

ЕЛЕМЕНТИ ГРАФІЧНОГО МОДЕЛЮВАННЯ ВЕКТОРА СИЛИ НА ПЛОЩИНІ З ВИКОРИСТАННЯМ ЗАСОБІВ ПРОГРАМУВАННЯ НА ПК

Oleg Chernysh, Mykola Berezovyy *

* National Agrarian University of Ukraine, Kyiv

Анотація. Запропонований алгоритм графічного представлення заданого вектора сили на площині при застосуванні програмних засобів персонального комп'ютера.

Ключові слова: алгоритм, графічне моделювання, вектор сили

ВВЕДЕНИЕ

Графічне моделювання векторів сил є важливою частиною процесу розв'язання задач механіки і інших галузей науки і техніки, пов'язаних з вибором силової схеми розрахунків.

Прогрес сучасних комп'ютерних технологій значно розширює умови і можливості вирішення проблем графічного моделювання, але вимагає нових знань і практичних навиків у сфері створення і використання програмних засобів графічного відображення інформації. У зв'язку з останніми міжнародними угодами про право використання інтелектуальної власності питання розробки методик програмування і створення власних вітчизняних прикладних програм відповідно до конкретних поставлених умов на сьогоднішній день досить актуальні.

В даній публікації розроблений алгоритм програми графічного моделювання вектора сили на площині на базі системи програмування високого рівня Turbo Pascal 7.0. Для запропонованої методики складання алгоритму можуть бути використані і інші системи програмування високого рівня (C++, VC++, Visual Basic і т.п.) [Марченко и Марченко 1997, Леонтьев 2001]. Крім того, наведена схема алгоритму може бути застосована в якості початкового елементу моделювання більш складних процесів навантаження і дії сил на реальні об'єкти.

ЗМІСТ ДОСЛІДЖЕННЯ

Проблема моделювання заданого вектора сили $\vec{F}$, що виходить з початку координат, на площині включає в себе задачі визначення нульової точки координатних осей на моніторі, зображення ліній координат X , Y , визначення масштабного коефіцієнта зображення величини вектора сили, побудову ліній вектора сили та його напрямку, вивід на екран значень модуля сили, кут його нахилу до горизонту.

Вхідними даними програми, що складається, є початкові параметри будь-якого заданого вектора сили, які повністю описують його величину (модуль) і положення на площині. Результатом роботи програми після вводу початкових параметрів має бути графічне зображення на екрані монітору ПК в координатах X , Y заданого вектора сили, його модуля та кута нахилу до горизонту (рис.1).

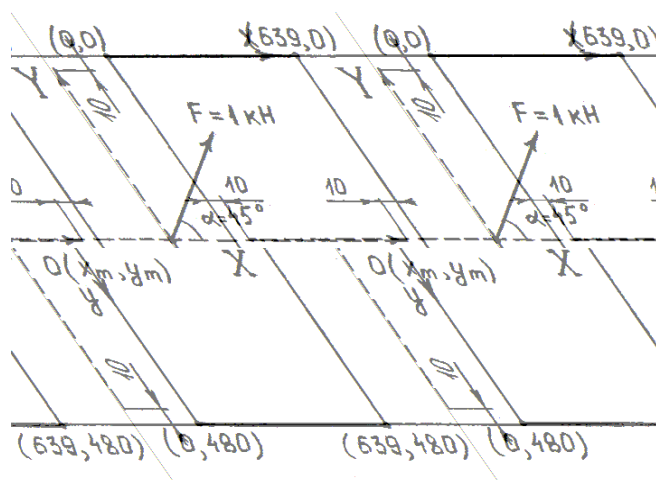


Рис.1. Графічне зображення заданого вектора сили $\vec{F}$ на площині у відповідності до розмірів екрану для VGA монітора

Fig. 1. A graphic image of the set force $\bar{F}$ vector on a plane according to sizes of the screen for the VGA monitor

