

Przedmowa (9)

Rozdział 1. Wprowadzenie (15)

- 1.1. Ewolucja języka C++ (15)
- 1.2. Idiomy jako sposób na złożoność problemów (16)
- 1.3. Obiekty lat 90-tych (18)
- 1.4. Projektowanie i język programowania (19)
- Bibliografia (20)

Rozdział 2. Abstrakcyjne typy danych (21)

- 2.1. Klasy (22)
- 2.2. Inwersja obiektowa (25)
- 2.3. Konstruktory i destruktory (28)
- 2.4. Funkcje rozwijane w miejscu wywołania (32)
- 2.5. Inicjacja statycznych danych składowych (34)
- 2.6. Statyczne funkcje składowe (35)
- 2.7. Zakresy i słowo kluczowe const (36)
- 2.8. Porządek inicjacji obiektów globalnych, stałych i składowych statycznych (38)
- 2.9. Słowo const i funkcje składowe (39)
- 2.10. Wskaźniki funkcji składowych (41)
- 2.11. Konwencje programowania (45)
- Ćwiczenia (46)
- Bibliografia (47)

Rozdział 3. Konkretny typy danych (49)

- 3.1. Ortodoksyjna postać kanoniczna klasy (50)
- 3.2. Zakresy i kontrola dostępu (56)
- 3.3. Przeciążanie - zmiana semantyki funkcji i operatorów (59)
- 3.4. Konwersja typu (64)
- 3.5. Zliczanie referencji i zmienne wykorzystujące "magiczną" pamięć (67)
- 3.6. Operatory new i delete (80)
- 3.7. Separacja tworzenia instancji i jej inicjacji (85)
- Ćwiczenia (88)
- Bibliografia (90)

Rozdział 4. Dziedziczenie (91)

- 4.1. Dziedziczenie pojedyncze (93)
- 4.2. Zakresy deklaracji i kontrola dostępu (99)
- 4.3. Konstruktory i destruktory (109)
- 4.4. Konwersje wskaźników klas (112)
- 4.5. Selektory typu (114)
- Ćwiczenia (116)
- Bibliografia (118)

Rozdział 5. Programowanie obiektowe (119)

- 5.1. Funkcje wirtualne (121)

- 5.2. Interakcje destruktorów i destruktory wirtualne (128)
- 5.3. Funkcje wirtualne i zakresy (129)
- 5.4. Funkcje czysto wirtualne i abstrakcyjne klasy bazowe (131)
- 5.5. Klasa kopertowa i klasa listu (133)
- 5.6. Funktory - funkcje jako obiekty (161)
- 5.7. Dziedziczenie wielokrotne (172)
- 5.8. Kanoniczna postać dziedziczenia (182)
- Ćwiczenia (186)
- Przykład iteratora kolejki (187)
- Przykład klas prostej aplikacji bankowej (188)
- Bibliografia (190)

Rozdział 6. Projektowanie obiektowe (191)

- 6.1. Typy i klasy (192)
- 6.2. Czynności projektowania obiektowego (196)
- 6.3. Analiza obiektowa i analiza dziedziny (199)
- 6.4. Związki obiektów i klas (202)
- 6.5. Podtypy, dziedziczenie i przekazywanie (210)
- 6.6. Praktyczne zasady tworzenia podtypów, stosowania dziedziczenia i niezależności klas (229)
- Ćwiczenia (231)
- Bibliografia (232)

Rozdział 7. Ponowne użycie i obiekty (233)

- 7.1. Gdy analogie przestają działać (235)
- 7.2. Projektowanie z myślą o ponownym użyciu (237)
- 7.3. Cztery mechanizmy ponownego użycia kodu (239)
- 7.4. Typy parametryczne czyli szablony (241)
- 7.5. Ponowne użycie i dziedziczenie prywatne (249)
- 7.6. Ponowne użycie pamięci (252)
- 7.7. Ponowne użycie interfejsu - warianty (253)
- 7.8. Ponowne użycie, dziedziczenie i przekazywanie (255)
- 7.9. Ponowne użycie kodu źródłowego (256)
- 7.10. Ogólne uwagi na temat ponownego użycia (259)
- Ćwiczenia (260)
- Bibliografia (261)

Rozdział 8. Programowanie za pomocą przykładów (263)

- 8.1. Przykład - przykłady pracowników (266)
- 8.2. Konstruktory ogólne - idiom zespołu przykładów (271)
- 8.3. Autonomiczne konstruktory ogólne (273)
- 8.4. Abstrakcyjne przykłady bazowe (275)
- 8.5. Ku idiomowi szkieletu przykładu (278)
- 8.6. Uwagi na temat notacji (280)
- 8.7. Przykłady i administracja kodem programu (282)
- Ćwiczenia (283)
- Prosty parser wykorzystujący przykłady (284)

- Przykład wykorzystujący szczeliny (286)
- Bibliografia (288)

Rozdział 9. Emulacja języków symbolicznych w C++ (289)

- 9.1. Przyrostowy rozwój programów w języku C++ (291)
- 9.2. Symboliczna postać kanoniczna (293)
- 9.3. Przykład - ogólna klasa kolekcji (304)
- 9.4. Kod i idiomy obsługujące mechanizm ładowania przyrostowego (308)
- 9.5. Odzyskiwanie nieużytków (318)
- 9.6. Hermetyzacja typów podstawowych (327)
- 9.7. Wielometody i idiom symboliczny (328)
- Ćwiczenia (332)
- Bibliografia (333)

Rozdział 10. Dynamiczne dziedziczenie wielokrotne (335)

- 10.1. Przykład - system okienkowy (336)
- 10.2. Ograniczenia (339)

Rozdział 11. Zagadnienia systemowe (341)

- 11.1. Statyczne projektowanie systemów (342)
- 11.2. Dynamiczne projektowanie systemów (350)
- Bibliografia (365)

Dodatek A Język C w środowisku języka C++ (367)

- A.1. Wywołania funkcji (367)
- A.2. Parametry funkcji (368)
- A.3. Prototypy funkcji (369)
- A.4. Przekazywanie parametrów przez referencję (370)
- A.5. Zmienna liczba parametrów (371)
- A.6. Wskaźniki funkcji (373)
- A.7. Słowo kluczowe const jako modyfikator typu (375)
- A.8. Interfejs z programami w języku C (377)
- Ćwiczenia (389)
- Bibliografia (390)

Dodatek B Reprezentacja figur geometrycznych w języku C++ (391)

Dodatek C Referencje jako wartości zwracane przez operatory (403)

Dodatek D Kopiowanie "bit po bicie" (407)

- D.1. Dlaczego kopiowanie składowych nie rozwiązuje problemu? (408)

Dodatek E Figury geometryczne i idiom symboliczny (409)

Dodatek F Programowanie strukturalne w języku C++ (447)

- F.1. Programowanie strukturalne - wprowadzenie (447)
- F.2. Elementy programowania strukturalnego w języku C++ (448)

- F.3. Alternatywa dla bloków z głęboko zagnieżdżonymi zakresami (451)
- F.4. Rozważania na temat implementacji (455)
- Ćwiczenia (456)
- Gra wykorzystująca idiom strukturalny (457)
- Bibliografia (460)

Spis rysunków (461)

Spis listingów (463)

Skorowidz (467)