

## Додатки

Додаток А. [Основні клавіші та комбінації клавіш інтегрованого середовища Turbo Pascal](#)

Додаток Б. [Коди помилок під час компіляції програми](#)

Додаток В. [Коди помилок під час виконання програми](#)

Додаток Г. [Теми рефератів](#)

Додаток Д. [Завдання до практичної частини курсової роботи](#)

Додаток Е. [Допоміжні модулі](#)

## Додаток А. Основні клавіші та комбінації клавіш інтегрованого середовища Turbo Pascal

Таблиця А.1. Клавіші команд і управління вікнами

Клавіша	Команда меню	Дія
Enter		Підтвердження виконання команди; Перехід до нового рядка
Esc		Відміна від команди
F2	File / Save	Записати файл
F3	File / Open	Відкрити файл
Alt+F3	Window / Close	Закрити файл
Alt+цифра		Перейти у вікно з номером ...
Alt+F5	Window / User screen	Вивести екран програми
F6	Window / Next	Переключитись у наступне вікно
Ctrl+Break		Переривання виконання програми
Alt+X	File / Exit	Завершити роботу з Turbo Pascal

Таблиця А.2. Клавіші для редагування програм

Клавіша	Команда меню	Дія
Ctrl+Home		Перейти у початок екрану
Ctrl+End		Перейти у кінець екрану
Ctrl+PgUp		Перейти у початок файла
Ctrl+PgDn		Перейти у кінець файла
Ctrl+Y		Видалити рядок

Ctrl+N		Вставити рядок
Shift+клавіші управління курсором		Виділення фрагменту
Ctrl+Del	Edit/Clear	Видалення виділеного фрагменту
Ctrl+Ins	Edit/Copy	Копіювання виділеного фрагменту у буфер
Shift+Del	Edit/Cut	Переміщення виділеного фрагменту у буфер
Shift+Ins	Edit/Paste	Вставка фрагменту із буфера у програму
Ctrl+K, B		Помітити початок фрагменту
Ctrl+K, K		Помітити кінець фрагменту
Ctrl+K, C		Скопіювати фрагмент
Ctrl+K, V		Перемістити фрагмент
Ctrl+K, Y		Видалити фрагмент
Ctrl+K, H		Зняти виділення фрагменту

Таблиця А.3. Клавіші для компіляції і налагодження програм

Клавіша	Команда меню	Дія
Alt+F9	Compile / Compile	Компіляція активного файлу
F9	Compile / Make	Компіляція програми і модулів, які вимагають перекомпіляції
Ctrl+F9	Run / Run	Виконання програми повністю
F4	Run / Go to cursor	Виконання програми до позиції курсору
F7	Run / Trace into	Виконання одного рядка програми із входженням у процедури та функції
F8	Run / Step over	Виконання одного рядка програми без входження у процедури та функції
Ctrl+F2	Run / Program reset	Переривання налагодження програми і повернення у початковий стан
Ctrl+F4	Debug / Evaluate	Обчислити вираз або змінити значення елемента програми
Ctrl+F7	Debug / Add watch	Відобразити вікно перегляду значень елементів програми або додати до нього елемент

Таблиця А.4. Клавіші довідкової служби

Клавіша	Команда меню	Дія
F1	Help / Contents	Виклик контекстної допомоги
Shift+F1	Help / Index	Виклик індексної допомоги
Alt+F1	Help / Previous topic	Повернутись у попередній екран довідки
Ctrl+F1	Help / Topic search	Виклик контекстної допомоги по мові Pascal

## Додаток Б. Коди помилок під час компіляції програми

Код помилки і повідомлення	Причини помилки
1 Out of memory	Недостатньо пам'яті Є кілька можливих рішень проблеми:

	<ul style="list-style-type: none"> <li>• Якщо опція Compile / Destination встановлена у Memory, замініть її на Disk;</li> <li>• Якщо опція Options / Linker / Linker buffer встановлена у Memory, замініть її на Disk;</li> <li>• Розбийте програму на кілька модулів або спробуйте відкомпілювати програму у захищеному а не реальному режимі.</li> </ul>
2 Identifier expected	<b>Очікується ідентифікатор</b> Можливо ви намагаєтесь перевизначити стандартний ідентифікатор.
3 Unknown identifier	<b>Невідомий ідентифікатор</b> Ідентифікатор не описаний або не видимий у поточній області.
4 Duplicate identifier	<b>Дублювання ідентифікатора</b> Ідентифікатор з таким ім'ям вже використовується для елемента програми, змінної тощо, описаного у поточному блоці.
5 Syntax error	<b>Синтаксична помилка</b> У тексті програми знайдено неприпустимий символ. Можливо ви не виділили апострофом рядкову константу.
6 Error in real constant	<b>Помилка у константі дійсного типу</b>
7 Error in integer constant	<b>Помилка у константі цілочислового типу</b>
8 String constant exceeds line	<b>Рядкова константа виходить за межі рядка</b> Мабуть ви забули закрити апостроф у рядковій константі або розірвали рядкову константу на два рядки.
10 Unexpected end of file	<b>Неочікуваний кінець файлу</b> Серед можливих причин такого повідомлення: <ul style="list-style-type: none"> <li>• Вихідний файл завершується перед останнім end головної операторної частини. Скоріше за все кількість операторів begin і end незбалансована;</li> <li>• Включений файл завершується всередині операторної частини. Кожна операторна частина повинна повністю міститися в одному файлі;</li> <li>• Не закритий коментар.</li> </ul>
11 Line too long	<b>Занадто довгий рядок</b> Довжина рядка не повинна перевищувати 127 символів.
12 Type identifier expected	<b>Очікується тип ідентифікатора</b> Ви описуєте новий ідентифікатор і не вказуєте його тип.
13 Too many open files	<b>Занадто багато відкрито файлів</b> У такому випадку або у файлі config.sys відсутній рядок files=xx, або ж за допомогою цього рядка визначено мало відкритих файлів. Необхідно збільшити їхню кількість.
14 Invalid file name	<b>Невірне ім'я файлу</b> Помилка у записі імені файлу або у шляху до нього.
15 File not found	<b>Файл не знайдений</b> Файл не знайдений у жодному з каталогів пошуку, визначених для даного типу файлів.
16 Disk full	<b>Диск заповнений</b> На диску (найчастіше на дискеті) немає вільного місця. Видаліть кілька файлів або вставте новий диск.
17 Invalid compiler directive	<b>Невірна директива компілятора</b> Невірний символ використано у якості директиви компілятора або невірним є один з параметрів компіляції або використовується глобальна директива компілятора, коли компіляція тіла програми вже розпочалася.
18 Too many files	<b>Занадто багато файлів</b> До компіляції програми або модулю включено занадто багато файлів. Спробуйте зменшити їхню кількість, наприклад шляхом об'єднання файлів, що підключаються. Можна спробувати перемістити усі файли, що підключаються, в один (поточний) каталог і скоротити їхні імена.
19 Undefined type	<b>Невизначений тип у визначенні покажчика</b>

in pointer def	Тип, що використовується при визначенні типу покажчика, ніде не описаний.
20 Variable identifier expected	<b>Очікується ідентифікатор змінної</b> Ідентифікатор не визначає змінну, хоча повинен це робити.
21 Error in type	<b>Помилка у визначенні типу</b> З такого символу не може розпочинатися визначення типу.
22 Structure too large	<b>Занадто велика структура</b> Розмір структурованого типу не може перевищувати 65 535 байт.
23 Set base type out of range	<b>Базовий тип множини виходить за межі припустимого діапазону</b> Базовий тип множини повинен бути піддіапазоном, що знаходиться всередині діапазону 0..255 або перелічуваним типом, який має не більше ніж 256 можливих значень.
24 File components may not be files or objects	<b>Компонентами файла не можуть бути файли або об'єкти</b> Файловими компонентами не можуть бути файли або об'єкти, а також структуровані типи, які включають типи об'єктів або файлів.
25 Invalid string length	<b>Невірна довжина рядка</b> При визначенні довжина рядкового типу повинна бути у діапазоні 1..255.
26 Type mismatch	<b>Несумісність типів</b> Причиною несумісності типів можуть бути: <ul style="list-style-type: none"> <li>• Несумісність типів змінної і виразу в операторі присвоєння;</li> <li>• Несумісність типів фактичного і формального параметрів у процедурах або функціях;</li> <li>• Несумісність типів виразу та індексу масиву;</li> <li>• Несумісність типів операндів у виразі.</li> </ul>
27 Invalid subrange base type	<b>Невірний піддіапазоновий базовий тип</b> Усі порядкові типи є припустимими базовими типами.
28 Lower bound > than upper bound	<b>Нижня межа більша за верхню</b> При визначенні піддіапазону значення нижньої межі є більшим за верхню.
29 Ordinal type expected	<b>Очікується порядковий тип</b> У даному випадку дійсний, рядковий, структурований і покажчиковий типи є неприпустимими.
30 Integer constant expected	<b>Очікується цілочислова константа</b>
31 Constant expected	<b>Очікується константа</b>
32 Integer or real constant expected	<b>Очікується цілочислова або дійсна константа</b>
33 Pointer Type identifier expected	<b>Очікується ідентифікатор типу</b>
34 Invalid function result type	<b>Неприпустимий тип результату функції</b> Припустимими типами результатів функцій є усі прості типи, рядкові типи і покажчики.
35 Label identifier expected	<b>Очікується ідентифікатор мітки</b>
36 BEGIN expected	<b>Очікується оператор begin</b> Відсутній оператор begin або є помилка у блоковій структурі програми або модулю.
37 END expected	<b>Очікується оператор end</b> Відсутній оператор end або є помилка у блоковій структурі програми або модулю.
38 Integer expression expected	<b>Очікується цілочисловий вираз</b> Попередній вираз повинен мати цілочисловий тип
39 Ordinal expression expected	<b>Очікується вираз порядкового типу</b> Попередній вираз повинен мати порядковий тип
40 Boolean expression expected	<b>Очікується вираз логічного типу</b> Попередній вираз повинен мати логічний тип
41 Operand types do	<b>Типи операндів не відповідають оператору</b>