Програма в інтегрованому середовищі системи Turbo Pascal в загальному вигляді складається із заголовка, що містить службове слово *Program* з назвою програми, розділу описання та розділу операторів (процедур), які відокремлюються службовими словами початку і кінця програми:

$$\begin{array}{l} \text{Program} \langle \text{Ім'я програми} \rangle; \\ \text{Розділ описання}; \\ \text{Begin} \\ \text{Розділ операторів(процедур)}; \\ \text{End.} \end{array} \quad (1)$$

При цьому всі розділи у програмі, оператори, процедури закінчуються знаком розділу у вигляді крапки з комою “;”. Це не стосується окремих випадків і службових слів в операторах циклу, умови, вибору та службового слова *Begin*. Кінець підрозділів програми (*End*) закінчується крапкою з комою, а всієї програми крапкою.

В розділі описання описуються ідентифікатори об’єктів, над якими проводять операції в даній програмі. В розділі операторів (процедур) вказується послідовність дій, які необхідно виконати над описаними об’єктами. Описати ідентифікатор означає вказати його тип. Тип визначає засіб внутрішнього представлення об’єкту і дій, які дозволяються з ним виконувати. Особливо треба відмітити такі типи ідентифікаторів як: цілий тип – *integer*, дійсний тип – *real*, логічний тип – *boolean*. Розділ описань обов’язково має складатися з розділу описань модулів – окремо виконаних і відкомпільованих стандартних програмних одиниць, розділу описань змінних та розділу описань констант, які відповідно починаються службовими словами:

```
Uses <назва модулів>;  
Var <перелік змінних із вказанням їх типу>;  
Const <перелік констант із вказанням їх типу>;.
```

(2)

У програмі поставленої задачі застосовуються стандартні модулі бібліотеки Turbo Pascal: *Crt* – для роботи в текстовому режимі, зчитування символів з клавіатури, очистки екрана; *Graph* – для роботи в графічному режимі при виводі графічної інформації на екран.

При запрошенні на введення початкових параметрів роботи програми застосовується вбудована процедура модулю *Crt*:

```
Write(<назва параметру вводу>);
```

(3)

а для безпосереднього введення параметрів у програму – процедура

```
Read(<значення параметру вводу>);.
```

(4)

Для зображення системи координат X,Y і ліній вектора сили в графічному режимі застосуємо процедури та функції, що включені в стандартний модуль *Graph.Tpu* і стають доступними тільки після об’явлення в програмі пропозиції *Uses Graph*. Налаштування графічних процедур на роботу з конкретним адаптером буде досягнуто за рахунок підключення необхідного драйверу. Графічний драйвер керує графічним адаптером і розміщується, як правило, в окремому підкаталозі *BGI* у вигляді файлів з розширенням *.BGI*. При використанні константи *Detect = 0* відбувається автовизначення типу драйверу і графічного режиму.

Для ініціалізації графічного режиму адаптеру використовується процедура з форматом звернення:

```
Initgraph(<драйвер>, <режим>, <шлях>);
```

(5)

де:

- <драйвер> – змінна типу *integer*, що визначає тип графічного драйверу;
- <режим> – змінна типу *integer*, що визначає режим роботи графічного адаптеру;
- <шлях> – вираз типу *string* (рядок), що визначає шлях звертання до файлу драйвера.

Більшість графічних процедур і функцій програми використовують внутрішній указник поточної позиції курсору на екрані, який, на відміну від текстового курсору, невидимий. Положення цього указника, як і взагалі будь-які координати на графічному екрані, задається відносно лівого верхнього кута, що має координати $(0,0)$. Таким чином, при побудові графічних елементів на екрані треба враховувати, що горизонтальна координата екрана збільшується зліва направо, а вертикальна – зверху до низу (рис.1).

Визначення центру координат x_m , y_m для даної задачі відбувається через оператор присвоєння за виразом:

$$\begin{aligned} x_m &:= \text{getmaxx} \text{ div } 2; \\ y_m &:= \text{getmaxy} \text{ div } 2; \end{aligned} \quad (6)$$

де:

- getmaxx*, *getmaxy* – функції, значеннями яких є максимально можливі для даного драйверу та режиму координати x та y ;
- div* – арифметична операція ділення без остачі.

