis 10-7. Debugowanie problemów pamięciowych poprzez przeciążanie metod new i delete	331
Przepis 10-8. Obliczanie wpływu zmian kodu na wydajność	340
Przepis 10-9. Poznajowanie wpływu wyboru rodzaju pamięci na wydajność	343
Przepis 10-10. Redukowanie pofragmentowania pamięci	347
Rozdział 11. Współbieżność	363
Przepis 11-1. Używanie wątków do wykonywania równoległych zadań	364
Przepis 11-2. Tworzenie zmiennych o zakresie wątku	371
Przepis 11-3. Uzyskiwanie dostępu do współdzielonych obiektów przy użyciu wzajemnego wykluczania	386
Przepis 11-4. Tworzenie wątków oczekujących na zdarzenia	398
Przepis 11-5. Pobieranie wyników z wątku	406
Przepis 11-6. Synchronizowanie kolejkowanych komunikatów pomiędzy wątkami	411
Rozdział 12. Praca z siecią	427
Przepis 12-1. Tworzenie aplikacji Berkeley Sockets w OS X	428
Przepis 12-2. Tworzenie aplikacji Berkeley Sockets w Eclipse na Ubuntu	432
Przepis 12-3. Tworzenie w Visual Studio aplikacji Winsock 2 dla systemu Windows	437
Przepis 12-4. Tworzenie połączenia gniazd pomiędzy dwoma programami	444
Przepis 12-5. Tworzenie protokołu sieciowego dla łączności pomiędzy dwoma programami	471

Rozdział 13. Skryptowanie	493
Przepis 13-1. Uruchamianie poleceń Lua w Visual Studio C++	494
Przepis 13-2. Tworzenie projektu biblioteki Lua w Eclipse	498
Przepis 13-3. Tworzenie projektu Lua w Xcode	501
Przepis 13-4. Posługiwanie się językiem programowania Lua	503
Przepis 13-5. Calling Lua Functions from C++	516
Przepis 13-6. Wywoływanie funkcji języka C z poziomu Lua	528
Przepis 13-7. Tworzenie asynchronicznych funkcji Lua	535
 Rozdział 14. Programowanie grafiki 3D	543
Przepis 14-1. Wprowadzenie do GLFW	544
Przepis 14-2. Renderowanie trójkąta	547
Przepis 14-3. Tworzenie teksturowanego czworokąta	562
Przepis 14-4. Ładowanie geometrii z pliku	589
Przepis 14-5. Korzystanie z modułów C++20	613
Indeks	617