not match	До таких операндів не можна застосовувати вказану операцію, наприклад, не можна множити рядкові дані.
42 Error in expression	<b>Помилка у виразі</b> Такий символ не може бути присутнім у виразі у даному місці. Можливо ви забули записати оператор між двома операндами.
43 Illegal assignment	<b>Невірне присвоєння</b> <ul style="list-style-type: none"> <li>• Файли і нетипізовані змінні не можуть бути значеннями, що присвоюються;</li> <li>• Ідентифікатору функції можна присвоювати значення тільки всередині операторної частини функції.</li> </ul>
44 Field identifier expected	<b>Очікується ідентифікатор поля</b> Знайдений ідентифікатор не визначає поле у відповідній змінній типу запис або об'єкт.
45 Object file too large	<b>Об'єктний файл є занадто великим</b> Turbo Pascal не може лінувати у *.obj-файл, розмір якого перевищує 64К.
46 Undefined EXTERN	<b>Невизначений зовнішній елемент</b> Зовнішня процедура або функція не має відповідного public-визначення в об'єктному файлі. Переконайтеся, що усі об'єктні файли задані у директивах {\$L ім'я файла} і перевірте написання ідентифікатора процедури або функції у *.asm-файлі.
47 Invalid object file record	<b>Невірний запис у об'єктному файлі</b> Перевірте, чи є файл *.obj-файлом
48 Code segment too large	<b>Сегмент коду є занадто великим</b> Максимальний розмір коду програми або модулю 65 520 байт. Якщо компілюється програма, треба перенести кілька процедур і функцій у модуль. якщо компілюється модуль, його слід розбити на кілька менших модулів.
49 Data segment too large	<b>Сегмент даних є занадто великим</b> Максимальний розмір сегменту даних програми 65 520 байт, включаючи дані, що визначені у модулях, які використовуються. Якщо необхідно більше глобальних даних, потрібно визначити структури невеликого розміру як покажчики і динамічно їх розмістити із застосуванням процедури New.
50 DO expected	<b>Очікується зарезервоване слово DO</b>
51 Invalid PUBLIC definition	<b>Невірне визначення PUBLIC</b> <ul style="list-style-type: none"> <li>• Дві або більше директиви PUBLIC у асемблері визначають один і той самий ідентифікатор;</li> <li>• *.obj-файл визначає PUBLIC-символи, які не знаходяться у сегменті CODE.</li> </ul>
52 Invalid EXTRN definition	<b>Невірне визначення EXTERN</b> <ul style="list-style-type: none"> <li>• На ідентифікатор було посилання за допомогою директиви EXTRN у асемблері, однак він не був визначений а ні у програмі, а ні у модулі, а ні у інтерфейсній частині модулів, які використовуються;</li> <li>• Цей ідентифікатор позначає абсолютну змінну;</li> <li>• Цей ідентифікатор позначає inline-процедуру або функцію.</li> </ul>
53 Too many EXTRN definitions	<b>Занадто багато визначень EXTERN</b> Turbo Pascal не може обробляти *.obj-файли з кількістю оголошень EXTERN понад 256.
54 OF expected	<b>Очікується зарезервоване слово OF</b>
55 INTERFACE expected	<b>Очікується зарезервоване слово INTERFACE</b>
56 Invalid relocatable reference	<b>Невірне перемішуване послання</b>

	<ul style="list-style-type: none"> <li>• *.obj-файл містить дані і переміщувані посилання у сегментах, що відрізняються від CODE. Наприклад, ви намагаєтесь визначити ініційовані змінні у сегменті DATA;</li> <li>• *.obj-файл містить посилання розміром у байт на переміщувані символи. Ця помилка трапляється, якщо використовуються оператори HIGH і LOW із переміщуваними символами, або якщо є посилання на переміщувані символи у директивах DB;</li> <li>• Операнд посилається на переміщуваний символ, який не був визначений а ні у сегменті CODE, а ні у сегменті DATA;</li> <li>• Операнд посилається на процедуру або функцію EXTERN зі зміщенням, наприклад, CALL SortProc+8.</li> </ul>
57 THEN expected	<b>Очікується зарезервоване слово THEN</b>
58 TO or DOWNTO expected	<b>Очікується зарезервоване слово TO або DOWNTO</b>
59 Undefined forward	<p><b>Не визначений елемент, який був вказаний раніше</b></p> <ul style="list-style-type: none"> <li>• Процедура або функція була оголошена у інтерфейсній частині модуля, але її визначення не зустрічається ніде у секції реалізації;</li> <li>• Ця процедура або функція була оголошена з ключовим словом forward, однак її визначення ніде не знайдено.</li> </ul>
61 Invalid typecast	<p><b>Невірне приведення типів</b></p> <ul style="list-style-type: none"> <li>• Розміри посилання на змінну і тип призначення відрізняються при приведенні типу змінної;</li> <li>• Спроба приведення типів у виразі, де припустимим є лише посилання на змінну.</li> </ul>
62 Division by zero	<p><b>Ділення на нуль</b> У попередньому операнді була спроба ділення на нуль.</p>
63 Invalid file type	<p><b>Невірний тип фала</b> Цей тип файла не підтримується процедурою обробки файлів, наприклад, readln із типізованим файлом або seek із текстовим файлом.</p>
64 Cannot read or write variables of this type	<p><b>Неможна зчитувати або виводити змінні такого типу</b> Процедури read і readln можуть зчитувати значення типу char, integer, real і string. Процедури write або writeln можуть виводити значення типів char, integer, real, string і boolean.</p>
65 Pointer variable expected	<p><b>Очікується змінна типу покажчик</b> Попередня змінна повинна мати тип покажчик.</p>
66 String variable expected	<p><b>Очікується змінна рядкового типу</b> Попередня змінна повинна мати рядковий тип.</p>
67 String expression expected	<p><b>Очікується вираз рядкового типу</b> Попередній вираз повинен мати рядковий тип.</p>
68 Circular unit reference	<p><b>Циклічне посилання на модуль</b> Не дозволяється, щоб два модулі використовували один одного в операторах uses інтерфейсної частини. Циклічні посилання припустимі лише у блоці реалізації.</p>
69 Unit name mismatch	<p><b>Невідповідність імені модуля</b> Ім'я модуля, знайденого у файлі *.tpu, *.tpp або *.tpw не відповідає імені, яке вказане у розділі uses.</p>
70 Unit version mismatch	<p><b>Невідповідність версії модуля</b> Один або більше модулів, які використовуються даним модулем, були змінені після компіляції. Для автоматичної компіляції модулів, яким потрібна перекомпіляція, слід використовувати команди меню Compile/Make або Compile/Build.</p>
71 Internal stack overflow	<p><b>Переповнення внутрішнього стеку</b> Внутрішній стек компілятора вичерпав свої можливості через велику кількість вкладених операторів. Переробіть вихідний код програми таким чином, щоб він не мав такого ступеня вкладеності. Наприклад,</p>

	перетворить внутрішні рівні вкладених операторів у окрему процедуру.
72 Unit file format error	<b>Помилка у форматі файлу модуля</b> Файл *.tpu (*.tpp, *.tpw) є невірним. Можливо, що цей файл був створений старою версією компілятора. У такому випадку перекомпілюйте модуль.
73 Implementation expected	<b>Очікується зарезервоване слово IMPLEMENTATION</b> Можливо до інтерфейсної частини ви намагаєтесь включити реалізацію процедури або функції.
74 Constant and case types don't match	<b>Константа і тип перемикача в операторі case не відповідають один одному</b> Тип case-константи є несумісним із виразом перемикача.
75 Record or object variable expected	<b>Очікується змінна типу record або object</b> Попередня змінна повинна мати тип запису або об'єкту
76 Constant out of range	<b>Константа виходить за межі діапазону</b> Була спроба: <ul style="list-style-type: none"> <li>• Індексувати масив за допомогою константи, що виходить за межі припустимого діапазону;</li> <li>• Присвоїти змінній значення константи, яке виходить за припустимі межі діапазону змінної;</li> <li>• Передати константу, яка виходить за межі припустимого діапазону, як фактичний параметр процедури або функції.</li> </ul>
77 File variable expected	<b>Очікується змінна типу file</b> Попередня змінна повинна мати тип file
78 Pointer expression expected	<b>Очікується вираз типу pointer</b>
79 Integer or real expression expected	<b>Очікується вираз типу integer або real</b>
80 Label not within current block	<b>Мітка знаходиться ззовні поточного блоку</b> Оператор goto не повинен посилатись на мітку, що знаходиться поза межами поточного блоку.
81 Label already defined	<b>Мітка вже визначена</b> Мітка із таким ім'ям вже помічає інший оператор
82 Undefined label in preceding stmt part	<b>Невизначена мітка у попередній операторній частині</b> Ця мітка вже була описана і на неї є посилання у попередній операторній частині, але ця мітка ніде не визначена.
83 Invalid @ argument	<b>Невірний аргумент для @</b> Припустимими аргументами для @ є посилання на змінні і ідентифікатори процедур або функцій.
84 UNIT expected	<b>Очікується зарезервоване слово UNIT</b>
85 ";" expected	<b>Очікується символ ";"</b> Ймовірно, що наприкінці попереднього рядка ви забули завершити оператор крапкою з комою.
86 ":" expected	<b>Очікується символ ":"</b>
87 ", " expected	<b>Очікується символ ", "</b> Скоріше за все кількість параметрів, які передаються у процедуру або функцію є меншою за оголошену.
88 "(" expected	<b>Очікується символ "("</b> Ймовірно, що ви викликаєте процедуру або функцію у яку слід передавати параметри.
89 ")" expected	<b>Очікується символ ")"</b> Напевне, що у процедурі або функції кількість параметрів є меншою, ніж ви намагаєтесь передати.
90 "=" expected	<b>Очікується символ "="</b>
91 ":=" expected	<b>Очікується символ ":="</b>
92 "[" or "(" expected	<b>Очікується символ "[" або "("</b> Напевне, що ви використовуєте масив і не вказуєте його індекс або некоректно присвоюєте значення множинного типу.
93 "]" or ")" expected	<b>Очікується символ "]" або ")"</b>