Слід зауважити, що при визначенні будь-яких значень координат для графічних побудов необхідно використовувати тільки цілі значення типу *integer*, оскільки вони безпосередньо пов'язані із значеннями координат точки на екрані монітору у пікселях.

Для зображення прямих відрізків на екрані використаємо процедуру *Line*, що креслить пряму лінію з вказаними координатами початку і кінця. Формат звернення до процедури:

$$\text{Line}(x_1, y_1, x_2, y_2); \quad (7)$$

де:

- x_1 , y_1 – координати початку прямої лінії;
- x_2 , y_2 – координати кінця прямої лінії. При цьому лінія креслиться із заданим стилем та кольором або за умовчанням – із поточними параметрами.

Встановити тип лінії можливо при застосуванні процедури *Setlinestyle* з форматом звернення:

$$\text{Setlinestyle}(\text{<mun>}, \text{<зразок>}, \text{<товщина>}); \quad (8)$$

де:

$\langle \text{тип} \rangle$, $\langle \text{зразок} \rangle$, $\langle \text{товщина} \rangle$ – параметри, для завдання яких використовуються спеціальні константи, описані в модулі *Graph*, або відповідні їм номери.

Наприклад, процедура *Setlinestyle* (0,0,1) встановлює лінію типу *Solidln* = 0 (суцільна лінія), стандартного зразка = 0, товщиною *NormWidth*=1 (товщина в один піксель).

При встановленні кольору застосовується процедура *Setcolor* з форматом звернення:

$$\text{Setcolor}(\langle \text{колір} \rangle); \quad (9)$$

де:

$\langle \text{колір} \rangle$ – параметр для задання кольору в модулі *Graph*, значення якого можна вибрати від 0 = *blac* (чорний) до 15 = *white* (білий).

По умовчанню використовується колір поточного значення, встановлений на початку роботи графічного режиму адаптеру.

У зв'язку з вищевказаним, графічне зображення координатних осей *X*, *Y* з початком у центрі екрану в нашому випадку можна описати наступним чином:

$$\begin{aligned} &\text{Setlinestyle}(3, 0, 1); \\ &\text{Line}(x_m, 10, x_m, \text{getmaxx} - 10); \\ &\text{Line}(10, x_m, \text{getmaxx} - 10, y_m); \end{aligned} \quad (10)$$

$$\begin{aligned} &\text{Setlinestyle}(0, 0, 1); \\ &\text{Line}(x_m, 10, x_m, x_m - 3, 25); \\ &\text{Line}(x_m, 10, x_m, x_m + 3, 25); \\ &\text{Line}(\text{getmaxx} - 10, y_m, \text{getmaxx} - 25, y_m - 3); \\ &\text{Line}(\text{getmaxx} - 10, y_m, \text{getmaxx} - 25, y_m + 3); \end{aligned} \quad (11)$$

де:

(10) – визначає вертикальні та горизонтальні відрізки координатних осей;

(8) – визначають стрілки осі *Y* та *X*.

Для позначення ліній координат символами *X* та *Y* застосуємо процедуру *Outtextxy*, що підтримує вивід текстової інформації в графічному режимі. Формат звернення до процедури:

$$\text{Outtextxy}(x_p, y_p, \langle \text{текст} \rangle); \quad (12)$$

де:

x_p, y_p – координата початку точки виводу;

$\langle \text{текст} \rangle$ – вираз типу *string* (строка).

Забезпечити вивід тексту різними шрифтами в горизонтальному та вертикальному виконанні можна за допомогою процедури з форматом звернення:

$$\text{Settextstyle}(\langle \text{шрифт} \rangle, \langle \text{напрямок} \rangle, \langle \text{розмір} \rangle); \quad (13)$$

де:

$\langle \text{шрифт} \rangle, \langle \text{напрямок} \rangle, \langle \text{розмір} \rangle$ – параметри коду (номера) відповідного шрифту, його напрямку та розміру.

