

Rada Naukowa „Informatyki Stosowanej”

prof. dr hab. inż. Tadeusz **Janowski**

prof. dr hab. inż. Jerzy **Lipski**

dr inż. Jerzy **Podgórski**

dr Jan **Smółka**

dr inż. Marek **Miłosz**

Redakcja zeszytu:

dr inż. Piotr **Muryjas** (redaktor wydania)

Magdalena **Latkowska**

Jolanta **Szydłowska-Kawa**

Recenzenci:

prof. dr hab. inż. Stanisław **Grzegórski**

dr inż. Marek **Miłosz**

dr Jan **Smółka**

Adres redakcji: Lublin, ul. Nadbystrzycka 36B, tel. (81) 525-20-46

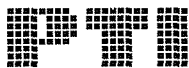
Adres e-mail: GAZETA@pluton.pol.lublin.pl

ISSN 1231-5990

Wydawca: Katedra Informatyki Politechnika Lubelska

Skład: Katedra Informatyki Politechniki Lubelskiej

Druk: Lubelska Drukarnia ALF-GRAF Lublin, ul. Kościuszki 4



Polskie
Towarzystwo
Informatyczne

Koło w Lublinie



LAFI 2001

V Lubelskie Akademiczne
Forum Informatyczne

Kazimierz Dolny, 17-18.05.2001

Organizatorzy:

Polskie Towarzystwo Informatyczne, Koło w Lublinie

Politechnika Lubelska, Katedra Informatyki

Informatyka Stosowana S2/01

DYDAKTYCZNE PROGRAMY KOMPUTEROWE PRZEZNACZONE DO WIZUALNEJ ANALIZY RELACYJNYCH BAZ DANYCH

Paweł Pylak, Paweł Urbanowicz

Katolicki Uniwersytet Lubelski, Katedra Analizy Numerycznej

Streszczenie

W referacie omówione zostały dwa programy dydaktyczne: *Algorytmy* i *Edytor Diagramów BD*, służące do graficznej prezentacji wybranych algorytmów sortowania i wyszukiwania danych oraz do wizualnej analizy relacyjnych baz danych.

1. Wstęp

Problem wizualizacji procesów i zagadnień związanych z przetwarzaniem informacji, w szczególności z relacyjnymi bazami danych, był niejednokrotnie podejmowany przez twórców oprogramowania. W większości przypadków mamy do czynienia z programami komercyjnymi zorientowanymi na wsparcie tworzenia baz danych. Natomiast rzadko można spotkać programy edukacyjne, prezentujące w sposób wizualny procesy przetwarzania danych, a także pozwalające na graficzne przedstawienie struktury relacyjnej bazy danych.

W tworzeniu oprogramowania wizualizującego procesy przetwarzania danych (np. sortowanie, wyszukiwanie), głównym problemem jest sposób graficznej prezentacji działania programu realizującego pewien algorytm. Jako oddzielne zagadnienia, komplikujące graficzną prezentację, można wymienić np. obecność różnych struktur danych, użycie rekurencji, potrzebę dynamicznego przedstawiania procesów, itd.

W programach pozwalających na wizualne tworzenie i analizę relacyjnych baz danych, w tym szczególnie struktury takich baz, zasadniczym problemem jest znalezienie czytelnego i klarownego sposobu wizualizacji bazy. Wykorzystuje się tu diagramy ER lub diagramy uproszczone. Jednak dla większych baz, z wieloma tabelami, problem czytelności staje się poważniejszy. Większość edytorów, także komercyjnych (np. w Access 2000 edytor służący do tworzenia kwerend), nie daje możliwości klarownego prezentowania dużych baz. Często też programy takie nie pozwalają na eksport przygotowanego diagramu do pliku graficznego.

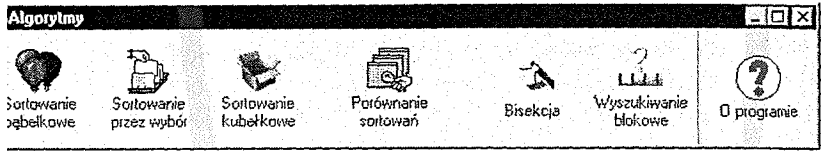
W referacie tym przedstawione zostały dwa programy, w których podjęto próbę opracowania udoskonalonej techniki graficznego przedstawienia struktury relacyjnych baz danych i procesów przetwarzania informacji.