94 "." expected	<b>Очікується символ "."</b> Перевірте щоб тип не використовувався як змінна, і щоб ім'я програми не перекривалося ідентифікаторами із інших модулів.
95 ".." expected	<b>Очікується символ ".."</b>
96 Too many variables	<b>Занадто багато змінних</b> <ul style="list-style-type: none"> <li>• Загальний розмір глобальних змінних, визначених всередині програми або модуля не може перевищувати 64К;</li> <li>• Загальний розмір локальних змінних, визначених всередині процедури або функції не може перевищувати 64К;</li> </ul>
97 Invalid FOR control variable	<b>Невірна керуюча змінна в операторі FOR</b> Керуюча змінна в операторі FOR повинна бути простою змінною і визначатись у розділі описування поточної програми.
98 Integer variable expected	<b>Очікується змінна цілочислового типу</b> Попередня змінна повинна бути цілочислового типу.
99 Files types are not allowed here	<b>Файловий і процедурний типи тут неприпустимі</b> Типізована константа не може бути файлового типу
100 String length mismatch	<b>Невідповідність довжини рядка</b> Довжина рядкової константи не відповідає кількості компонент у символьному масиві.
101 Invalid ordering of fields	<b>Невірний порядок полів</b> Поля констант типу record і object повинні бути записаними у порядку визначення
102 String constant expected	<b>Очікується константа рядкового типу</b>
103 Integer or real variable expected	<b>Очікується змінна типу integer або real</b>
104 Ordinal variable expected	<b>Очікується змінна порядкового типу</b>
105 INLINE error	<b>Помилка у INLINE</b> Оператор ">" неприпустимий разом із переміщуваними посиланнями на змінні. Такі посилання завжди мають розмір, що ділиться без остачі на слово.
106 Character expression expected	<b>Очікується вираз символьного типу</b>
107 Too many relocation items	<b>Занадто багато переміщуваних елементів</b>
108 Overflow in arithmetic operation	<b>Переповнення у арифметичній операції</b> Результат попередньої арифметичної операції виходить за межі діапазону LongInt.
109 No enclosing For, While or Repeat statement	<b>Зовні оператора For, While або Repeat</b> Стандартні процедури Break і Continue не можуть використовуватись поза межами операторів For, While або Repeat
112 CASE constant out of range	<b>Константа оператора case виходить за межі припустимого діапазону</b>
113 Error in statement	<b>Помилка в операторі</b> Оператор не може починатися з цього символу
114 Cannot call an interrupt procedure	<b>Неможна безпосередньо викликати процедуру переривання</b>
116 Must be in 8087 mode to compile	<b>Компіляція можлива лише у режимі 8087</b> Така інструкція може компілюватись лише з директивою {\$N+}. Операція для 80x87 над типами Single, Double, Extended і Comp є неприпустимою у стані {\$N-}.
117 Target address not found	<b>Адреса, що відшукується, не знайдена</b> Команда головного меню Search / Find error не може визначити оператор, який відповідає вказаній адресі.
118 Include files are not allowed here	<b>Файли, що підключаються, тут неприпустимі</b> Кожна операторна частина повинна міститись у одному файлі.
119 No inherited	<b>Методи inherited тут недоступні</b>



methods are accessible here	Ключове слово <code>inherited</code> використовується поза методом об'єкту, який не має предка.
121 Invalid qualifier	<b>Невірний кваліфікатор</b> Була спроба: <ul style="list-style-type: none"> <li>• Індексувати змінну, яка не є масивом;</li> <li>• Визначити поля у змінній, яка не є записом;</li> <li>• Отримати значення, на яке вказує змінна, яка не є покажчиком.</li> </ul>
122 Invalid variable reference	<b>Невірне посилання на змінну</b> Попередня конструкція відповідає синтаксису посилання на змінну, але не позначає розміщення пам'яті. Скоріше за все, ви намагались модифікувати параметр-константу або звернулися до покажчика на функцію, не отримавши значення, на яке він посилається.
123 Too many symbols	<b>Занадто багато символів</b> Програма або модуль описує понад 64К символів. Якщо програма або модуль компілюються з опцією <code>{SD+}</code> , тоді спробуйте її відключити. В іншому випадку можна спробувати перемістити кілька оголошень у окремий модуль.
124 Statement part too large	<b>Операторна частина занадто велика</b> Розмір операторної частини не повинен перевищувати 24К. Якщо зустрічається така помилка, перемістіть розділи операторної частини до однієї або кількох процедур. У такому випадку обов'язково слід уточнити структуру вашої програми.
126 Files must be var parameters	<b>Файли повинні бути параметрами-змінними</b>
127 Too many conditional symbols	<b>Занадто багато умовних символів</b> Для визначення додаткових умовних символів недостатньо місця. Спробуйте виключити кілька символів або скоротити деякі символічні імена.
128 Misplaced conditional directive	<b>Невірно розташована директива умовної компіляції</b> Компілятор зустрів директиву <code>{ELSE}</code> або <code>{ENDIF}</code> без відповідної директиви <code>{IFDEF}</code> , <code>{IFNDEF}</code> або <code>{IFOPT}</code> .
129 ENDIF directive missing	<b>Відсутня директива EDNIF</b> Вихідний файл завершується всередині конструкції умовної компіляції. Кількість директив <code>{IFxxx}</code> і <code>{ENDIF}</code> повинна бути однаковою.
130 Error in initial conditional defines	<b>Помилка у початкових умовних визначеннях</b> Початкові умовні символи, які визначені у <code>Options / Compiler / Conditions defines</code> . Turbo Pascal вважає, що зустрілись кілька ідентифікаторів, розділених пробілами, комами або крапками із комами.
131 Header does not match previous definition	<b>Заголовок не відповідає попередньому визначенню</b> Заголовок процедури або функції, заданий у інтерфейсній частині або у оголошенні <code>forward</code> не відповідає даному заголовку.
133 Cannot evaluate this expression	<b>Цей вираз виражувати неможливо</b> Була спроба використати недопустимий у Pascal синтаксис у константному виразі. Можлива помилка у пріоритеті виконання операцій.
134 Expression incorrectly terminated	<b>Вираз завершується ненормально</b> Тільки для вбудованого налагоджувача. Turbo Pascal очікує або оператор, або кінець виразу у цій точці, але не знаходить ні того, ні іншого.
135 Invalid format specifier	<b>Невірний специфікатор формату</b> Тільки для вбудованого налагоджувача. Використовується невірна специфікація формату або числовий аргумент специфікації формату виходить за межі припустимого діапазону.
136 Invalid indirect reference	<b>Невірне непряме посилання</b> Оператор намагається виконати невірне непряме посилання. Наприклад, використовується <code>absolute</code> -змінна, у якої базова змінна невідома у поточному модулі, або використовується <code>inline</code> -модуль, який посилається на змінну, невідому у поточному модулі.
137 Structured variables are not allowed here	<b>Структурні змінні тут неприпустимі</b> Спроба виконати операцію над структурною змінною, яка не підтримується мовою Pascal. Наприклад, спроба розділити два записи.

138 Cannot evaluate without System unit	<b>Неможна обчислити без модуля System</b> Тільки для вбудованого налагоджувача. Бібліотека Turbo.tpl повинна містити модуль System для того, щоб налагоджувач міг обчислити вираз.
139 Cannot access this symbol	<b>Доступ до цього символу неможливий</b> Тільки для вбудованого налагоджувача. Уся множина символів у програмі доступна одразу після компіляції програми. Однак до деяких символів, таких як змінні, доступ неможливий до тих пір, поки програма не розпочне фактично виконуватись.
140 Invalid floating-point operation	<b>Невірна операція із плаваючою точкою</b> Операція над двома дійсними значеннями призвела до переповнення або до ділення на нуль.
141 Cannot compile overlays to memory	<b>Неможна компілювати оверлей у пам'ять</b> Для реального режиму. Програма, яка використовує оверлей, повинна компілюватись на диск.
142 Pointer or procedural variable expected	<b>Очікується змінна типу procedure або pointer</b> Стандартна функція Assigned вимагає, щоб аргумент був змінною типу procedure або pointer.
143 Invalid procedure or function reference	<b>Невірне посилання на процедуру або функцію</b> <ul style="list-style-type: none"><li>• Ви намагаєтесь викликати процедуру у виразі;</li><li>• Якщо процедура або функція повинна бути присвоєна змінній типу procedure, то її треба компілювати із директивою {\$F+} і неможна оголошувати з ключовими словами inline або interrupt.</li></ul>
144 Cannot overlay this unit	<b>Неможливо зробити цей модуль оверлейним</b> Для реального режиму. Була спроба створити оверлейний модуль, який не був відкомпільований у стані {\$O+}.
145 Too many nested scopes	<b>Занадто багато вкладених областей</b>
146 File access denied	<b>Відмова у доступі до файла</b> Файл не може бути відкритий або створений. Скоріше за все, компілятор намагається записувати інформацію у файл, призначений тільки для зчитування.
147 object type expected	<b>Очікується об'єктовий тип</b>
148 Local object types are not allowed	<b>Локальний об'єктний тип є неприпустимим</b> Об'єктний тип може бути визначений тільки в зовнішній сфері дії програми або модуля. Визначення об'єктного типу всередині процедури або функції є неприпустимим.
149 Virtual expected	<b>Очікується зарезервоване слово virtual</b>
150 Method identifier expected	<b>Очікується ідентифікатор методу</b>
151 Virtual constructors are not allowed	<b>Конструктори не можуть бути віртуальними</b>
152 Constructor identifier expected	<b>Очікується ідентифікатор конструктора</b>
153 Destructor identifier expected	<b>Очікується ідентифікатор деструктора</b>
154 Fail only allowed within constructors	<b>Fail припустимий лише всередині конструктора</b>
155 Invalid combination of opcode and operands	<b>Невірна комбінація коду операції і операндів</b> Асемблерний код операції не припускає такої комбінації операндів. Можливими причинами є: <ul style="list-style-type: none"><li>• Занадто багато або занадто мало операндів для цього коду операцій, наприклад, INC AX, BX або MOV AX;</li></ul>