Наприклад, процедура $\text{Settextstyle}(0,0,1)$; встановлює стандартний шрифт $\text{DefaultFont} = 0$, горизонтального напрямку $\text{HorizDir} = 0$, нормального розміру $\text{NormSize} = 1$.

В нашому випадку позначення осі X та Y можна описати наступним чином:

$$\begin{aligned} &\text{Settextstyle}(1,0,1); \\ &\text{Outtextxy}(\text{getmaxx} - 20, y_m + 10, 'X'); \\ &\text{Outtextxy}(x_m - 20, 10, 'Y'); \end{aligned} \quad (14)$$

Аналогічно за допомогою вищевказаних процедур можна графічно відобразити і заданий вектор сили. Але при цьому необхідно встановити зв'язок між його величиною і координатами точок зображення на екрані.

Масштабний коефіцієнт k для зображення векторів сил визначимо із міркувань максимального зображення по вертикалі його модуля із середини екрана, розмірами $B \times H$ пікселей:

$$k = \frac{H}{2F}. \quad (15)$$

Визначимо геометричні параметри, які необхідні і достатні для завдання на площині вектора сили будь-якої величини та напрямку при умові його відображення з початку координат.

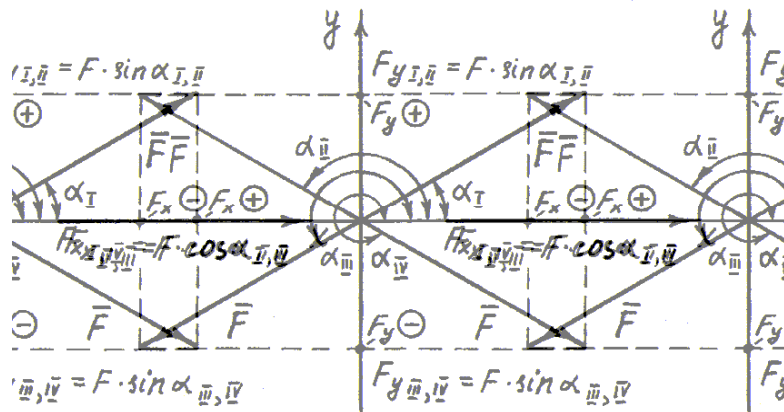


Рис. 2. Зв'язок знаків проекцій вектора сили із знаками синуса та косинуса кута його нахилу в різних квадрантах

Fig. 2. Link of signs of force vector projections with signs of sine and cosine of its declination corner in different quadrants

Аналізуючи знаки проекцій вектора сили $\vec{F}$ в залежності від його кут нахилу до осі X , починаючи з першого квадранту (рис.2), можна зробити висновок, що заданий вектор буде повністю описаний на площині через його проекції $F_{x,y}$ і кут нахилу α , який задається в межах значень $0^\circ - 360^\circ$. Виходячи з цього, за основні змінні для програми, що мають бути описані на її початку, приймаємо вхідні значення модуля f вектора сили $\vec{F}$, його кут нахилу a до осі X , а також поточні координати кінця вектора $\vec{F}$ – x, y , яким присвоюють значення проекцій $f_{x,y}$ вектора сили на відповідні осі:

$$\begin{aligned} x &:= \text{round}(k \cdot (\cos(a \cdot \pi / 180))); \\ y &:= \text{round}(k \cdot (\sin(a \cdot \pi / 180))); \end{aligned} \quad (16)$$

де:

$\text{round}(f_{x,y})$ – стандартна функція округлення значення $f_{x,y}$ до цілого значення типу *integer*;

k – масштабний коефіцієнт; f – значення модуля вектора сили $\vec{F}$;

a – значення куту нахилу α вектору сили $\vec{F}$;

π – число

π – постійна. Слід відмітити, що функції $\sin(a)$, $\cos(a)$ в Turbo Pascal потребують значення куту a в радіанах.

