

Dokumentacja do Projektu 1 Radosław Podgórski

środa, 24 kwietnia 2024 10:48

Opis programu

Program pełni funkcje systemu informatycznego wypożyczalni rowerów. Pozwala na dodawanie nowych klientów i rowerów do bazy danych, usuwanie klientów oraz rowerów, modyfikacje atrybutów rowerów takich jak kolor i cena. Ostatnią opcją jest wyświetlenie posortowanej listy na podstawie wybranego wcześniej atrybutu po którym użytkownik chce posortować. Dane klientów i rowerów przechowywane są w plikach binarnych.

Opis przyjętych założeń

Maksymalne długości pól: Nazwisko klienta i model roweru mogą mieć maksymalnie 30 znaków. Imię klienta i kolor roweru mogą mieć maksymalnie 20 znaków. Numer PESEL max 11 cyfr.

Maksymalna ilość rowerów: 100.

Pliki danych: dane rowerów i klientów są przechowywane w plikach "rowery.bin" i "klienci.bin". Podczas usuwania klienta lub roweru dane są zapisywane tymczasowo do pliku "temp.bin" a następnie oryginalny plik jest usuwany i zastępowany plikiem tymczasowym (usunięcie oryginału i zmienienie nazwy pliku "temp.bin").

Pseudokod istotnych algorytmów

Algorytm scalania

```
scalanie(tablica[], lewy, srodek, prawy, wybor)
  n1 <- srodek - lewy + 1
  n2 <- prawy - srodek

  lewa[], prawa[] <- nowa tablica o rozmiarach n1 i n2

  // Kopiowanie danych do tablic pomocniczych
  for i from 0 to n1-1
    lewa[i] <- tablica[lewy + i]
  for j from 0 to n2-1
    prawa[j] <- tablica[srodek + 1 + j]

  i <- 0
  j <- 0
  k <- lewy

  // Scalanie tablic pomocniczych z powrotem do tablicy głównej
  while i < n1 && j < n2
    if wybor == 1 || wybor == 2
      porównaj lewa[i].kolor/prawa[j].kolor lub lewa[i].model/prawa[j].model
    else if wybor == 3
```

```
porównaj lewa[i].cena/prawa[j].cena
```

```
k++
```

```
// Skopiuj pozostałe elementy z lewej tablicy
```

```
while i < n1
```

```
    tablica[k] <- lewa[i]
```

```
    i++
```

```
    k++
```

```
// Skopiuj pozostałe elementy z prawej tablicy
```

```
while j < n2
```

```
    tablica[k] <- prawa[j]
```

```
    j++
```

```
    k++
```

Algorytm sortowania przez scalanie, rekurencyjny

```
sortowaniePrzezScalanie(tablica[], lewy, prawy, wybor)
```

```
    if lewy < prawy
```

```
        srodek <- lewy + (prawy - lewy) / 2
```

```
        sortowaniePrzezScalanie(tablica, lewy, srodek, wybor)
```

```
        sortowaniePrzezScalanie(tablica, srodek + 1, prawy, wybor)
```

```
    scalanie(tablica, lewy, srodek, prawy, wybor)
```

Opis istotnych struktur danych

Struktura Klient:

- "imie": tablica przechowująca imię klienta

- "nazwisko": tablica przechowująca nazwisko klienta

- "PESEL": tablica przechowująca numer PESEL klienta

Struktura Rower:

- "identyfikator": liczba całkowita przechowująca unikalny identyfikator roweru

- "model": tablica przechowująca model roweru

- "kolor": tablica przechowująca kolor roweru

- "cena": liczba całkowita przechowująca cenę roweru

Opis rozmieszczenia danych w pliku

Każdy rekord w pliku odpowiada jednemu obiektowi typu Klient lub Rower. Struktury są zapisywane do pliku bez separatorów. Odczyt i zapis wykorzystuje strumienie plikowe i operacje binarne. Usuwanie i dodawanie klientów i rowerów odbywa się przy pomocy plików pomocniczych tymczasowych.