	<ul style="list-style-type: none"> <li>• Вірна кількість операндів, але їх тип і порядок не відповідають коду операції, наприклад, <code>DEC 1, MOV AX, CL</code> або <code>MOV 1, AX</code>;</li> <li>• Є коментарі всередині операторів асемблера, наприклад, <code>MOV {Init value} AX, 1</code>.</li> </ul>
156 Memory reference expected	<b>Очікується посилання</b> Даний операнд асемблера не є посиланням, яке потрібно у цьому місці. Ймовірно ви забули поставити [ ] навколо регістрового операнда. Наприклад, <code>MOV AX, BX+SI</code> замість <code>MOV AX, [BX+SI]</code> .
157 Cannot add or subtract relocatable symbols	<b>Неможна додавати або віднімати переміщувані символи</b> Над переміщуваними символами можна виконувати тільки додавання і віднімання констант. Змінні, процедури, функції і мітки є переміщуваними символами. Якщо <code>var</code> - змінна, а <code>const</code> - константа, тоді інструкції <code>MOV AX, const+const</code> і <code>MOV AX, var+const</code> є вірними, а <code>MOV AX, var+var</code> - ні.
158 Invalid register combination	<b>Невірна комбінація регістрів</b> Правильні індексні регістрові комбінації: <code>[BX]</code> , <code>[BP]</code> , <code>[SI]</code> , <code>[DI]</code> , <code>[BX+SI]</code> , <code>[BX+DI]</code> , <code>[BP+SI]</code> , <code>[BP+DI]</code> . Інші індексні регістрові комбінації заборонені.
159 286/287 instructions are not enabled	<b>Інструкції 286/287 недоступні</b> Використовуйте директиву компілятора <code>{ \$G+ }</code> , для того, щоб дозволити коди операцій 286/287.
160 Invalid symbol reference	<b>Невірне посилання на символ</b> До цього символу неможна звертатись у асемблерному операнді. Можливими причинами є: <ul style="list-style-type: none"> <li>• Ви намагаєтесь звернутись у асемблерному операнді до стандартної процедури, стандартної функції або до спеціальних масивів <code>Mem</code>, <code>MemW</code>, <code>MemL</code>, <code>Port</code>, <code>PortW</code>;</li> <li>• Ви намагаєтесь звернутись у асемблерному операнді до константи рядкового типу, множини, або дійсної константа з плаваючою точкою;</li> <li>• Ви намагаєтесь звернутись у асемблерному операнді до <code>inline</code> процедури або функції;</li> <li>• Ви намагаєтесь звернутись до спеціального символу <code>@Result</code> поза функцією;</li> <li>• Ви намагаєтесь генерувати коротку <code>JMP</code> інструкцію, яка переходить на щось, відмінне від мітки.</li> </ul>
161 Code generation error	<b>Помилка генерації коду</b> Попередня операторна частина містить інструкцію <code>LOOPNE</code> , <code>LOOPE</code> , <code>LOOP</code> або <code>JCZX</code> , яка не може досягнути призначеної мітки.
162 ASM expected	<b>Очікується ASM</b> Ви намагаєтесь відкомпілювати вбудовану асемблерну функцію, яка містить оператори <code>begin...end</code> , замість <code>asm...end</code> .
163 Duplicate dynamic method index	<b>Дублювання індексу динамічного методу</b> Даний індекс динамічного методу вже використовується іншим методом. Можливо ви намагались перекрити динамічний метод, але припустили помилку у його імені.
164 Duplicate resource identifier	<b>Дублювання ідентифікатора ресурсу</b> Тільки для захищеного і Windows-режимів. Даний файл ресурсів містить ресурс з ім'ям або ідентифікатором, яке вже використовується іншим ресурсом.
165 Duplicate or invalid export clause	<b>Індекс експорту дублюється або є невірним</b> Тільки для захищеного і Windows-режимів. Порядковий номер, визначений у реченні <code>index</code> , виходить за межі діапазону 1..32 767 або вже використовується іншим модулем, що експортується.
166 Procedure or function identifier expected	<b>Очікується ідентифікатор процедури або функції</b> Тільки для захищеного і Windows-режимів. У реченні <code>export</code> можуть використовуватись тільки процедури або функції.
167 Cannot export	<b>Не можна експортувати даний символ</b>

this symbol	Тільки для захищеного і Windows-режимів. Процедура або функція не може експортуватись, якщо вона не була оголошена із директивою export.
168 Duplicate export name	<b>Дублювання імені, що експортується</b> Тільки для захищеного і Windows-режимів. Ім'я, визначене у реченні name, вже використовується іншим модулем, що експортується.
169 Executable file header too large	<b>Занадто великий заголовок файлу, що виконується</b> Тільки для захищеного і Windows-режимів. Розмір заголовка EXE-файла перевищує межу 64К. Скоріше за все, ви імпортуєте або експортуєте занадто багато процедур або функцій
170 Too many segments	<b>Занадто багато сегментів</b> Тільки для захищеного і Windows-режимів. EXE-файл містить понад 254 сегментів. Можливо, розмір сегменту коду для файлу є маленьким. Збільшіть його за допомогою директиви компілятора {\$S розмір_сегменту}.

## Додаток В. Коди помилок під час виконання програми

Помилки періоду виконання програми можуть бути розділені на чотири категорії:

1. [Помилки операційної системи: 1 - 99;](#)
2. [Помилки введення/виведення: 100-149;](#)
3. [Критичні помилки: 150-199;](#)
4. [Фатальні помилки: 200-255.](#)

Код помилки і повідомлення	Причини помилки
<b>Помилки операційної системи</b>	
1 Invalid function number	<b>Невірний номер функції</b> Ви викликаєте функцію DOS, яка не існує.
2 File not found	<b>Файл не знайдений</b> Видається або процедурами Reset, Rewrite, Append, Rename, якщо невірно вказане ім'я файлу, або процедурою Erase, якщо ім'я, призначене файлової змінній, вказує на файл, який не існує.
3 Path not found	<b>Шлях не знайдений</b> Видається процедурами Reset, Rewrite, Append, Rename, Erase, якщо ім'я, призначене файлової змінній, вказує на файл, що не існує або вказує на неіснуючий каталог. Також видається процедурами ChDir, Mkdir, Rmdir, якщо шлях невірний або вказує неіснуючий каталог.
4 Too many open files	<b>Занадто багато відкритих файлів</b> Видається процедурами Reset, Rewrite, Append, якщо програма має занадто багато відкритих файлів.
5 File access denied	<b>Відмова у доступі до файлу</b> <ul style="list-style-type: none"> <li>• Видається процедурами Reset, Append, якщо FileMode дозволяє запис, а ім'я файлової змінної задає каталог або файл, який призначений тільки для зчитування;</li> <li>• Видається процедурою Rewrite, якщо каталог заповнений або ж ім'я, присвоєне файлової змінній, вказує на каталог або файл, призначений тільки для зчитування;</li> <li>• Видається процедурою Rename, якщо ім'я файлової змінної задає каталог або задає файл, що існує;</li> <li>• Видається процедурою Erase, якщо ім'я, присвоєне файлової</li> </ul>

	<p>змінній, вказує на каталог або файл, призначений тільки для зчитування;</p> <ul style="list-style-type: none"> <li>• Видається процедурою Mkdir, якщо файл з таким ім'ям існує у "батьківському" каталозі, або якщо шлях доступу вказує на пристрій, або якщо немає місця у "батьківському" каталозі;</li> <li>• Видається процедурою Rmdir, якщо каталог непустий; якщо шлях доступу вказує каталог, або шлях доступу вказує не кореневий каталог;</li> <li>• Видається процедурою Read або BlockRead для типізованого або не типізованого файла, якщо цей файл не відкритий для зчитування;</li> <li>• Видається процедурою Write або BlockWrite для типізованого або не типізованого файла, якщо цей файл не відкритий для записування;</li> </ul>
6 Invalid file handle	<b>Невірний дескриптор файла</b> Така помилка з'являється, якщо невірний дескриптор файла передається виклику системи DOS. Такого ніколи не повинно бути, але якщо таке трапляється, то це вказує на те, що файлова змінна якимось чином зіпсувалась.
12 Invalid file access code	<b>Невірний код доступу до файла</b> Видається процедурами Reset або Append для типізованого або не типізованого файла, якщо значення FileMode є невірним.
15 Invalid drive number	<b>Невірний номер пристрою</b> Видається процедурами GetDir або ChangeDir, якщо номер пристрою є невірним.
16 Cannot remove current directory	<b>Не можна знищити поточний каталог</b> Видається процедурою Rmdir, якщо шлях визначає поточний каталог.
17 Cannot rename across drive	<b>Не можна перейменувати файл, змінюючи ім'я пристрою</b> Видається процедурою Rename, якщо вказуються імена файлів із різними пристроями.
18 No more files	<b>Більше файлів немає</b> Видається через змінну DosError у модулях Dos і WinDos, якщо процедури FindFirst і FindNext не знаходять файла із визначеними іменем і множиною атрибутів.
<b><i>Помилки введення/виведення</i></b>	
Ці помилки викликають завершення програми при компіляції із директивою {\$I+}. У стані {\$I-} програма продовжує роботу, а помилка фіксується функцією IOResult	
100 Disk read error	<b>Помилка зчитування з диска</b> Видається процедурою Read для типізованого файла при спробі зчитати інформацію після завершення файла.
101 Disk write error	<b>Помилка запису на диск</b> Видається процедурами Close, Write, Writeln, Flush, якщо диск заповнений.
102 File not assigned	<b>Файл не призначений</b> Видається процедурами Reset, Rewrite, Append, Rename, Erase, якщо файлової змінній не було присвоєно ім'я за допомогою процедури Assign.
103 File not open	<b>Файл не відкритий</b> Видається процедурами Close, Read, Write, EoF, Seek, EoLn, FilePos, FileSize, Flush, BlockRead, BlockWrite, якщо файл не був відкритий.
104 File not open for input	<b>Файл не відкритий для введення</b> Видається процедурами Read, Readln, EoF, SeekEoF, EoLn, SeekEoLn для текстових файлів, якщо файл не був відкритий для введення.
105 File not open for output	<b>Файл не відкритий для виведення</b>

	Видається процедурами Write, Writeln для текстових файлів, якщо файл не був відкритий для виведення.
107 Invalid numeric format	<b>Невірний числовий формат</b> Видається процедурами Read, Readln, якщо числове значення, прочитане із текстового файла не відповідає правильному числовому формату.
<b>Критичні помилки</b>	
Трапляються тільки у реальному і захищеному режимах.	
150 Disk is write-protected	<b>Диск захищений від запису</b>
151 Unknown unit	<b>Невідомий пристрій</b>
152 Drive not ready	<b>Пристрій не готовий</b>
153 Unknown command	<b>Невідома команда</b>
154 CRC error in data	<b>Помилка у даних, знайдена при контролі циклічним надлишковим кодом</b>
155 Bad drive request structure length	<b>Невірний запит пристрою про довжину структури</b>
156 Disk seek error	<b>Помилка пошуку на диску</b>
157 Unknown media type	<b>Невідомий тип носія</b>
158 Sector not found	<b>Сектор не знайдений</b>
159 Printer out of paper	<b>У принтері немає паперу</b>
160 Device write fault	<b>Помилка запису на пристрій</b>
161 Device read fault	<b>Помилка зчитування з пристрою</b>
162 Hardware failure	<b>Відмова апаратних засобів</b>
<b>Фатальні помилки</b>	
Такі помилки одразу ж припиняють виконання програми	
200 Division by zero	<b>Ділення на нуль</b> Спроба розділити на нуль при виконанні операцій /, mod, або div.
201 Range check error	<b>Вихід за межі діапазону</b> Ця помилка видається операторами, відкомпільованими з директивою {\$R+}, коли виникає одна із ситуацій: <ul style="list-style-type: none"> <li>• Індексний вираз кваліфікатора масиву виходить за межі припустимого діапазону;</li> <li>• Була спроба присвоїти змінній значення, яке виходить за межі припустимого діапазону;</li> <li>• Була спроба передати процедурі або функції як параметр значення, яке виходить за межі діапазону.</li> </ul>
202 Stack overflow error	<b>Переповнення стеку</b> Ця помилка з'являється при входженні у процедуру або функцію, відкомпільовану з директивою {\$S+}, коли у стеку недостатньо місця для розміщення локальних змінних підпрограм. У такому випадку слід збільшити розмір стеку за допомогою директиви компілятора \$M. Ця помилка може виникати також внаслідок нескінченної рекурсії або асемблерної процедури, яка належним чином не підтримує стек.
203 Heap overflow error	<b>Переповнення купи</b> Ця помилка видається процедурами New або GetMem, коли у купі недостатньо місця для розміщення блоку потрібного розміру.
204 Invalid pointer	<b>Невірна операція з покажчиком</b>