Тоді будь-який заданий вектор сили $\vec{F}$ можна відобразити на моніторі через його координати початку x_m, y_m і кінця – x, y за допомогою процедури:

$$\text{Line}(x_m, y_m, x_m + x, y_m - y); \quad (17)$$

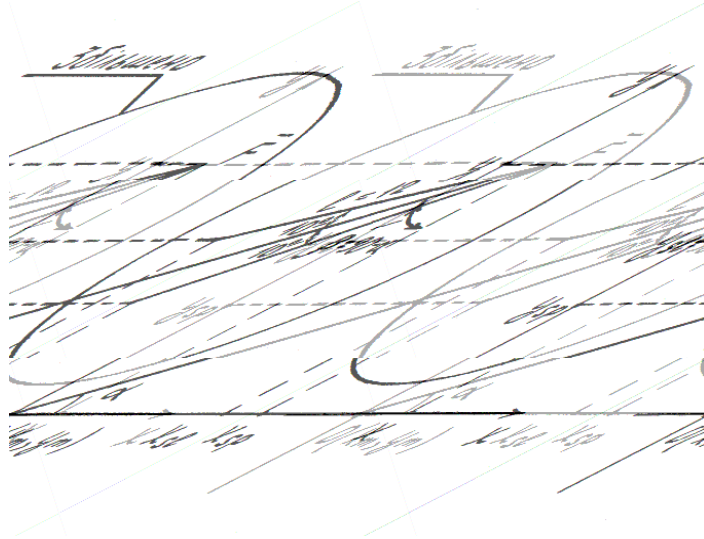
Для позначення напрямку вектора сили використовується аналогічна процедура зображення ліній стрілок з координатами $x_{sp}, y_{sp}, x_{sl}, y_{sl}$, які пов'язані з координатами x, y кінця заданого вектора (рис. 3).

Якщо задатись довжиною лінії стрілки $l = 12$ пікселей і кутом нахилу стрілки відносно лінії сили в 10° (рис.3), координати правої x_{sp}, y_{sp} і лівої x_{sl}, y_{sl} кінців стрілок відносно точки $O(x_m, y_m)$ приймають значення:

$$\begin{aligned} x_{sp} &:= x - \text{round}(12 \cdot \sin((80 - a) \cdot \pi / 180)); \\ y_{sp} &:= y - \text{round}(12 \cdot \cos((80 - a) \cdot \pi / 180)); \\ x_{sl} &:= x - \text{round}(12 \cdot \sin((100 - a) \cdot \pi / 180)); \\ y_{sl} &:= y - \text{round}(12 \cdot \cos((100 - a) \cdot \pi / 180)); \end{aligned} \quad (18)$$

де:

$a = \alpha$ – кут нахилу вектора сили $\vec{F}$.

Рис. 3. Координати стрілок напрямку вектора сили $\bar{F}$ Fig. 3. Coordinates of arrows of a force $\bar{F}$ vector direction

Тоді права і ліва стрілки напрямку вектора сили відображаються відповідними процедурами:

$$Line(x_m + x, y_m - y, x_m + x_{sp}, y_m - y_{sp}); \quad (19)$$

$$Line(x_m + x, y_m - y, x_m + x_{sl}, y_m - y_{sl});$$

Для наочності зображення кута α нахилу сили введемо в перелік змінних програми параметр приведенного кута α_p , що може приймати значення від 0° до 90° в кожному із чотирьох квадрантів розташування вектора сили (рис. 4) Застосуємо умовний оператор до умов зміни α_p в залежності від значень α (рис. 4):

$$\begin{aligned} & \text{If} \quad <\text{умова значення } a> \\ & \text{then} \quad <\text{присвоєння значення } a_{pi}> \\ & \text{else} \quad <\text{присвоєння значення } a_{pi+1}>, \end{aligned} \quad (20)$$

де:

If – якщо,
then – тоді,
else – інакше - зарезервовані слова, знак розділу між якими у вигляді коми з крапкою “;” не ставиться.

