Artur Krawczyk

Programowanie w MicroStation BASIC [cz. 2]

Jednym z podstawowych zadań programowania w systemie CAD jest umożliwienie automatyzacji przetwarzania danych graficznych. MicroStation BASIC zapewnia trzy podstawowe metody dostępu do tego typu danych. Można również stosować czwartą metodę, nieopisywaną w książkowej instrukcji MStB, którą przedstawimy w kolejnych odcinkach kursu.

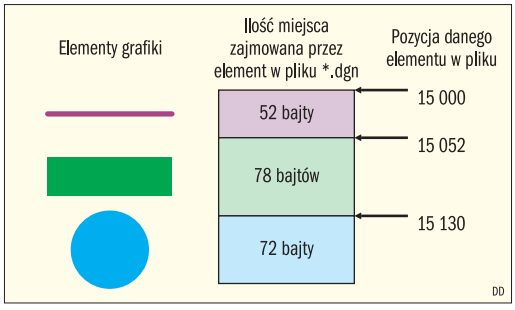
Pierwszą metodą uzyskania dostępu do danych graficznych jest przeszukiwanie całego pliku graficznego, drugą – przeszukiwanie i przetwarzanie danych graficznych wybranych narzędziami Selection Set lub Fence. Trzecia, która wydaje się najbardziej oczywista dla każdego użytkownika systemu CAD, polega na przetwarzaniu pojedynczego elementu grafiki wskazanego kursorem przez użytkownika. W tym numerze CADCAM Forum przedstawimy metodę pierwszą.

Przeszukiwanie wszystkich elementów w pliku graficznym

Przeszukiwanie pliku krok po kroku pozwala na przetworzenie przez program wszystkich elementów grafiki pliku projektowego. Wyszukany element może zostać podświetlony i wówczas sprawdzimy jego cechy: kolor, warstwę, grubość, styl itp. Wybrane cechy elementu można zmodyfikować. Wprowadzenie zmian wiąże się z koniecznością ich zapisu.

Jak widać, do oprogramowania tego zadania potrzeba relatywnie dużo informacji. Dlatego w tym odcinku zajmiemy się napisaniem programu, który przeszukuje plik i podświetla elementy graficzne. Natomiast sprawdzanie warunków cech graficznych obiektów oraz ich modyfikacja zostanie omówiona w kolejnym artykule.

Do przeszukiwania używamy programu, który w pętli dokonuje odczytu pojedynczego elementu, począwszy od pierwszego elementu położnego na początku pliku, aż do końca pliku *.dgn.



Rys. 1.

Nasz program musi więc najpierw zidentyfikować element graficzny, a następnie go odczytać i zapisać w pamięci RAM. Odczytując wybrany element, należy pamiętać, że w programie musi on być reprezentowany w postaci zmiennej. W MStB taką zmienną jest obiekt „MbeElement". Deklarujemy więc użycie zmiennej typu MbeElement, a następnie przyporządkowujemy jej wartość, którą jest właśnie odczytany element grafiki. I nie jest ważne, czy tym elementem będzie lina, kwadrat, czy tekst. Zmienna MbeElement jest bardzo elastyczna, potrafi reprezentować każdy element grafiki odczytany z pliku. Dopiero tak „uchwycony" element grafiki możemy przetwarzać. W programie zostanie on po prostu podświetlony. W następnym kroku pętli operację powtarzamy ponownie dla drugiego elementu grafiki i tak dalej, aż do ostatniego elementu w pliku. Aby napisać program realizujący takie przeszukiwanie, musimy zapoznać się z kilkoma zagadnieniami dotyczącymi programowania. Kluczowe znaczenie ma sposób składowania danych graficznych w pliku *.dgn. Omówimy także nowy obiekt rozszerzeń MicroStation BASIC – MbeElement oraz składnię i sposób wykorzystania pętli „Do...Loop" i instrukcji warunkowej „If".

Dane graficzne w pliku *.dgn

Wielkość pliku graficznego *.dgn zależy od sumy wielkości elementów graficznych, które zostały w nim umieszczone oraz ilości danych niegraficznych, zapisanych w pliku *.dgn. Należy bowiem pamiętać, że w plikach zapisywane są również dane niewidoczne podczas pracy z plikiem *.dgn, zawierające informacje wykorzystywane zarówno przez samo MicroStation, jak i przez inne aplikacje.