2. Program „Algorytmy”

Program „Algorytmy” przeznaczony jest do wizualizacji działania podstawowych algorytmów sortowania i wyszukiwania danych. W obecnej wersji są to: sortowanie bąbelkowe, sortowanie przez wybór kolejnych minimów, sortowanie kubełkowe oraz

wyszukiwanie blokowe i wyszukiwanie za pomocą metody bisekcji. W przygotowaniu jest szybkie sortowanie (QuickSort).

Główne okno programu wygląda następująco:



Prezentację wybranego algorytmu uruchamia się odpowiednim przyciskiem.

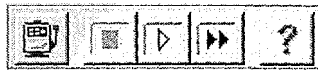
Okna prezentacji algorytmów składają się z następujących części:

- „tablica z danymi”,
- „pseudoprogram” (kod IPL – Informal Programming Language),
- „wartości używanych zmiennych”,
- „panel kontrolny”,
- „regulator prędkości”,
- elementy charakterystyczne dla danego algorytmu.

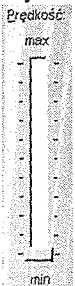
„Tablica z danymi” w sposób wizualny pokazuje zawartość bloku danych, na którym wykonywane są operacje. Poza tym zaznaczone są tu również, o ile to możliwe, wartości zmiennych indeksujących. Program pokazuje, na który element w danym momencie wskazuje taka zmienna. Oprócz tego w przypadku niektórych algorytmów „tablica z danymi” zawiera jeszcze inne dodatkowe informacje. Do „tablicy z danymi” zalicza się również okienko, za pomocą którego ustala się liczbę danych (5 – 22).

„Pseudoprogram” jest to zapis algorytmu w postaci „pseudokodu”. Nie wykorzystuje się tu składni żadnego konkretnego języka programowania, choć w pewnym sensie stosowany tu „pseudokod” przypomina Pascala. W czasie prezentacji algorytmu program *Algoritmy* symuluje wykonywanie się „pseudoprogramu”. Przypomina to opcję śledzenia wykonywania się programów (debugger), dostępną w większości języków programowania.

W części „wartości używanych zmiennych”, która jest dopełnieniem „pseudoprogramu”, pokazywane są aktualne wartości wszystkich używanych w nim zmiennych. Oprócz tego program podaje ilość wykonanych porównań i zamian danych. „Panel kontrolny”



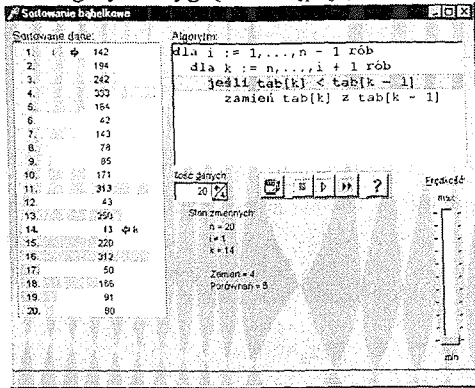
Poszczególne przyciski panelu, to kolejno od lewej: „nowe losowanie”, „stop”, „jeden krok”, „film”, „opis algorytmu”, które służą odpowiednio do: wpisania w tablicę nowych losowo wybranych danych, zatrzymania wykonywania algorytmu, wykonania jednego kroku algorytmu, uruchomienia pseudoprogramu (z szybkością ustaloną w „regulatorze prędkości”) i wyświetlenia okna pomocy ze skrótną informacją o istocie i właściwościach algorytmu. „Regulator prędkości” służy do określania szybkości wykonywania się „pseudoprogramu”, gdy włączony jest tryb „film” (patrz wyżej).



a) *Sortowanie bąbelkowe*

W celu zwiększenia czytelności prezentacji zastosowano najprostszą (pozbawioną wszelkich optymalizacji) wersję algorytmu.

Okno prezentujące ten algorytm wygląda następująco:



Dodatkową informacją, jaką podaje „tablica z danymi” jest uwaga o wymianie danych.