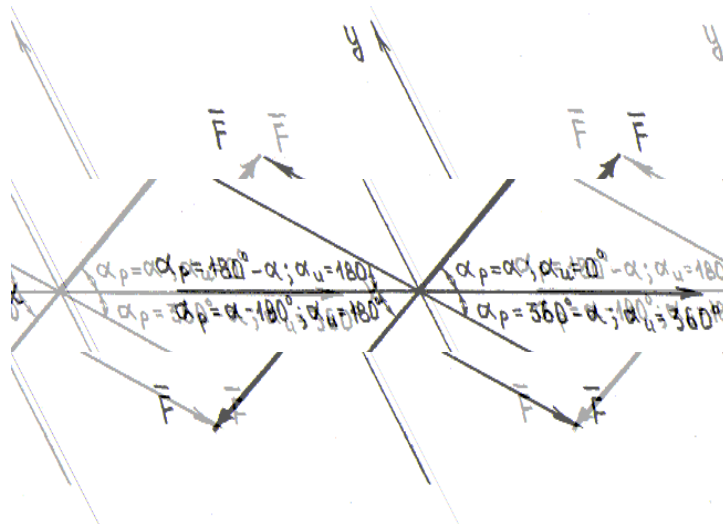


Рис. 4. Значення кутів α_p і α_u в залежності від значень заданого кута α в різних квадрантах

Fig. 4. Values of corners α_p and α_u depending on values of the set corner α in different quadrants

Тоді можемо вивести на екран значення кута нахилу сили за допомогою процедури рядка:

$$Str(a_{pi} : W_1, S); \quad (21)$$

і процедури виводу значень рядка

$$Outtextxy(x_m, y_m, S); \quad (22)$$

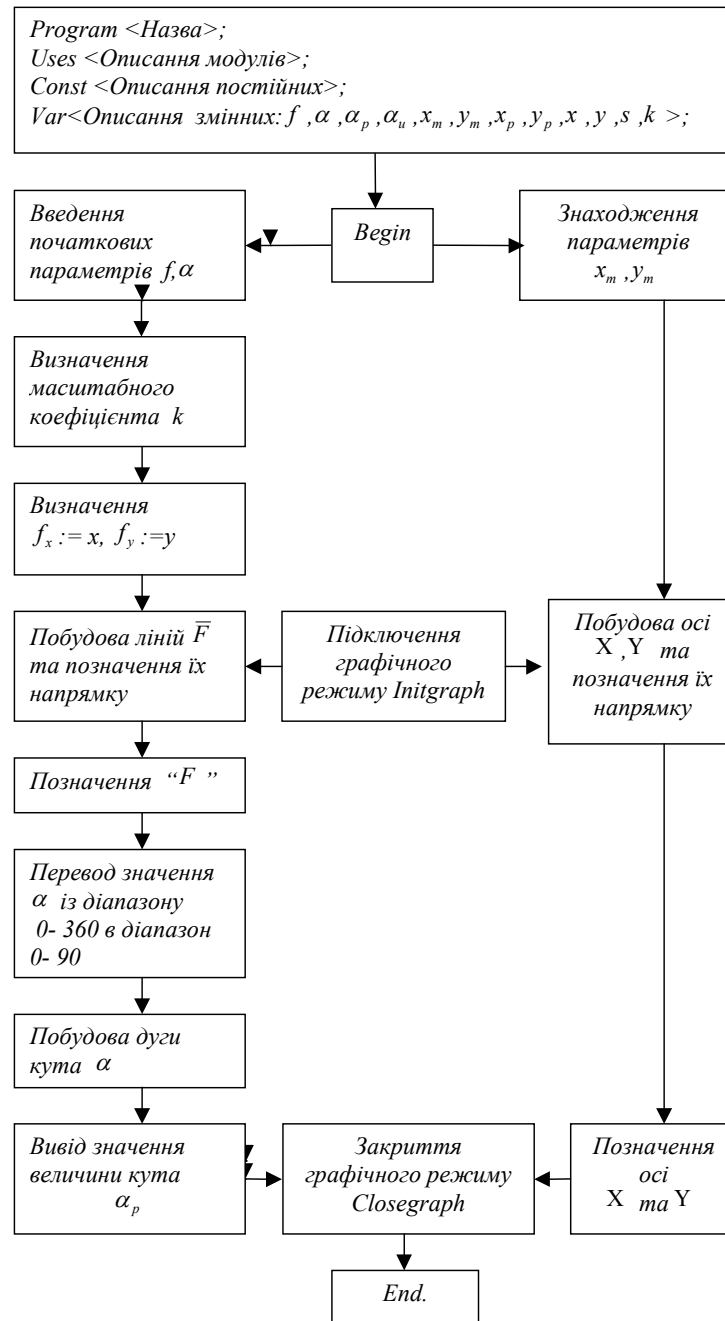
де:

