

Лабораторна робота № 7

Розробка програми для побудови графіка функції

Мета роботи: вивчити та освоїти на практиці принципи побудови графічних зображень у середовищі Turbo Pascal; розробити програму для побудови графіка заданої функції на заданому інтервалі.

Теоретичні відомості

Як кожна сучасна мова програмування, Turbo Pascal містить засоби для створення екранних зображень. В інженерній практиці найчастіше зустрічаються графіки. Усі засоби створення та управління графічними зображеннями Turbo Pascal зосереджені в модулі Graph.tpu. Таким чином, якщо програма передбачає побудову графічних зображень, вона повинна містити виклик зовнішнього модуля у рядку **uses**:

uses crt, graph;

Пригадаємо, що модуль Crt.tpu (Cathode ray tube) містить процедури та функції, що управляють відображенням інформації на екрані у текстовому режимі.

Модуль Graph.tpu містить опис констант, змінних, процедур та функцій для роботи з Turbo Pascal у графічному режимі, основні з яких наведено в додатку.

Робота з графічною підсистемою звичайно починається з ініціалізації графіки – програмних механізмів з апаратними засобами відеосистеми комп'ютера і переведення останніх в графічний режим відображення. Графічний режим відрізняється від текстового тим, що програма одержує можливість адресувати на знакомісце, а окрему точку (pixel) і керувати її кольором.

Для ініціалізації графіки використовують процедуру InitGraph модуля Graph.tpu.

InitGraph (var GraphDriver: Integer; var GraphMode: Integer; PathToDriver: String).

Змінна GraphDriver передає в процедуру значення, що характеризує тип відеопідсистеми (CGA, VGA, Hercules, тощо).

Змінна GraphMode передає номер графічного режиму, який необхідно ініціювати.

Параметр PathToDriver вказує шлях до каталогу, який повинен містити драйвери графічних відеоадапторів. Якщо програму інстальовано на диску C в каталог BP, то такий шлях, найчастіше, має вигляд: 'C:\BP\BGI' і передається рядковою константою.

При розробці програм для одного комп'ютера, досить вказувати необхідні параметри для змінних безпосередньо в рядку виклику процедури. Якщо програма повинна бути відносно універсальною, використовують засоби автоматичного визначення типу графічної відеосистеми – процедуру **DetectGraph:**

Procedure DetectGraph (var GraphDriver, GraphMode: Integer);

Таким чином операція ініціалізації графічної відеосистеми може мати такий вигляд.

**DetectGraph(GrDr, GrMd);
InitGraph(GrDr, GrMd, 'C:\BP\BGI');**

Змінні GrDr та GrMd повинні бути описані як цілі.

Альтернативним способом можна вважати використання процедури **InitGraph** з константою **Detect** замість першого аргумента:

**GrDr:=Detect;
InitGraph (GrDr, GrMd, 'C:\BP\BGI').**

В результаті виконання одного з двох варіантів ініціалізації, відеосистема переходить у відповідний режим. Результати ініціалізації графіки можна програмно оцінити за допомогою аналізу результатів функцій GraphResult та GraphErrorMsg. Параметри екрану в числовому форматі можна одержати за допомогою функцій **GetMaxX** та **GetMaxY**.

Весь простір екрану, переведеного в графічний режим являє собою сукупність точок, які нумеруються зліва – направо по горизонталі від 0 до

GetMaxX (для відеорежиму VGA - 639) та згори – вниз по вертикалі від 0 до **GetMaxY** (для того ж відеорежиму 479).

Безпосередній вивід графічних об'єктів на екран здійснюється за допомогою процедур, короткий перелік яких наведено нижче, а більш повний опис заголовків дано в додатку.

Procedure PutPixel (X, Y: Integer, Pixel: Word) – засвітити точку з координатами X та Y кольором Pixel;

Procedure Line (x1, y1, x2, y2 :Integer) – проводить лінію між двома точками з використанням встановленого кольору та стилю;

Procedure LineTo (X, Y: Integer) – проводить лінію починаючи з поточної точки і закінчуючи вказаною;

Procedure LineRel (Dx, Dy: Integer) – проводить лінію починаючи з поточної точки на відстань, що визначається вказаними зміщеннями.

Procedure Rectangle (x1, y1, x2, y2: Integer) – будує прямокутник з вказаними вершинами;

Procedure Arc(X, Y: Integer; StartAngle, EndAngle, Radius:Word) – дуга з початком в точці X, Y і відповідними значеннями початкового та кінцевого кутів та радіуса);

Procedure Circle(X, Y: Integer; Radius: Word) – коло з відповідними значеннями координат центра та радіуса;

Для побудови графіка функції необхідно вивести осі координат – за допомогою процедури Line та лінію графіка – за допомогою серії точок з використанням процедури PutPixel. Щоб відмінити прямий вивід на екран текстової інформації, який може бути некоректним в графічному режимі, можна задати значення False змінній DirectVideo:

DirectVideo := False

Розробка програми

Крім безпосередньо графічної частини, правила роботи з якою описано вище, програма повинна містити обчислювальні елементи, що підготують

дані для побудови графіка. Загальний алгоритм програми може бути записаний у такому вигляді.