operation	Ця помилка видається процедурами Dispose або FreeMem, якщо показчик вказує на nil або посилається на позицію за межею купи.
205 Floating point overflow	<b>Переповнення при операції з плаваючою точкою</b> Операція з плаваючою точкою обробляє занадто велике для компілятора або сопроцесора число.
206 Floating point underflow	<b>Втрата значущості при операції з плаваючою точкою</b> Операція з плаваючою точкою призводить до втрати значущості. Ця помилка видається лише у тому випадку, якщо використовується числовий сопроцесор 8087 з контрольним словом, яке не маскує втрату значущості. Звичайно втрата значущості призводить до нульового результату.
207 Invalid floating point operation	<b>Невірна операція з плаваючою точкою</b> <ul style="list-style-type: none"> <li>• Дійсне значення, яке передається функціям trunc або round, не може бути перетворене у ціле число всередині діапазону для типу LongInt;</li> <li>• Аргумент, який передається функції Sqrt, є від'ємним числом;</li> <li>• Аргумент, який передається функції Ln, дорівнює нулю або є від'ємним;</li> <li>• Відбулося переповнення стеку сопроцесора 8087.</li> </ul>
208 Overlay manager not installed	<b>Не встановлений монітор оверлеїв</b> Тільки для реального режиму. Програма викликає оверлейну процедуру або функцію, але монітор оверлеїв не інстальований. Скоріше за все ви не викликали OvrInit або виклик OvrInit був невдалим. Зверніть увагу на те, що якщо ініціювання знаходиться у будь-якому з оверлейних модулів, тоді потрібно створити додатковий не оверлейний модуль, який викликає OvrInit, і використовувати цей модуль перед будь-яким з оверлейних модулів.
209 Overlay file read error	<b>Помилка зчитування оверлейного файла</b> Тільки для реального режиму. Трапилася помилка зчитування, коли монітор оверлеїв намагався прочитати оверлей з оверлейного файла.
210 Object not initialized	<b>Об'єкт не ініційований</b> Коли включена перевірка діапазону, ви викликаєте віртуальний метод об'єкта через виклик конструктора.
211 Call to abstract method	<b>Виклик абстрактного методу</b> Ця помилка генерується процедурою abstract у модулі objects. Вона вказує на те, що ваша програма намагається виконати абстрактний віртуальний метод. Коли об'єктовий тип містить один або більше абстрактних методів, він називається абстрактним об'єктним типом. Створення екземплярів об'єктів абстрактного типу буде помилкою - абстрактні типи існують лише для того, щоб ви могли успадковувати від них абстрактні методи і перекривати їх.
212 Stream registration error	<b>Помилка реєстрації потоку</b> Ця помилка генерується процедурою RegisterType у модулі Objects, вказуючи на те, що трапилася одна із ситуацій: <ul style="list-style-type: none"> <li>• Запис реєстрації потоку не розміщується у сегменті даних;</li> <li>• Поле ObjType запису реєстрації потоку дорівнює 0;</li> <li>• Тип вже зареєстрований;</li> <li>• Існує інший тип з тим самим значенням ObjType.</li> </ul>
213 Collection index out of range	<b>Індекс набору виходить за межі</b> Індекс, що передається у метод із TCollection, виходить за межі припустимого діапазону.
214 Collection overflow error	<b>Помилка переповнення набору</b> Ця помилка видається TCollection при спробі додати елемент, коли набір не може бути розширений.
215 Arithmetic overflow error	<b>Помилка арифметичного переповнення</b> Ця помилка видається при компіляції з директивою {\$O+}, коли цілочислова арифметична операція тягне за собою переповнення, наприклад, при виході результату операції за межі діапазону.

216 General Protection fault	<p><b>Помилка загального захисту</b></p> <p>Тільки для реального режиму. Ця помилка виникає при невдалій спробі отримати з програми доступ до певної ділянки пам'яті. Операційна система припиняє виконання вашої програми і повідомляє про помилку "загального захисту". При цьому система продовжує працювати. Помилка може статися через такі причини:</p> <ul style="list-style-type: none"> <li>• Завантаження сталої величини у сегментний регістр;</li> <li>• Виконання арифметичних операцій із сегментними регістрами селекторів;</li> <li>• Використання сегментних регістрів для збереження поточних результатів процесу запису у сегменти коду;</li> <li>• Спроба отримати доступ до пам'яті, яка знаходиться за межами виділеного програмі адресного простору;</li> <li>• Спроба "розадресування" nil-показчиків.</li> </ul>
---------------------------------	---

## Додаток Г. Теми рефератів

1. Перші механічні лічильні пристрої
2. Аналітична машина Чальза Беббіджа
3. Перші електромеханічні лічильні пристрої
4. Алан Т'юрінг і його внесок у розвиток комп'ютерної техніки і технології
5. Джон фон Нейман і принципи функціонування універсальних обчислювальних пристроїв
6. Перші електронні обчислювальні машини
7. Академік Лебедев С.О. і його внесок у розвиток вітчизняної обчислювальної техніки
8. Академік Глушков В.М. і його внесок у розвиток вітчизняної обчислювальної техніки
9. Історія і здобутки Інституту кібернетики НАН України ім. В.М. Глушкова
10. Винаходження транзистора, принципи його дії та можливості застосування у комп'ютерній техніці
11. Винаходження мікросхем та можливості їхнього застосування у комп'ютерній техніці
12. Розвиток персональних комп'ютерів у 70-х роках ХХ сторіччя
13. Розвиток персональних комп'ютерів у 80-90-х роках ХХ сторіччя
14. Розвиток мережевих технологій та інтернету
15. Стародавні системи числення
16. Сучасні материнські плати для персональних комп'ютерів
17. Сучасні центральні процесори для персональних комп'ютерів
18. Сучасні модулі оперативної пам'яті для персональних комп'ютерів
19. Сучасні накопичувачі на жорстких магнітних дисках для персональних комп'ютерів
20. Сучасні накопичувачі на оптичних дисках для персональних комп'ютерів
21. Сучасні магнітооптичні накопичувачі для персональних комп'ютерів
22. Розвиток Flash-пам'яті для персональних комп'ютерів
23. Сучасні відеомонітори на електронних променевих трубках для персональних комп'ютерів
24. Сучасні рідкокристалічні відеомонітори для персональних комп'ютерів
25. Сучасні відеоадаптери для персональних комп'ютерів
26. Сучасні струменеві принтери для персональних комп'ютерів
27. Сучасні лазерні принтери для персональних комп'ютерів
28. Сучасні сканери для персональних комп'ютерів
29. Сучасні акустичні системи і звукові адаптери для персональних комп'ютерів
30. Сучасні комунікаційні пристрої для персональних комп'ютерів

## Додаток Д. Завдання до практичної частини



За номером варіанту індивідуального завдання ви маєте можливість ознайомитись із кресленням механізму та відповідними формулами, а також додатково переглянути анімовану тривимірну модель механізму і подивитись, як такий механізм працює.

### Варіанти:

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)  
[11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#)  
[21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#)

## Додаток Е. Допоміжні модулі

На компакт-диску у папці "Units" зосереджені вихідні тексти модулів, які можуть стати у нагоді при виконанні курсової роботи або реалізації інших проектів. Для використання елементів модулів у власних програмах слід зробити таке:

- Скопіювати вихідні тексти модулів до власної папки, або до папки, яка є у переліку "Unit Directories" ([див. 19.3.3](#));
- По одному відкомпілювати необхідні модулі та переконатися у появі відповідних tru-файлів;
- Підключити потрібні модулі до вашої програми ([див. 19.1](#));
- Скористатися в програмі необхідними елементами з підключених модулів.

Далі розглянемо модулі та їхні елементи.

---

## Модулі для Pascal-програм

При створенні Pascal-програм можна скористатися елементами таких модулів:

- [PVMath](#) - модуль математичних функцій;
  - [PVServis](#) - сервісний модуль, який дозволяє керувати видимістю текстового курсору, а також створювати текстові вікна;
  - [PVEdit](#) - модуль організації коректного введення початкових даних рядкового, цілочислового і дійсного типів;
  - [PVMenu](#) - модуль організації текстових меню;
  - [PVMech](#) - модуль для вимальовування елементів окремих ланок механізмів у графічному режимі;
  - [PVGr](#) - модуль для формування графіків.
- 

### Елементи модулю PVMath

Модуль містить три математичні функції, які доволі часто зустрічаються при моделюванні роботи механізмів, але відсутні серед стандартних математичних функцій.

**function** arcsin(x:real):real; - розраховує арксинус аргументу. Аргумент -

величина без розмірності, результат функції - кут в радіанах.

**function** arcctn(x:real):real; - розраховує арккотангенс аргументу. Аргумент - величина без розмірності, результат функції - кут в радіанах.

**function** arccos(x:real):real; - розраховує арккосинус аргументу. Аргумент - величина без розмірності, результат функції - кут в радіанах.

## Елементи модулю PVServis

Модуль містить чотири процедури, які дозволяють керувати [видимістю текстового курсору](#), визначати [текстові вікна](#) та [кольори](#).

**Procedure** Colors(Text, Back:byte); - дозволяє встановити колір тексту (параметр Text) і фону (параметр Back). Можливі варіанти кольорів наведені у таблиці Е.1.

*Приклад:*

Colors(Yellow, Blue);

- встановлено синій колір фону і жовтий колір символів.

Таблиця Е.1. Кольори у текстовому режимі

Число	Константа	Колір	Зразок
<i>Темні кольори (для символів і фону)</i>			
0	Black	Чорний	
1	Blue	Темно-синій	
2	Green	Зелений	
3	Cyan	Морської хвилі	
4	Red	Темно-червоний	
5	Magenta	Фіолетовий	
6	Brown	Оливковий	
7	LightGray	Світло-сірий	
<i>Світлі кольори (тільки для символів)</i>			
8	DarkGray	Темно-сірий	
9	LightBlue	Синій	
A	LightGreen	Світло-зелений	
B	LightCyan	Блакитний	
C	LightRed	Світло-червоний	
D	LightMagenta	Світло-фіолетовий	
E	Yellow	Жовтий	

F	White	Білий
---	-------	-------

**Procedure** `CursorOn;` - робить текстовий курсор видимим. Такий стан видимості курсору є "стандартним" і необхідне для організації введення даних.