(21) – процедура, що перетворює числове значення типу *real* або *integer* в рядок символів *S*;

S:string – змінна рядкового типу, що задається в розділі описання на початку програми;

a_{pi} – значення приведенного кута α_p ;

W_1 – формат перетворення, що визначає загальну ширину поля, виділеного під відповідне символічне представлення заданого числа .

Рис. 5. Блок-схема програми зображення вектора сили $\vec{F}$ на площиніFig. 5. A flow diagram of the force $\vec{F}$ vector map on a plane

Аналогічно виводимо на екран значення f модуля сили:

$$\begin{aligned} &Str(f: 6: 1, S); \\ &Outtextxy(x_m, y_m, S); \end{aligned} \quad (23)$$

Дуга кута α , (рис. 4) визначається процедурою з форматом звернення

$$Arc(x, y, <початковий\ кут>, <кінцевий\ кут>, <радіус>); \quad (24)$$

де:

x, y – координати центру дуги; $<початковий\ кут>$,
 $<кінцевий\ кут>$ – відповідно параметри кутів дуги в градусах, відлік яких
 відбувається проти годинникової стрілки;
 $<радіус>$ – радіус дуги в пікселях.

Введемо також додаткову допоміжну змінну кута α_u за умов її граничних значень 0° , 180° , 360° в залежності від значень кута α (рис. 4).

З урахуванням значень α_u і α для нашого випадку (24+++) має наступні умови застосування:

$$\begin{aligned} &\text{If } \alpha > \alpha_u \text{ then} \\ &Arc(x_m, y_m, \alpha_u, \alpha) \text{ else} \\ &Arc(x_m, y_m, \alpha, \alpha_u); \end{aligned} \quad (25)$$

Робота адаптеру в графічному режимі буде завершена процедурою *Closegraph*, як і робота всієї програми в цілому - службовим словом *End*.

Блок-схема програми наведена на рис. 5.

ВИСНОВКИ

1. На основі програмної мови високого рівня Turbo Pascal 7.0 вирішена задача методики графічного представлення вектора сили $\vec{F}$ на площині.
2. Визначені вхідні параметри, що можуть повністю описати заданий вектор сили будь-якого модуля і напрямку на графічній площині.
3. Запропонована методика алгоритму програми може бути використана для наочної демонстрації поняття сили як вектора, а також як початковий складовий елемент для інших графічних задач механіки.

ЛІТЕРАТУРА

- Брябрин В.М., 1990: Программное обеспечение персональных ЭВМ. М.: Наука, 272.
Леонтьев В.П. 2001: Новейшая энциклопедия компьютера. М.: Олма-Пресс, 847.
Павловський М.А., 2002: Теоретична механіка. К.: Техніка, 510.
Марченко А.И., Марченко Л.А. 1997: Borland Pascal 7.0. К.: Юниор, 495.
Фараонов В.В. 1997: Turbo Pascal 7.0. Практика программирования. М.: Нолидж, 432.
Хершель Р., 1991: Turbo Pascal. М.: Квадрат, 342.

SOFTWARE OF A PERSONAL COMPUTER FOR GRAPHIC REPRESENTATION
OF THE SET FORCE VECTOR ON A PLANE

Summary. The algorithm of graphic representation of the set force vector in a plane is offered at usage of personal computer software.

Key words: algorithm, graphics representation, force vector

Reviewer: Volodymyr Bulgakov, Prof. D. Sc. Eng.