1. Визначення функції;
2. Визначення інтервалу для побудови графіка функції;
3. Знаходження максимального і мінімального значення функції на заданому інтервалі;
4. Ініціалізація графічної підсистеми;
5. Визначення масштабних коефіцієнтів окремо для горизонтального і вертикального напрямків;
6. Побудова графіка.

Розшифруємо наведені пункти трохи докладніше:

1. Визначення функції. Операція, яка дозволяє деякому математичному виразу присвоїти ідентифікатор з метою полегшення використання в подальшому. Проводиться в блоці описів програми за допомогою ключового слова `Function`. Наприклад:

```
Function Abcd (X : real) : real;  
  Begin  
    Abcd := Sin (X) + Sqr(Sin (4*X));  
  End;
```

2. Визначення інтервалу для побудови графіка функції. Інтервал для побудови графіка функції необхідно одержувати від користувача у діалоговому режимі за допомогою стандартних операторів вводу-виводу. Доцільно включити до програми ряд операторів для перевірки справедливості нерівності $X1 < X2$ і, в протилежному випадку обміну цих змінних значеннями.

3. Знаходження мінімального та максимального значення функції на інтервалі. Необхідно організувати цикл для проходження інтервалу з рівним кроком, що складає $1/10$ або $1/20$ інтервалу. Початкові значення мінімального (Y_{min}) та максимального (Y_{max}) значення функції прирівнюються значенню функції в точці $X1$ – початку інтервалу. Для кожного кроку циклу перевіряється дві нерівності $Y > Y_{max}$ і $Y < Y_{min}$ і, у випадку позитивного результату Y_{max} або Y_{min} одержують нове значення.

Оскільки одержані значення Y_{min} та Y_{max} будуть наближеними, доцільно збільшити одержаний інтервал по Y на 10% по 5% з кожного боку:

$Y_{min} := Y_{min} - (Y_{max} - Y_{min}) * 0.05$

$Y_{max} := Y_{max} + (Y_{max} - Y_{min}) * 0.05$

4. Ініціалізація графічної підсистеми. Ця частина докладно описана в Теоретичних відомостях.

5. Визначення масштабних коефіцієнтів окремо для горизонтального і вертикального напрямків. Оскільки екран монітора в графічному режимі має відомі розміри, а інтервали числового простору, що містить функцію можуть змінюватись в широких межах, необхідно знайти масштабні коефіцієнти, що дозволять вивести графік з максимальним використанням робочого простору екрана. В найпростішому випадку для виводу графіка будемо використовувати весь простір екрану, і виводити будемо тільки графік – без осей. В такому випадку масштабний коефіцієнт по горизонталі можна знайти за формулою:

$Mx := (X2 - X1) / GetMaxX$

Зворотнє перетворення поточного значення X в координату екрана Xs :

$Xs := (X - X1) / Mx$ з обов'язковим перетворенням до типу Integer.

Аналогічно для вертикального напрямку:

$My := (Ymax - Ymin) / GetMaxY$

$Ys := GetMaxY - (Y - Ymin) / My$

теж з перетворенням до типу Integer. Вираз для Ys забезпечує також інвертацію вертикальної осі до загальноприйнятого вигляду.

При розрахунку масштабу, можна використовувати не числа, що повертаються функціями $GetMaxX$ і $GetMaxY$, а інші числа, але обов'язково менші. Це дасть змогу розташовувати графік функції не на усьому екрані, а на виділеній його частині. Одні й ті ж числа повинні використовуватись як для знаходження масштабних коефіцієнтів, так і для перетворення значень в координати екрану.

6. Побудова графіка. Для побудови графіка необхідно на заданому інтервалі вибрати N рівновіддалених точок. Число N доцільно вибирати рівним горизонтальному розміру частини екрану, що виділена під графік. Таким чином, якщо під графік вибрати весь екран, то кількість точок буде рівною

GetMaxX. Графік буде складатися з вказаної кількості точок, виведених на екран процедурою PutPixel.

Завершити програму рекомендується порожнім циклом:

Repeat

Until Keypressed;

Це дозволить призупинити виконання програми після побудови графіка до натискання будь-якої клавіші на клавіатурі.

Завдання

Розробити, скласти та відлагодити програму побудови графіка довільної функції на заданому інтервалі.

Варіанти:

Програма побудови графіка на весь екран без виводу осей.

Програма побудови графіка з виводом осей.

Програма побудови графіка у прямокутнику заданих розмірів.

Зробити висновки.

Рекомендована література:[3], [5], [6].