**Procedure** `CursorOff;` - робить текстовий курсор невидимим. Така дія може бути корисною для організації меню в програмі. Наприкінці вашої програми не забудьте включити відображення курсору.

**Procedure** `BWindow(LHCol, LHRow, RLCol, RLRow, FrameType, Shadow, WindAttr:byte);`

- дозволяє визначити текстове вікно, яке може займати довільне місце на екрані, мати одинарну або подвійну рамку (або не мати такої), імітувати або ні наявність тіні "під вікном", а саме вікно додатково може формуватись різними кольорами. Параметрами процедури є:

`LHCol` - стовпчик, в якому розташований верхній лівий кут вікна.

`LHRow` - рядок, в якому розташований верхній лівий кут вікна.

`RLCol` - стовпчик, в якому розташований нижній правий кут вікна.

`RLRow` - рядок, в якому розташований нижній правий кут вікна.

Значення стовпчиків (`Col`) у текстовому режимі відраховуються від лівого боку праворуч і можуть отримувати значення від 1 до 80. Значення рядків (`Row`) відраховуються від верхнього краю екрану вниз і можуть отримувати значення від 1 до 25. Слідкуйте за тим, щоб координати правого нижнього кута були більшими за відповідні координати лівого верхнього кута.

`FrameType` - вказує на наявність рамки: одинарної (`FrameType=1`), подвійної (`FrameType=2`), або на її відсутність (`FrameType=0`).

`Shadow` - вказує на наявність (`Shadow=1`) або відсутність (`Shadow=0`) "тіні під вікном".

`WindAttr` - визначає колір фону і колір символів для вікна. Таке значення визначається, як шістнадцяткове число - `$AB`, де `A` - колір фону вікна (значення від 0 до 7), `B` - код кольору символів (значення від 0 до 9 та від `A` до `F`). Можливі значення кольорів наведені у [табл. Е.1](#).

*Приклад:*

```
BWindow(20, 5, 60, 15, 2, 0, $70);
```

- визначено вікно, лівий верхній кут якого розміщений у 20-му стовпчику 5-го рядка, правий нижній кут розміщений у 60-му стовпчику 15-го рядка, вікно буде обведене подвійною рамкою, не буде мати "тіні", фон вікна буде світло-сірим (код 7), а колір символів - чорним (код 0)

## Елементи модулю PVEdit

Модуль містить три процедури, які дозволяють організувати зручне і безпомилкове введення з клавіатури значень змінних [рядкового](#), [цілочислового](#) і [дійсного](#) типів, а також, [логічну функцію](#), яка коректно буде обробляти натискання клавіш "Y", "N" на клавіатурі.

**Procedure** `ReadString(Prompt:string; Col, Row, MaxLen, PromptAttr, StringAttr:byte; var S:string);`

- дозволяє організувати введення з клавіатури значення рядкової змінної у визначеному місці на екрані, певними кольорами, з певною підказкою. Параметрами процедури є:

`Prompt` - текстова підказка, яка буде виводитись перед введенням рядкового значення.

`Col` - стовпчик, у якому буде розпочинатися виведення підказки (`Prompt`). Може приймати значення від 1 до 80.

`Row` - рядок, у якому буде виводитись підказка (`Prompt`). Може приймати значення від 1 до 25.

`MaxLen` - максимальна довжина поля, яке відводиться під введення значення рядкової змінної.

`PromptAttr` - визначає колір фону і тексту для підказки.

`StringAttr` - визначає колір фону і тексту для поля введення значення рядкової змінної.

Значення параметрів `PromptAttr` і `StringAttr` вказуються аналогічно до параметру `WindAttr` процедури [BWindow](#).

`S` - ім'я змінної, яке передає до процедури початкове значення рядка і після редагування отримує його назад. Якщо перед викликом процедури `ReadString` значення змінної буде непустим, то це значення з'явиться, як значення "за замовчуванням", в іншому випадку поле для введення буде пустим.

*Приклад:*

```
ReadString('Прізвище:', 10, 3, 15, $07, $70, Prizv);
```

- призведе до виведення від 10-го стовпчика 3-го рядка на чорному фоні світло-сірими символами тексту "Прізвище:" і дозволить ввести до 15 символів прізвища на чорному фоні світло-сірими символами. Після натискання клавіші *Enter* результат буде присвоєний змінній `Prizv`, яка повинна бути попередньо визначена у вашій програмі.

```
procedure ReadInteger(Prompt:string; Col,Row,MaxLen,
PromptAttr,StringAttr:byte; MinValue,MaxValue:integer; var
I:integer);
```

- дозволяє організувати введення з клавіатури значення цілочислової змінної у визначеному місці на екрані, певними кольорами, з певною підказкою і контролем потрапляння у діапазон. Однойменні параметри є аналогічними параметрам попередньої процедури [ReadString](#).

Новими параметрами процедури є:

`MinValue` - мінімально можливе уведене значення.

`MaxValue` - максимально можливе уведене значення.

`I` - ім'я цілочислової змінної, яка передає до процедури початкове значення цілого числа і після редагування отримує його назад. Якщо перед викликом процедури `ReadInteger` значення змінної буде непустим, то це значення з'явиться, як значення "за замовчуванням", в іншому випадку поле для введення буде пустим.

*Приклад:*

```
ReadInteger('Діаметр заготовки (10-200мм) =',
12, 5, 3, $07, $70, 10, 200, d1);
```

- призведе до виведення від 12-го стовпчика 5-го рядка на чорному фоні світло-сірими символами тексту "Діаметр заготовки (10-200мм) =" і дозволить ввести до 3-х числових символів на чорному фоні світло-сірими. Після натискання клавіші *Enter* результат буде проконтрольований і, у випадку потрапляння його у діапазон 10-200, присвоєний змінній `d1`, яка повинна бути попередньо визначена у вашій програмі. У випадку некоректного введення, поле введення автоматично очиститься і буде запропоновано повторити введення.

```
procedure ReadReal (Prompt:string; Col,Row,MaxLen, DecimalPlace,
PromptAttr,StringAttr:byte; MinValue,MaxValue:real; var R:real);
```

- дозволяє організувати введення з клавіатури значення дійсної змінної у визначеному місці на екрані, певними кольорами, з певною підказкою і контролем потрапляння у діапазон, з певною точністю. Однойменні параметри є аналогічними параметрам попередньої процедури [ReadInteger](#). Новими параметрами процедури є:

`DecimalPlace` – кількість позицій, яка відводиться на дробову частину дійсного числа.  
`R` – ім'я дійсної змінної, яка передає до процедури початкове значення дійсного числа і після редагування отримує його назад. Якщо перед викликом процедури `ReadReal` значення змінної буде непустим, то це значення з'явиться, як значення "за замовчуванням", в іншому випадку поле для введення буде пустим.

*Приклад:*

```
ReadReal('Подача (0.50-2.0 мм/об) =', 8,12,4,2, $07,$70,
0.05,2.00,s);
```

- призведе до виведення від 8-го стовпчика 12-го рядка на чорному фоні світло-сірими символами тексту "Подача (0.50-2.0 мм/об) =" і дозволить ввести до 4-х числових символів (з яких два останніх можуть бути десятими і сотими частинами числа) на чорному фоні світло-сірими. Після натискання клавіші *Enter* результат буде проконтрольований і, у випадку потрапляння його у діапазон 0,05-2,00, присвоєний змінній `s`, яка повинна бути попередньо визначена у вашій програмі. У випадку некоректного введення, поле введення автоматично очиститься і буде запропоновано повторити введення.

**Function** `YorNo(Prompt:string; Col,Row,PromptAttr:byte):boolean;`

- дозволяє організувати виведення на екран запиту відповідного кольору, у відповідному місці і коректно обробити натискання клавіш "Y" і "N" на клавіатурі (незалежно від стану регістру і мови). Всі параметри цієї функції є аналогічними до однойменних параметрів процедур, які були розглянуті вище. Сама функція повертає логічний результат, що робить зручним її використання в операторах перевірки умови.

*Приклад:*

```
if YorNo('Записати результати у файл ?',30,12,$07)
  then begin
    {Блок операторів, який реалізує запис результатів у файл}
  end;
```

- наведений фрагмент коду програми дозволяє вивести від 30-ї позиції 12-го рядка запит "Записати результати у файл ?" і, у випадку натискання клавіші "Y" (позитивна відповідь), виконати блок операторів запису результатів у файл. У випадку негативної відповіді (натискання клавіші "N") запис у файл не виконуватиметься. Виведення тексту відбувається на чорному фоні світло-сірими символами (значення \$07).

## Елементи модулю **PVMenu**

Модуль містить одну процедуру, дві константи і два типи, які дозволяють організувати виведення текстових меню.

**Константи:**

`LStr=40` – максимальна довжина пункту меню (до 40 символів).

`NMenu=10` – максимальна кількість пунктів меню (до 10 пунктів).

**Типи:**

`MenuArr=array[1..NMenu] of string[LStr];` - визначає тип змісту меню, яке може складатися не більше ніж з 10 пунктів, довжиною не більше ніж 40 символів.

`MType = (V,G);` - визначає вертикальний (V) або горизонтальний (G) тип меню.

**Procedure** `SMenu(var Nd:byte;Col,Row,NuStr:byte; MenuStr:MenuArr;`

```
Typ: MTyp; SaveAttr, MainAttr, HiAttr, Frame, Shadow: byte);
```

- процедура дозволяє у позиції екрану вивести горизонтальне або вертикальне текстове меню, організувати вибір відповідного пункту шляхом натискання на клавіші управління курсором і після натискання клавіш *Enter* або *Esc* повернути у програму код вибраного пункту.

Параметрами процедури є:

**Nd** - код вибраного пункту меню. Після виходу з процедури при натисканні клавіші *Enter* змінна буде містити номер вибраного пункту. при завершенні роботи з меню натисканням клавіші *Esc* змінна матиме значення "0".

**Col** - стовпчик, у якому буде розпочинатися виведення меню. Може приймати значення від 1 до 80.

**Row** - рядок, у якому буде виводитись меню. Може приймати значення від 1 до 25.

**NuStr** - кількість пунктів меню. Зміст меню (змінна *MenuStr*) повинен мати точно 10 пунктів, а змінна *NuStr* визначає скільки з них слід виводити на екран. Значення цієї змінної повинно знаходитись у межах 2-10.

