

Spis treści

Przedmowa	7
-----------------	---

CZEŚĆ I. PROGRAMOWANIE OBIEKTOWE W TP6.0 DEFINIOWANIE I PRZETWARZANIE OBIEKTÓW STATYCZNYCH

1. Główne cechy programowania obiektowego	9
2. Na czym polega dziedziczenie?	9
2.1. Realizacja dziedziczenia w Turbo Pascalu bez OOP	10
2.2. Rozwiązanie problemu dziedziczenia cech w programowaniu obiektowym	10
3. Definiowanie typów obiektowych i typów obiektowych potomnych	11
3.1. Deklaracja zmiennych typu obiektowego	12
3.2. Deklarowanie zmiennych z wartością początkową typu obiektowego	13
4. W jaki sposób są dostępne pola obiektu?	13
5. Co to są metody?	15
6. Definiowanie metod	15
7. Zasięg metody	16
8. Przekazywanie typów obiektowych przez moduły	18
9. Definiowanie pól lub metod prywatnych	20
10. Na czym polega rozszerzanie obiektów?	21
11. Co wynika z dziedziczenia metod statycznych?	23
11.1. Sposób wiązania metod statycznych z wywołującymi je zmiennymi	23
11.2. Sposób wywoływania metod dziedziczonych	24
12. Co to są metody wirtualne i polimorfizm?	24
13. Zasady zgodności typów obiektowych	25
14. Korzyści ze stosowania obiektów polimorficznych	27
15. Definiowanie metod wirtualnych	28
16. Znaczenie słowa constructor przy definiowaniu typu obiektowego	30
17. Rola konstruktora	31
17.1. Reprezentacja obiektów w pamięci operacyjnej komputera	31
17.2. Co zawiera tablica metod wirtualnych VMT ?	32
18. Automatyczna kontrola wywołań metod wirtualnych	34
19. Podejmowanie decyzji: procedura czy metoda ?	40
20. Budować metodę wirtualną czy statyczną ?	41

CZEŚĆ II. PROGRAMOWANIE OBIEKTOWE W TP6.0 – PRZETWARZANIE OBIEKTÓW DYNAMICZNYCH

21. Co to są obiekty dynamiczne?	43
22. Alokacja i inicjowanie obiektów dynamicznych za pomocą procedury New	44
23. Zwalnianie pamięci przydzielonej obiektom dynamicznym	46
24. Co to jest destruktor?	47
25. Mechanizm zwalniania pamięci za pomocą Dispose z destruktozem jako drugim parametrem wywołania	49
26. Procedury i funkcje w bibliotece TP6.0 związane z przetwarzaniem obiektów	54

CZEŚĆ III. ZMIANY W PROGRAMOWANIU OBIEKTOWYM W WERSJI TP7.0

27. Dyrektywa public	59
28. Słowo kluczowe inherited	59
29. Metody dynamiczne i Tablica Metod Dynamicznych (DMT)	61
29.1. Metody dynamiczne	61
29.2. Tablica Metod Wirtualnych i Tablica Metod Dynamicznych	62
Dodatek A	
BAZA DANYCH NA OBIEKTACH	65
1. Wprowadzenie	65
2. Kolekcja	66
3. Typ TCollection	67
4. Strumienie	70

5. Typ TStream	71
6. Typ TBufstream	73
7. Tworzenie kolekcji i strumienia	74
8. Przykład bazy danych	76

Dodatek B

KRÓTKI PRZEGLĄD RÓŻNIC JĘZYKA TP7.0 W STOSUNKU DO TP6.0	82
1. Dwa zintegrowane środowiska	82
1.1. Nowe cechy środowiska	82
2. Nowe dyrektywy kompilatora	83
3. Parametry procedur lub funkcji deklarowane jako stałe	84
4. Nowe procedury i funkcje w module System	85
5. Rozszerzenia języka dotyczące stosowania parametrów o nie ustalonej liczbie elementów.	88
5.1. Tablice o nie ustalonej liczbie elementów	88
5.2. Parametry łańcuchowe o nie określonej długości łańcucha	91

Dodatek C

PODZIAŁ PAMIĘCI PODCZAS URUCHAMIANIA PROGRAMU NAPISANEGO W JĘZYKU TURBO PASCAL (PLIKU *.EXE) Z POZIOMU SYSTEMU OPERACYJNEGO DOS	93
Indeks	97

Przedmowa

Niniejszy podręcznik powstał z inspiracji studentów Politechniki Wrocławskiej, dla których autorka prowadziła wykłady dotyczące m.in. programowania zorientowanego obiektowo w języku Turbo Pascal. Czytelnik, aby mógł korzystać z tego podręcznika, powinien znać programowanie w języku TP6.0 lub TP7.0 bez przetwarzania obiektów, czyli powinien umieć posługiwać się wszystkimi typami prostymi, typami strukturalnymi (tablice, zbiory, rekordy i pliki), typem wskaźnikowym, korzystać z podstawowych modułów standardowych (Crt, DOS, Graph) oraz tworzyć własne podprogramy (procedury i funkcje) i moduły.