13.	211	Stan zmiennych: ← Te dwie dane → potrzebują zamiany
14.	48	
15.	23 ← k	
16.	229	

b) *Sortowanie przez wybór*

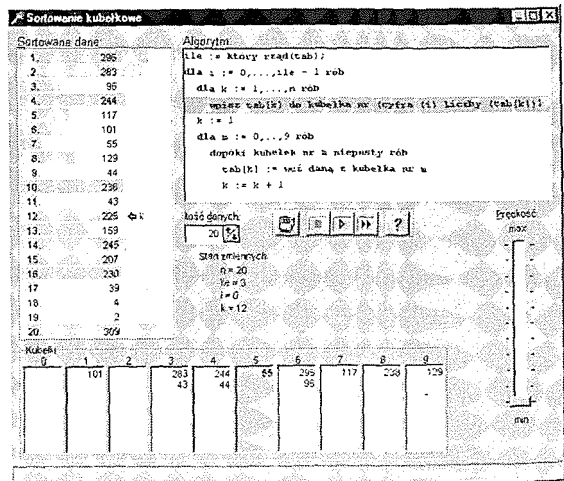
Okno wizualizujące ten algorytm właściwie nie ma żadnych dodatkowych elementów w porównaniu z poprzednim. Oczywiście odmienna jest treść „pseudoprogramu”. Dla tego algorytmu wygląda on następująco:

```

dla i := 1, ..., n - 1 rób
    min := i
    dla k := i + 1, ..., n rób
        jeśli tab[k] < tab[min]
            min := k
    zamień tab[i] z tab[min]
    
```

c) *Sortowanie kubelkowe (pozycyjne)*

Jak widać, w dolnej części okna znajduje się dodatkowy element – panel z kubkami, czyli kolejkami, do których wpisywane są dane mające na odpowiedniej pozycji cyfrę równą numerowi kubelka (dokładniejsze informacje można znaleźć w [3]).



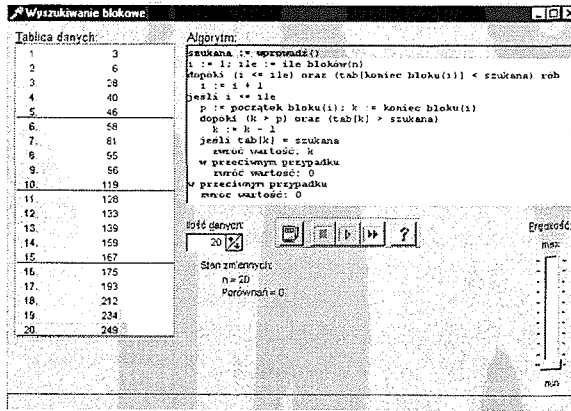
d) Wyszukiwanie metodą bisekcji

Jest to wizualna implementacja znanego algorytmu wyszukiwania za pomocą metody bisekcji, czyli rekurencyjnego dzielenia przedziału na dwa podprzedziały i określania, w którym z nich szukany element może wystąpić. Oczywiście algorytm ten wymaga tablicy z posortowanymi danymi.

Prezentacja ta wygląda podobnie, jak w przypadku sortowań, za wyjątkiem tego, że na początku „pseudoprogram” pyta o liczbę do znalezienia.

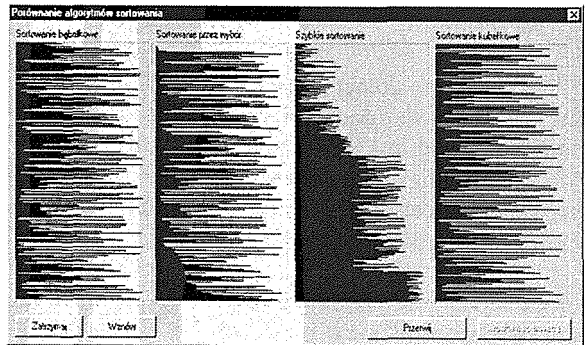
e) Wyszukiwanie blokowe

W algorytmie tym zastosowano podział posortowanej tablicy danych na bloki w taki sposób, by ich wielkość była zbliżona do pierwiastka z liczby wszystkich elementów. Ten sposób dzielenia tablicy jest najbardziej efektywny. W programie poszczególne bloki w „tablicy z danymi” oddzielone są od siebie poziomymi liniami:



Dla wszystkich algorytmów, oprócz wymienionych powyżej okien, dostępne są również okna z informacjami o każdym z nich, zawierające opisy i skrótowo podane właściwości poszczególnych algorytmów. Dostępne są one po naciśnięciu przycisku „opis algorytmu” na „panelu kontrolnym”.