**MenuStr** - змінна типу *MenuArr*, яка зберігає зміст пунктів вашого меню. Кількість таких пунктів повинна обов'язково дорівнювати 10. Якщо вам потрібна менша кількість, необхідно додати "пусті" пункти. Зміст цієї змінної потрібно визначити у власній програмі (див. приклад далі).

**Typ** - визначає тип (орієнтацію меню). Якщо вказати значення "V" або "v", буде виводитись вертикальне меню, Якщо вказати значення "G" або "g", буде виводитись горизонтальне меню.

**SaveAttr** - визначає атрибути кольорів вікна або екрану, поверх якого виводиться меню.

**MainAttr** - визначає атрибути кольорів, якими виводяться всі пункти меню за винятком активного.

**HiAttr** - визначає атрибути кольорів, якими виводиться активний (обраний) пункт меню.

Атрибутів кольорів визначаються, як шістнадцяткові числа - \$AB, де B - колір фону вікна (значення від 0 до 7) A - код кольору символів (значення від 0 до 9 та від A до F). Можливі варіанти кольорів вказані у [таблиці E.1](#).

**Frame** - вказує на наявність рамки: одинарної (*Frame*=1), подвійної (*Frame*=2), або на її відсутність (*Frame*=0).

**Shadow** - вказує на наявність (*Shadow*=1) або відсутність (*Shadow*=0) "тіні під меню".

Приклад:

```
const {Розділ описування констант вашої програми}
  MenuMain: MenuArr =
    (' Розрахунки ',
     ' Рисунки ',
     ' Рух ',
     ' Графіки ',
     ' Інформація ',
     ' Вихід ', '', '', '', '');
  MenuProces: MenuArr =
    (' Введення даних ',
     ' Результати ', '', '', '', '', '', '', '');
  . . . . .
{Використання меню у вашій основній програмі}
  CursorOff;
  SMenu (KodMain, 1, 1, 6, MenuMain, g, $07, $07, $70, 2, 0);
  case KodMain of
    1: repeat SMenu (KodProces, 1, 4, 2, MenuProces, v, $07, $07, $70, 1, 0);
      case KodProces of
        1: InputDatas;
        2: Result;
      end; {case}
```

```

        until KodProces=0;
        . . . .
    end;

```

У наведеному фрагменту спочатку визначається зміст двох меню: MenuMain - головного меню і MenuProcess - меню для розрахунків. Зверніть увагу, що загальна кількість пунктів цих двох меню, включаючи "пусті пункти" складає 10. Зручно, щоб усі пункти меню мали однакову довжину, тоді при виборі пунктів не буде змінюватись довжина виділеного транспаранта. Також бажано перед першим символом пункту меню і після останнього залишати принаймні один пробіл, - у такому випадку покращується сприйняття такого меню.

Далі в розділі операторів основної програми перед викликанням процедури роботи з меню слід погасити мерехтіння текстового курсору (процедура CursorOff).

До першої процедури SMenu прикладу передаються такі параметри:

KodMain - ім'я змінної, яка після завершення роботи процедури отримає код вибраного пункту меню;

1, 1 - виводити меню починаючи від першого стовпчика першого рядка екрана;

6 - кількість потрібних транспарантів;

MenuMain - ім'я змінної (типізованої константи), яка містить тексти пунктів меню;

g - горизонтальний тип меню;

\$07 - меню накладається на початкове вікно зі світло сірими символами на чорному фоні;

\$07 - меню виводиться світло-сірими символами на чорному фоні;

\$70 - активний пункт меню виводиться чорними символами на світло-сірому фоні (інверсно до основних кольорів);

2 - подвійна рамка навколо меню;

0 - відсутність імітації "тіні під меню".

Далі в операторі case аналізується код вибраного пункту меню і, якщо він дорівнює "1" викликається наперед визначена процедура InputData, якщо він дорівнює "2" - наперед визначена процедура Result оператор циклу repeat...until.

Тут багаторазово (до натискання клавіші *Esc*) може викликатись наступна вкладена процедура SMenu, до якої передаються параметри:

KodProces - ім'я змінної, яка після завершення роботи процедури отримає код вибраного пункту меню;

1, 4 - виводити меню починаючи від першого стовпчика першого рядка екрана;

2 - кількість потрібних транспарантів;

MenuProces - ім'я змінної (типізованої константи), яка містить тексти пунктів меню;

v - горизонтальний тип меню;

\$07 - меню накладається на початкове вікно зі світло сірими символами на чорному фоні;

\$07 - меню виводиться світло-сірими символами на чорному фоні;

\$70 - активний пункт меню виводиться чорними символами на світло-сірому фоні (інверсно до основних кольорів);

1 - одинарна рамка навколо меню;

0 - відсутність імітації "тіні під меню".

Докладно створення розгалужених меню за допомогою процедури SMenu представлено у прикладі виконання курсової роботи у розділі кодування основної програми.



## Елементи модулю PVMech

Модуль містить 13 процедур, які спрощують створення параметричних моделей механізмів.

**procedure** Opora(x, y, Orient:integer);

- процедура виведення нерухокої опори (без кола у нерухокій точці). Параметрами процедури є:

x, y - координати нерухокої точки опори.

Orient - кут повороту основи опори. Може приймати значення 0, 90, 180 і 270, що відповідає повороту основи опори відносно додатного напрямку вісі X. Використання процедури проілюстровано рисунком.



**procedure** Uzel(x, y:integer; Name:string);

- процедура виведення вузла механізму. Вузол виводиться як невелике коло із надписом біля нього. Параметрами процедури є:

x, y - координати центру вузла.

Name - текст біля вузла (ім'я вузла).

Використання процедури проілюстровано рисунком.



**procedure** RazmerLinear(x1, y1, x2, y2:integer; Text:string);

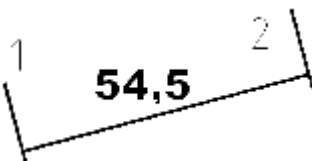
- процедура виведення лінійного розміру від точки до точки. Процедура дозволяє виводити горизонтальні, вертикальні та нахилені лінійні розміри. Для спрощення стрілочки на розмірах не виводяться. Напис на розмірі завжди є текстовим, тому, якщо необхідно вивести числове значення, його слід попередньо перетворити у рядок. Для виведення розміру слід вказати такі параметри:

x1, y1 - координати першої точки розміру.

x2, y2 - координати другої точки розміру.

Text - позначення розміру або його величина (текст).

Використання процедури проілюстровано рисунком.



**procedure** RazmerAngular(x1, y1, x2, y2, x3, y3:integer; Text:string);

- процедура виведення кутового розміру. Дуга кутового розміру проводиться від точки 2 до точки 3 проти годинникової стрілки. Точка 1 визначає центр дуги.

Для виведення розміру слід вказати такі параметри:

x1, y1 - координати точки центру кутового розміру.

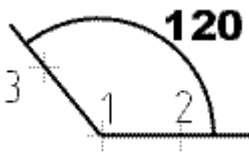
x2, y2 - координати точки від якої проводиться дуга.

x3, y3 - координати точки до якої проводиться дуга.

Text - позначення розміру або його величина (текст).

Використання процедури проілюстровано рисунком.





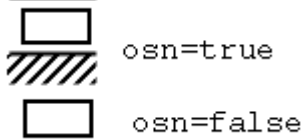
**procedure** PolzunHor (x, y: integer; osn: boolean);

- процедура виведення повзуна, що переміщується горизонтально. Базовою точкою повзуна є його центр. Повзуни можуть виводитись у двох модифікаціях - із напрямною (перша схема) і без напрямної (друга схема). Відображати чи ні такі напрямні визначає логічний параметр osn. Параметрами процедури є:

x, y - координати центральної точки повзуна.

osn - логічний параметр, який визначає, виводити (true) чи ні (false) напрямні для повзуна.

Використання процедури проілюстровано рисунками.



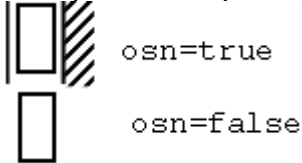
**procedure** PolzunVer (x, y: integer; osn: boolean);

- процедура виведення повзуна, що переміщується вертикально. Параметрами процедури є:

x, y - координати центральної точки повзуна.

osn - логічний параметр, який визначає, виводити (true) чи ні (false) напрямні для повзуна.

Використання процедури проілюстровано рисунками.



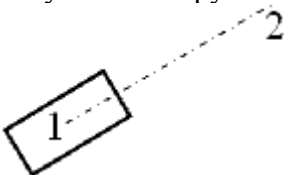
**procedure** PolzunAng (x1, y1, x2, y2: integer);

- процедура виведення повзуна, що має довільний нахил. Параметрами процедури є:

x1, y1 - координати центральної точки повзуна

x2, y2 - координати точки, на яку "дивиться" повзун.

Рисунок ілюструє використання процедури.



**procedure** Kulisa (x1, y1, x2, y2, l: integer);

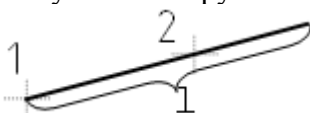
- процедура виведення спрощеної куліси (пряма лінія фіксованої довжини). Параметрами процедури є:

x1, y1 - координати нерухомої точки куліси.

x2, y2 - координати точки, через яку куліса проходить.

l - довжина куліси.

Рисунок ілюструє використання процедури (на екран виводиться тільки одна лінія).



**procedure** KulisaPaz ( $x1, y1, x2, y2, L, Lp:integer$ ; Torej:boolean);

- процедура виведення ускладненої куліси. Параметрами процедури є:

$x1, y1$  - координати нерухомої точки куліси.

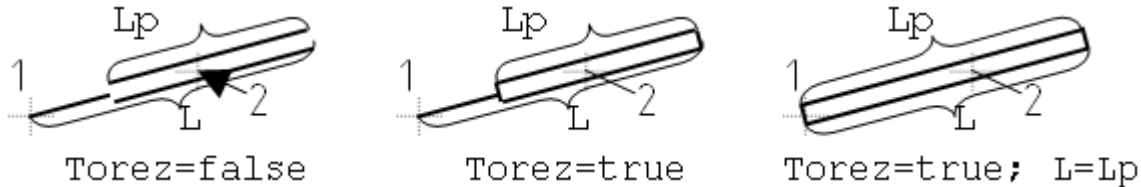
$x2, y2$  - координати точки, через яку куліса проходить.

$L$  - загальна довжина куліси.