Każdemu nowemu elementowi graficznemu nadawany jest numer oznaczający pozycję w pliku. Jego wartość dla danego elementu jest podczas każdego otwarcia pliku obliczana od nowa. Mechanizm obliczania pozycji elementu w pliku wykorzystuje informacje o wielkości elementu graficznego wyrażonej w bajtach. Ilość miejsca zajmowanego przez dany element w pliku

Rys. 2.

Rys. 2a.

zależy od jego typu. Na przykład zapis 1 linii w pliku *.dgn zajmuje 52 bajty, a zapis prostokąta 78 bajtów. Wszystkie elementy wrysowywane do pliku są składowane w określonej kolejności.

Numer pierwszej wolnej pozycji w pliku, w czasie dodawania nowego elementu, jest obliczany na podstawie wielkości (wyrażonej w bajtach) dodawanego elementu. Jeśli pierwszą wolną pozycją jest np. 14500, po narysowaniu linii pierwszą wolną pozycją w pliku będzie 14552.

Na rysunku nr 1 schematycznie przedstawiono zależność między typem elementu graficznego, jego rozmiarem oraz obliczaną pozycją w pliku.

Mechanizm ten umożliwia jednoznaczną identyfikację poszczególnych elementów grafiki w pliku *.dgn. Posiadanie informacji o pozycji danego elementu grafiki pozwala edytować jego własności po załadowaniu go do zmiennej obiektowej MbeElement.

MbeElement

Jednym z podstawowych rozszerzeń Basica w MicroStation jest obiekt MbeElement, przeznaczony do wykonywania operacji na elementach graficznych (rys. 2.). Posiada on około 70 metod i własności. Omawiając je, skupimy się na tych, które zostały wykorzystane do napisania przykładowego programu.

Pierwszą czynnością podczas pisania programu przetwarzającego grafikę jest zadeklarowanie zmiennych. Deklaracja zmiennych powoduje zarezerwowanie określonej ilości pamięci RAM, do przechowania wartości danej zmiennej.

W przypadku deklaracji zmiennej reprezentującej element grafiki wcześniejsze określenie jego wielkości nie jest możliwe. Nie wiadomo, czy program będzie przetwarzał linię (52 bajty), okrąg (72 bajty), prostokąt (78 bajtów), czy inne elementy. Z tego względu za każdym razem gdy do tego obiektu będzie przyporządkowywany element grafiki, reprezentująca go zmienna zostanie określona dopiero po odczytaniu wielkości danego elementu. Deklaracja obiektu musi być więc poprzedzona słowem New, np.:

```
Dim NazwaZmiennej as New MbeElement
```

W opracowywanym programie będziemy wykorzystywać dwie metody:

- .fromFile – Na podstawie informacji o pozycji danego elementu w pliku pozwala na pobranie elementu grafiki z pliku i przeniesienie do obiektu w przestrzeni danych interpretera Basica w MicroStation. Po wykonaniu operacji element grafiki może być przetwarzany w programie. Aby móc skorzystać z tej metody, trzeba podać argument, czyli aktualną pozycję elementu w pliku;
- .display – pozwala na sterowanie wyświetlaniem elementu w poszczególnych oknach widokowych oraz na ustalenie jednego z trzech trybów wyświetlania elementu: podświetlenie elementu MBE_Hilite, usunięcie z widoku elementu MBE_Erase oraz przywrócenie normalnego trybu wyświetlania MBE_NormalDraw, i dwie własności obiektu MbeElement (rys. 2a):
- .isGraphics – pozwala sprawdzić, czy dany element jest grafiką, czy też niegraficznym elementem pliku *.dgn. Przyjmuje ona dwie wartości: TRUE, jeśli dany element jest obiektem graficznym, i FALSE w przeciwnym wypadku;
- .fileSize – pozwala sprawdzić wielkość pamięci (w bajtach) zajmowanej przez dany element w pliku. Dodając do siebie wartości opisujące aktualną pozycję elementu w pliku i wartość wyrażającą rozmiar elementu, uzyskamy informację o pozycji następnego elementu w pliku.

Podstawowe informacje o instrukcjach w Basicu

W przykładowym programie zapoznamy się z działaniem dwóch instrukcji Basica. Pierwszą z nich jest pętla „Do...Loop”, która dopóty wykonuje zestaw danych instrukcji, dopóki nie zostanie spełniony warunek zakończenia pętli.