Na głównym panelu programu *Algorytmy* dostępna jest również opcja „Porównanie sortowań”. Po jej wybraniu ukazuje się okienko służące do wizualnego porównania szybkości działania uwzględnionych w programie algorytmów sortowania oraz algorytmu szybkiego sortowania (QuickSort).

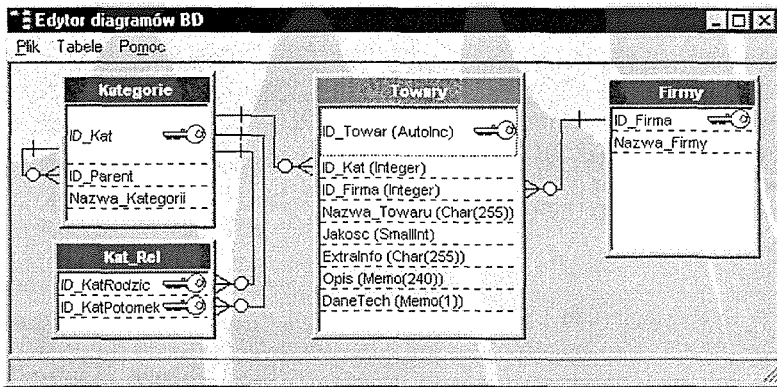


3. Program „Edytor Diagramów BD”

Program „Edytor Diagramów BD” (dalej *Edytor*) jest narzędziem do wizualnego tworzenia, edytowania i prezentacji struktury bazy danych (dalej w skrócie BD). Jest on oparty na uproszczonych diagramach ER.

Tabele przedstawiane są za pomocą prostokątnych okienek z tytułem – nazwą tabeli, zaś poszczególne pola są elementami listy zawartej w takim okienku. Edytor umożliwi również zaznaczanie połączeń (relacji) między tabelami, a dokładniej, między poszczególnymi polami.

Okno programu z przykładowym diagramem:

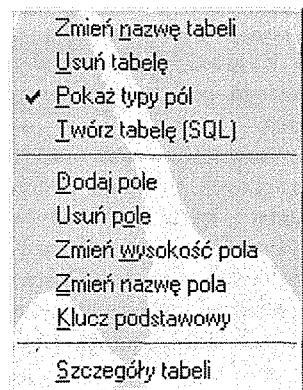


3.1 Edycja projektu

Program *Edytor* umożliwia dość elastyczną edycję poszczególnych elementów struktury BD.

Operacje na tabelach i polach:

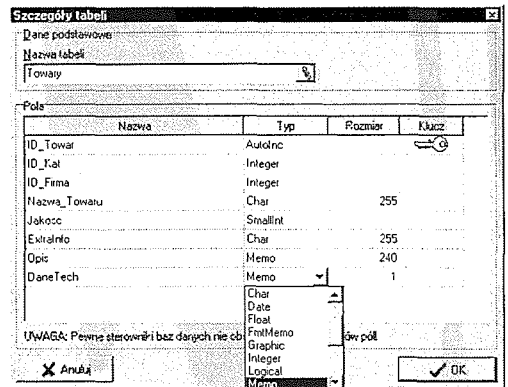
- Tworzenie tabeli i nadanie jej nazwy. Służy do tego np. opcja dostępna w menu Tabele→Nowa tabela.
- Zmiana położenia i rozmiaru okienka tabeli – wykonuje się to tak samo, jak odpowiadające im operacje na oknach w Windows.
- Zmiana koloru nagłówka okienka tabeli (osiągane przez podwójne kliknięcie na nagłówku). Umożliwia to grupowanie tabel w logiczne bloki (o takich samych kolorach).
- Zmiana nazwy tabeli.
- Usunięcie tabeli.
- Włączenie wyświetlania typów pól (przy ich nazwach).
- Dodanie pola.
- Usunięcie pola.