Ze względu na to, że filozofia programowania strukturalnego różni się istotnie od programowania obiektowego, to ostatnie stwarza wielu początkującym problemy. Z doświadczenia autorki wynika, że dużą trudność stanowi zrozumienie istoty „późnego wiązania”, tj. ustalania, która implementacja metody o danym nagłówku ma być w danym momencie wykonana. Rozszerzona kompatybilność typów obiektowych i polimorfizm dają dość duże możliwości programiście, warto je więc poznać, by nie programować „mechanicznie”, opierając się na przykładach. Dla tych, którzy zrozumieli istotę dziedziczenia, programowanie obiektowe nie jest większym problemem niż strukturalne. Prezentowany podręcznik składa się z trzech części:

W *Części I* wprowadzono podstawowe pojęcia związane z programowaniem obiektowym w TP6.0, omówiono definiowanie i przetwarzanie obiektów statycznych, korzystanie z rozszerzonej kompatybilności typów obiektowych. Wyjaśniono również, na czym polega polimorfizm i jak go stosować, a zatem - rolę konstruktora i tablicy metod wirtualnych.

W *Części II* omówiono problemy związane z przetwarzaniem obiektów dynamicznych w TP6.0. Wyjaśniono też rolę destruktorów w prawidłowej gospodarce pamięcią

W *Części III* pokazano, jakie zmiany dotyczące definicji i przetwarzania obiektów wprowadzono w wersji TP7.0. Najważniejsza ze zmian to wprowadzenie tzw. metod dynamicznych.

Dodatek A zawiera krótkie omówienie tej części biblioteki TurboVision, która jest potrzebna do utworzenia bazy danych dla obiektów z możliwością przechowywania jej w pliku dyskowym. Pokazano również przykład bazy danych z zapisem kolekcji obiektów na dysk i jej odczytem. Zasygnalizowano możliwości przetwarzania takiej bazy danych. Celem tego dodatku jest zachęcenie czytelnika do dalszego studiowania tematu oraz pokazanie, że tworzenie takich baz danych na obiektach nie jest zadaniem trudnym.

W *Dodatku B* omówiono krótko podstawowe różnice między językiem TP6.0 a TP7.0 ze szczególnym zwróceniem uwagi na parametry podprogramów typu otwartych tablic i otwartych łańcuchów.

W pracy znajduje się wiele prostych przykładów ilustrujących omawiane zagadnienia. Przez dwie części podręcznika przewija się przykład obiektów graficznych, który jest dostarczany z pakietem TP6.0. Zdaniem autorki jest to przykład dobrze ilustrujący główne cechy i możliwości programowania zorientowanego obiektowo, a fakt, iż jest on w pakiecie TP6.0, ułatwia korzystanie z niego. Autorka uzyskała zgodę firmy BORLAND na zamieszczenie tego przykładu. Są też inne przykłady, zaproponowane przez autorkę, pisane w ten sposób, by dobrze ilustrowały omawiany problem. Przykłady są numerowane zgodnie z numerami rozdziałów, w których występują; jeśli w którymś rozdziale jest więcej przykładów, to są one oznaczane kolejnymi literami alfabetu. Umieszczane są najczęściej na końcu rozdziału. Słowa kluczowe w języku Turbo Pascal są pisane pismem pogrubionym.

Specjalnym znakiem @oznaczono fragmenty, na które należy zwrócić szczególną uwagę; znakiem 9 - te paragrafy, które można pominąć.

Składam serdeczne podziękowania mgr Hannie Mazur z Centrum Informatycznego (Obecnie Wydziałowego Zakładu Informatyki) Politechniki Wrocławskiej za wnikliwe przeczytanie „surowej wersji” tej pracy. Uwzględnienie Jej uwag przyczyniło się do tego, że niniejszy tekst jest znacznie przystępniejszy.

Dziękuję również studentom II roku IB (w roku akademickim 1994/1995), za inspirację napisania tej książki oraz dr. inż. Janowi Kwiatkowskiemu za słowa zachęty i umożliwienie wydania tego podręcznika.

Specjalne podziękowania składam firmie BORLAND za zgodę na wykorzystanie w pracy programów „FigureDemo”, „ListLemo” i modułu „Figures” zawartych w „Borland Pascal 6.01 Users Guide” oraz rysunku 8.2 ze strony 161 „Turbo Pascal 7.0, User's Guide”

Halina Kwaśnicka