Istnieje kilka wariantów tej pętli. Jednym z podstawowych jest wykonywanie pętli, dopóki warunek jest prawdziwy (TRUE).

```
Do While warunek instrukcje Loop
```

Najpierw sprawdzany jest warunek, a potem wykonywane są instrukcje zawarte między warunkiem a słowem kluczowym pętli – Loop. Kolejny wariant tej pętli zawiera zamiast słowa „While” słowo „Until”, co powoduje, że pętla jest wykonywana wtedy, kiedy

sprawdzany warunek nie został spełniony (FALSE).

W prezentowanym programie pętla posłuży do sterowania cyklicznym wykonywaniem odczytu elementu grafiki, podświetlaniem odczytanego elementu, wyświetleniem okna dialogowego z informacją o jego wielkości i pozycji w pliku.

Na podstawie tych informacji zostanie obliczona pozycja kolejnego elementu. Pętla jest wykonywana aż do próby odczytu pierwszej nieistniejącej pozycji w pliku. Próba taka powoduje, że zamiast wartości pozycji, zwracany jest kod błędu, którego wartość równa się minus 1. Dzięki temu możemy użyć warunku sprawdzającego pojawienie się wartości ujemnej.

W przypadku liczby dodatniej warunek przyjmuje wartość TRUE, w przypadku liczby ujemnej wartość FALSE i wykonywanie pętli zostaje zakończone. Instrukcja warunkowa „If" służy do „podejmowania decyzji" o wykonaniu alternatywnych instrukcji W języku Basic instrukcję warunkową zapisujemy:

```
If warunek Then Instrukcje1 Else Instrukcje2 End If
```

Instrukcja warunkowa uzależnia wykonywanie zestawu kilku instrukcji od spełnienia określonego przez programistę warunku. W przykładowym programie zastosowano dwukrotnie instrukcję warunkową. W pierwszym przypadku sprawdza ona, czy element odczytany z pliku jest elementem graficznym. Pozwala to oddzielić elementy niegraficzne od graficznych pliku *dgn.

Drugim miejscem, gdzie zastosowano sprawdzanie warunku, jest kontrola przycisku w oknie dialogowym MbeMessageBox. W tym przypadku instrukcja warunkowa sprawdza, czy użytkownik nacisnął przycisk „CANCEL". Gdy ten warunek zostanie spełniony, wykonywanie programu zostanie przerwane.

Aplikacja

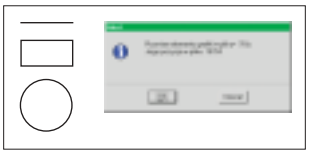
Zadaniem aplikacji prezentowanej w tym odcinku jest wyświetlenie informacji o każdym elemencie graficznym pliku *dgn. Sekwencja czynności wykonywanych przez program jest następująca: znalezienie i podświetlenie elementu graficznego, wyświetlenie informacji w oknie dialogowym o jego pozycji w pliku oraz jego wielkości (w bajtach). Okno dialogowe pozwala na kontynuację przeszukiwania (przycisk OK.). Podświetlony element zostaje wówczas wygaszony, podświetlony zostaje następny element graficzny. Wybór przycisku CANCEL powoduje zakończenie działania programu.