menu podręczne dostępne dla tabeli

- Zmiana wysokości pola.
- Zmiana nazwy pola.
- Zmiana kolejności pól (SHIFT + „przenies i upuść”).
- Zmiana dodatkowych atrybutów pól: opcje „Klucz podstawowy” i „Szczegóły tabeli”.
- Grupowe zaznaczanie tabel (zaznaczanie z trzymanym klawiszem SHIFT). Umożliwia to grupowe przesuwanie lub ustawianie koloru nagłówek.
- Możliwość włączenia opcji kolorowania pól (menu Plik→Koloruj pola). Przy czym pola zgodnych typów mają ten sam kolor.

W oknie „Szczegóły tabeli” (patrz rysunek obok) można zmieniać typy pól, a także ich wielkość i funkcję w tabeli („klucz podstawowy”). W przyszłości będzie tu jeszcze można ustawiać takie atrybuty, jak: NOT NULL, wartość domyślna, wartości minimalna i maksymalna, UNIQUE, itp.



Operacje na połączeniach

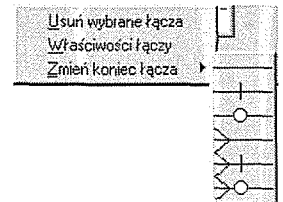
Do utworzenia połączenia używa się metody przeciągania (drag & drop), tzn. myszką „łapie się” wybrane pole i przesuwa na pole docelowe, które musi mieć typ zgodny z typem pola źródłowego. Połączenie można utworzyć również w przypadku, gdy jedno lub obydwa pola nie mają ustalonego typu.

Rysowanie tak utworzonych połączeń odbywa się automatycznie. „Trasę” i wygląd każdego połączenia program *Edytor* ustala samodzielnie na podstawie parametrów łącza.

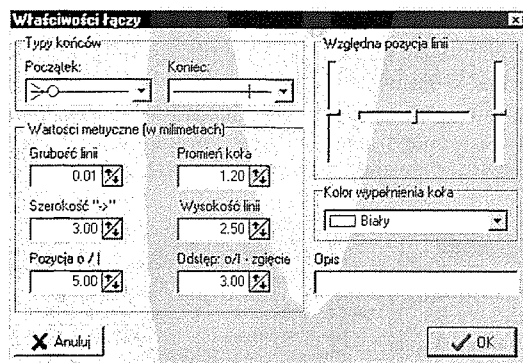
Parametry łącza ustawia się za pomocą opcji dostępnych w menu podręcznym tego łącza. Można tu usunąć łącze, wybrać odpowiednią końcówkę lub wyświetlić okno służące do edycji szczegółowych właściwości łącza. Dostępными końcówkami łącz są (kolejno od góry – patrz na rysunek obok):

- jeden + uczestnictwo nieokreślone,
- jeden + uczestnictwo wymagane (tzn. dokładnie jeden),
- jeden + uczestnictwo opcjonalne (tzn. żaden lub jeden),
- wiele + uczestnictwo nieokreślone,
- wiele + uczestnictwo wymagane (tzn. jeden lub więcej),
- wiele + uczestnictwo nieobowiązkowe (tzn. zero lub więcej).

Oczywiście niezależnie ustala się typ obydwu zakończeń, stąd łatwo można przedstawić różne rodzaje łącz: jeden-do-jednego, jeden-do-wielu, wiele-do-wielu, w różnych



kombinacjach stopni uczestnictwa. Dodatkowe parametry łącz można ustawić w oknie „Właściwości łączy”:



Można tu ustalić wygląd linii i zakończeń, a także w pewnym zakresie „trasę” łącza.

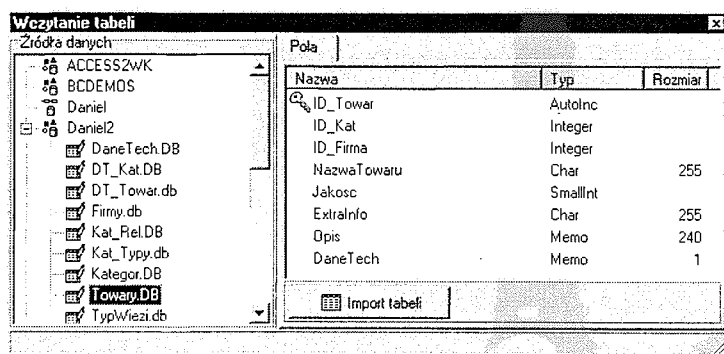
3.2 Operacje importu i eksportu

Zapisywanie i odczytywanie projektów

Projekty tworzone przy pomocy programu *Edytor* mogą być zapisywane na dysku w postaci plików z rozszerzeniem „.edi”. Oczywiście program pozwala na wczytanie i dalsze edytowanie tak zapisanych plików.