Sub main - Rozpoczęcie części wykonywalnej programu.
Dim Pozycja as long - Deklaracja zmiennej przechowującej numer pozycji elementu grafiki w pliku.
Dim Przycisk as long - Deklaracja zmiennej przechowującej wartość zmiennej zwróconej przez okno dialogowe MbeMessageBox.
Dim Grafika as New MbeElement - Deklaracja zmiennej przechowującej element graficzny.
Dim Iloscb as string - Deklaracja zmiennej przechowującej dane o ilości pamięci zajmowanej przez element graficzny.
Pozycja = Grafika.fromFile (0) - Przeszukiwanie pliku rozpoczynamy od przyjęcia wartości początkowej dla zmiennej Pozycja.
Do while Pozycja >=0 Początek pętli „do" - Definiujemy warunek zakończenia pętli. Jeśli zmienna Pozycja przyjmie wartość mniejszą od zera, wykonywanie pętli zostanie zakończone.
If Grafika.isGraphics then Kontrola warunku - Sprawdzanie, czy załadowany element jest elementem graficznym. Jeśli ten warunek jest spełniony, wykonywane są dalsze instrukcje, jeśli nie - wykonywane są instrukcje umieszczone za pierwszą instrukcją warunkową (*).
Grafika.display MBE_Hilite - Korzystając z metody .display podświetlamy aktualnie pobrany element grafiki.
Iloscb = str$(Grafika.fileSize) - Korzystając z metody .fileSize, odczytujemy liczbę bajtów zajmowaną w pliku przez aktualnie załadowany element grafiki.
Przycisk = MbeMessageBox („Rozmiar elementu grafiki w pliku= " + Iloscb +„ b."+chr$(10)+„Jego pozycja w pliku" + str$(pozycja) ,MBE_OKCancelBox or MBE_InfoIcon) - Wyświetlamy okno dialogowe, w którym definiujemy użycie dwóch przycisków. Informacja o wybranym przez użytkownika przycisku zostanie przechowana w zmiennej Przycisk.
Grafika.display MBE_NormalDraw - Korzystając ponownie z metody .display, przywracamy zwykły tryb wyświetlania.
If Przycisk = MBE_BUTTON_Cancel then - Sprawdzenie, czy w oknie dialogowym MbeMessageBox został przyciśnięty przycisk CANCEL. Jeśli tak, warunek jest spełniony i wykonywana jest instrukcja exit sub, jeśli nie - wykonywane są instrukcje umieszczone po drugiej instrukcji warunkowej (**).
exit sub - Zakończ program (dosłownie wyjdź z programu).
end if (**) - Zakończenie drugiej instrukcji warunkowej if.
end if (*) - Zakończenie pierwszej instrukcji warunkowej if.
Pozycja = Grafika.fromfile (Pozycja + Grafika.fileSize) - Po przetworzeniu danego elementu do zmiennej Grafika ładowany jest nowy element graficzny. Jednocześnie obliczana jest pozycja nowego elementu.
Loop - Koniec zestawu instrukcji wykonywanych w pętli do.
end sub - Koniec programu.

W wyniku uruchomienia prezentowanego programu zostaje podświetlony pierwszy element graficzny w otwartym pliku projektowym oraz wyświetlone okno dialogowe. Okno zawiera informacje o pozycji elementu w pliku oraz o wielkości tego elementu w pliku *.dgn (wyrażonej w bajtach).

Na rys. 3 przedstawiono trzy elementy graficzne, z których drugi jest podświetlony, odczytany i przyporządkowany do zmiennej obiektowej MbeElement). W oknie dialogowym programu wyświetlone zostają informacje o wybranym elemencie, o jego wielkości i pozycji w pliku.

Uruchomienie programu nie powinno zakończyć naszej pracy z programem. Polecam wykonanie kilku dodatkowych ćwiczeń zamieszczonych w ramce. Zostały przygotowane na podstawie przykładowego pliku projektowego znajdującego się razem z przykładowym programem na stronie autora tego artykułu. Istotnym wnioskiem z prezentowanych ćwiczeń jest informacja, że pozycja elementu w pliku *.dgn nie jest stała i zmienia się w czasie edycji danych w pliku projektowym. Do napisania programu edytującego dane w pliku konieczna jest znajomość nowych obiektów MStB oraz własności pliku *.dgn, które zostaną zaprezentowane w kolejnym odcinku naszego cyklu.



Rys. 3.

Ćwiczenie 1. Proszę z pliku projektowego skasować prostokąt i uruchomić program przykładowy. Następnie odczytać pozycje w pliku elementu poprzedzającego (linia) i następnego (okrąg). Plik projektowy należy poddać kompresji (Menu > File >Compress), uruchomić program i porównać pozycje okręgu w pliku przed i po kompresji.

Ćwiczenie 2. Proszę zapisać pozycję linii w pliku *.dgn. Następnie narzędziem Selection Set zaznaczyć linię. W oknie KEYIN wpisać kolejno komendy WSET ADD i WSET DROP. Porównać pozycję linii w pliku, przed i po wykonaniu tej operacji. Komendy te powodują przepisanie danego elementu z dowolnego miejsca w pliku na jego koniec.

Ćwiczenie 3. Ostatnie ćwiczenie polega na wykonaniu kilku modyfikacji istniejących elementów. Można wypełnić kolorem okrąg lub dodać nowy punkt do linii. Następnie proszę porównać wielkość elementu przed i po modyfikacji oraz zauważyć zmianę pozycji w pliku. Ważna jest informacja, które modyfikacje elementu grafiki powodują przepisanie elementu na koniec pliku projektowego, a które nie wpływają na zmianę tej wartości.