Import danych

Poza tym program umożliwia import struktury poszczególnych tabel z istniejących baz danych. Służy do tego opcja z menu Tabele→Importuj tabelę. Po wybraniu tej opcji pojawia się następujące okienko:

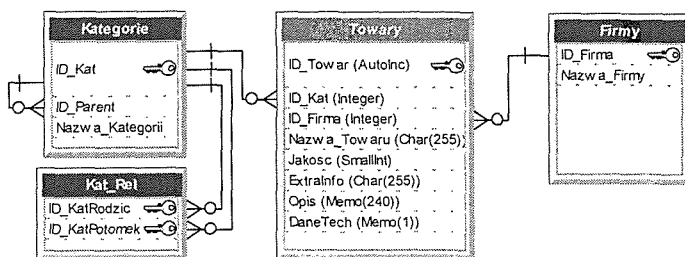


Można tu wybrać bazę danych, z której ma nastąpić import (tu: Daniel2), a następnie konkretną tabelę (tu: Towary.DB). Wtedy program wyświetla podstawowe informacje o tabeli (jak na rysunku). Kliknięcie przycisku “Import tabeli” powoduje wygenerowanie odpowiedniego okienka w projekcie, z odpowiednio ustawionymi polami i kluczami. W kolejnej wersji planowana jest opcja importu całej bazy danych,

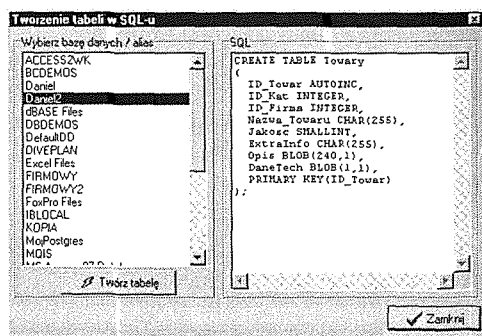
która brałaby pod uwagę także połączenia między tabelami (po polach o identycznych nazwach).

Eksport danych

Program *Edytor* ma możliwość konwersji utworzonego diagramu do postaci graficznej. Używany jest tu format Windows'owych metaplików. Tak utworzony obrazek można albo wydrukować albo nagrać na dysk albo skopiować do schowka i użyć w innym programie, np. w edytorze tekstu. Oto przykład schematu umieszczonego w dokumencie tekstowym:



Poza tym istnieje w programie możliwość wygenerowania zapytania tworzącego wybraną tabelę. Wykonuje się to za pomocą opcji „Twórz tabelę (SQL)” z menu podręcznego tabeli. Po wybraniu tej opcji na ekranie pojawia się specjalne okienko (widoczne obok). Istnieje tu również możliwość rzeczywistego utworzenia takiej tabeli w wybranej bazie danych. Wystarczy tylko zaznaczyć odpowiednią bazę i wcisnąć przycisk „Twórz tabelę”.



4. Zakończenie

Autorzy wyrażają przekonanie, że zaprezentowane w referacie programy mogą być z powodzeniem wykorzystywane do celów edukacyjnych w nauczaniu informatyki. Obecnie wykorzystywane są one na zajęciach z programowania w Pascalu i z baz danych, prowadzonych dla studentów Wydziału Matematyczno-Przyrodniczego KUL.

Literatura

- [1] Cormen T. H., Leiserson C. E., Rivest R. L., Wprowadzenie do algorytmów, WNT, Warszawa 2000.
- [2] Date C. J., Wprowadzenie do systemów baz danych, WNT, Warszawa 2000.
- [3] Sedgewick R., Algorytmy w C++, Wydawnictwo RM, Warszawa 1999.