$Lp$  - довжина пазу куліси (не повинна перевищувати  $L$ ).

Torej - логічний параметр, який визначає, виводити (true) чи ні (false) торцеві частини куліси.

Рисунки ілюструють використання процедури (на екран виводяться тільки товсті лінії).



**procedure** StoykaHor ( $x, y, LUp, LDn, Orient:integer$ );

- процедура виведення горизонтальної стійки з пазом. Процедура дозволяє вивести стійку, що може переміщуватись горизонтально. Базовою точкою стійки є точка, яка лежить на перетині пазу стійки та її напрямної. Параметр Orient дозволяє вибрати одне із положень стійки. Для лівої стійки на схемі він має значення 0, для правої - 180. Параметрами процедури є:

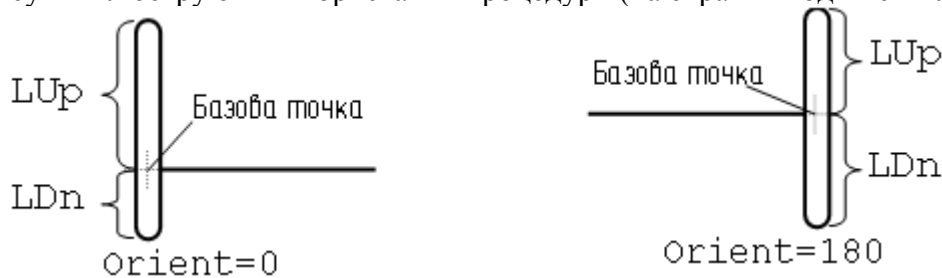
$x, y$  - координати базової точки стійки.

$LUp$  - розмір пазу вверх.

$LDn$  - розмір пазу вниз.

Orient - напрямок напрямної: 0 - управо, 180 - уліво.

Рисунки ілюструють використання процедури (на екран виводяться тільки товсті лінії).



**procedure** StoykaVer ( $x, y, LRt, LLf, Orient:integer$ );

- процедура виведення вертикальної стійки з пазом. Параметрами процедури є:

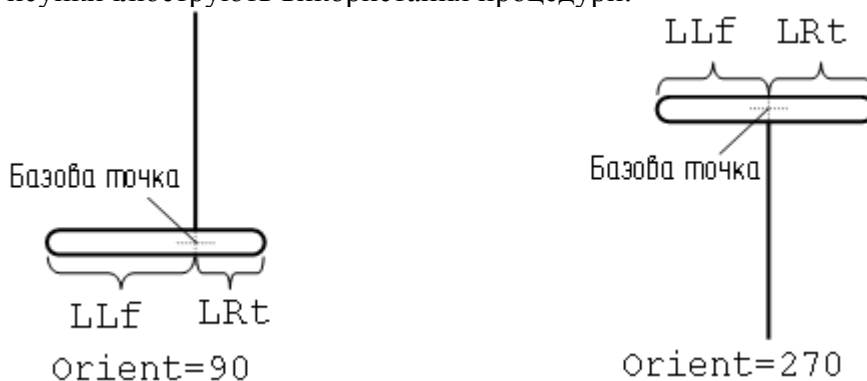
$x, y$  - координати базової точки стійки.

$LRt$  - розмір пазу вправо.

$LLf$  - розмір пазу вліво.

Orient - напрямок напрямної: 90 - уверх, 270 - униз.

Рисунки ілюструють використання процедури.



**procedure** NapravHor (x, y, l:integer);

- процедура виведення горизонтальної напрямної. Параметрами процедури є:  
 x, y - координати базової точки напрямної. (Точка завжди ліворуч).  
 l - довжина напрямної.

Рисунок ілюструє використання процедури.



**procedure** NapravVer (x, y, l:integer);

- процедура виведення вертикальної напрямної. Параметрами процедури є:  
 x, y - координати базової точки напрямної. (Точка завжди знизу).  
 l - довжина напрямної.

Рисунок ілюструє використання процедури.



## Елементи модулю PVGr

### Константи і типи

```

const
  NumbOfPoint = 360;
type
  ArrReal = array [0..NumbOfPoint] of real;
  
```

Константа NumbOfPoint визначає кількість точок, з яких буде складатися графік. ArrReal - масив, який зберігатиме значення, по яких будуть формуватися графіки.

### Процедура формування координатної сітки графіка

```

Procedure BuildCoord (X1, Y1, X2, Y2: real;
  GraphColor, LineColor, NumColor: word;
  Nx, AllX, DecX, Ny, AllY, DecY: integer;
  MinX, MaxX, MinY, MaxY: real;
  TextGraph, TextX, TextY: string;
  TextColor: word);
  
```

- процедура формування координатної сітки графіка. Така сітка автоматично отримує вісі, лінії градуювання, підписи до осей та графіка в цілому. На таку сітку в наступному (процедурою BuildChart) буде виводитись сам графік.

Параметрами процедури є:

X1, Y1 - координати лівого нижнього кута координатної сітки у долях екрана (від 0 до 1);

X2, Y2 - координати правого верхнього кута координатної сітки кінця у долях екрана (від 0 до 1); Такий вибір визначення координат початку і кінця системи координат пояснюється бажанням створити процедуру, яка б однаково її виводила на відеомоніторах різних типів.

GraphColor - колір робочої площини координатної сітки;

LineColor - колір ліній, якими виводиться координатна сітка;

NumColor - колір цифр градуювання осей координатної сітки;

Nx - кількість позицій розбиття осі абсцис ;

AllX - загальна кількість цифр у числах градуювання осі абсцис;

DecX - кількість цифр дробової частини градуювання осі абсцис (для цілих чисел - 0);

Ny - кількість позицій розбиття осі ординат ;

AllY - загальна кількість цифр у числах градуювання осі ординат;

DecY - кількість цифр дробової частини градуювання осі ординат (для цілих чисел - 0);

MinX, MaxX - мінімальне і максимальне значення осі абсцис;

MinY, MaxY - мінімальне і максимальне значення осі ординат;

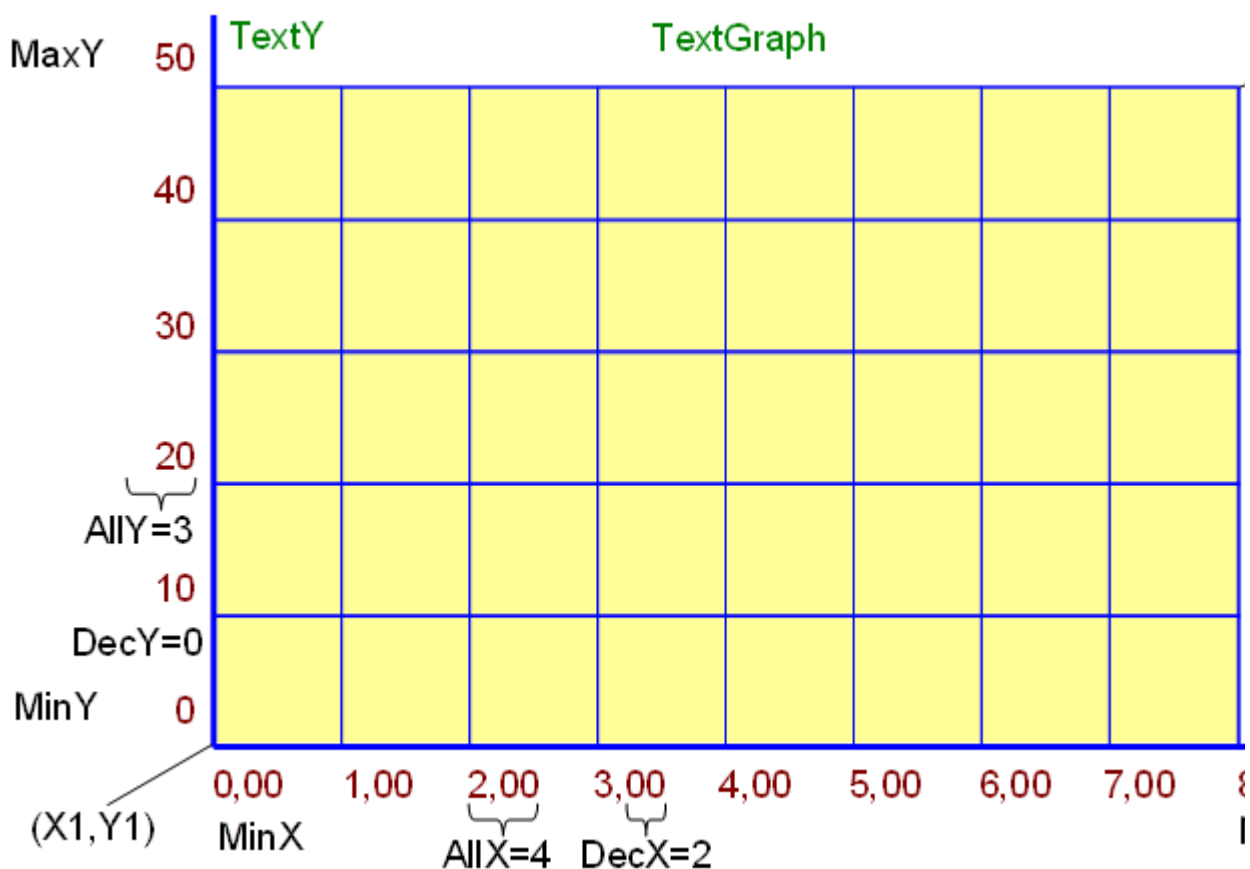
TextGraph - текст заголовку графіка;

TextX, TextY - пояснення до осі абсцис і осі ординат;

TextColor - колір для виведення тексту заголовка і пояснень.

Наступний рисунок ілюструє використання процедури, яка викликана приблизно так

```
BuildCoord(0.15, 0.85, 0.15, 0.25, Yellow, Blue, Red, 9, 4, 2, 6, 3,
0, 0, 8, 0, 50, 'TextGraph', 'TextX', 'TextY', Green);
```



Все, що відображається на рисунку чорним кольором слугує тільки поясненням.

```

procedure BuildChart (X1, Y1, X2, Y2           : real;
                      NumbPoint                : integer;
                      MinX, MinCoordX, MaxX, MaxCoordX,
                      MinY, MinCoordY, MaxY, MaxCoordY : real;
                      ColorChart, LineStyle, Thickness: word;
var ArrX, ArrY           : ArrReal);

```

- процедура виведення графіка. На попередньо виведену сітку графіка (процедурою BuildCoord) зверху виводиться графік, який є сукупністю відрізків, проведених через масив точок.

Параметрами процедури є:

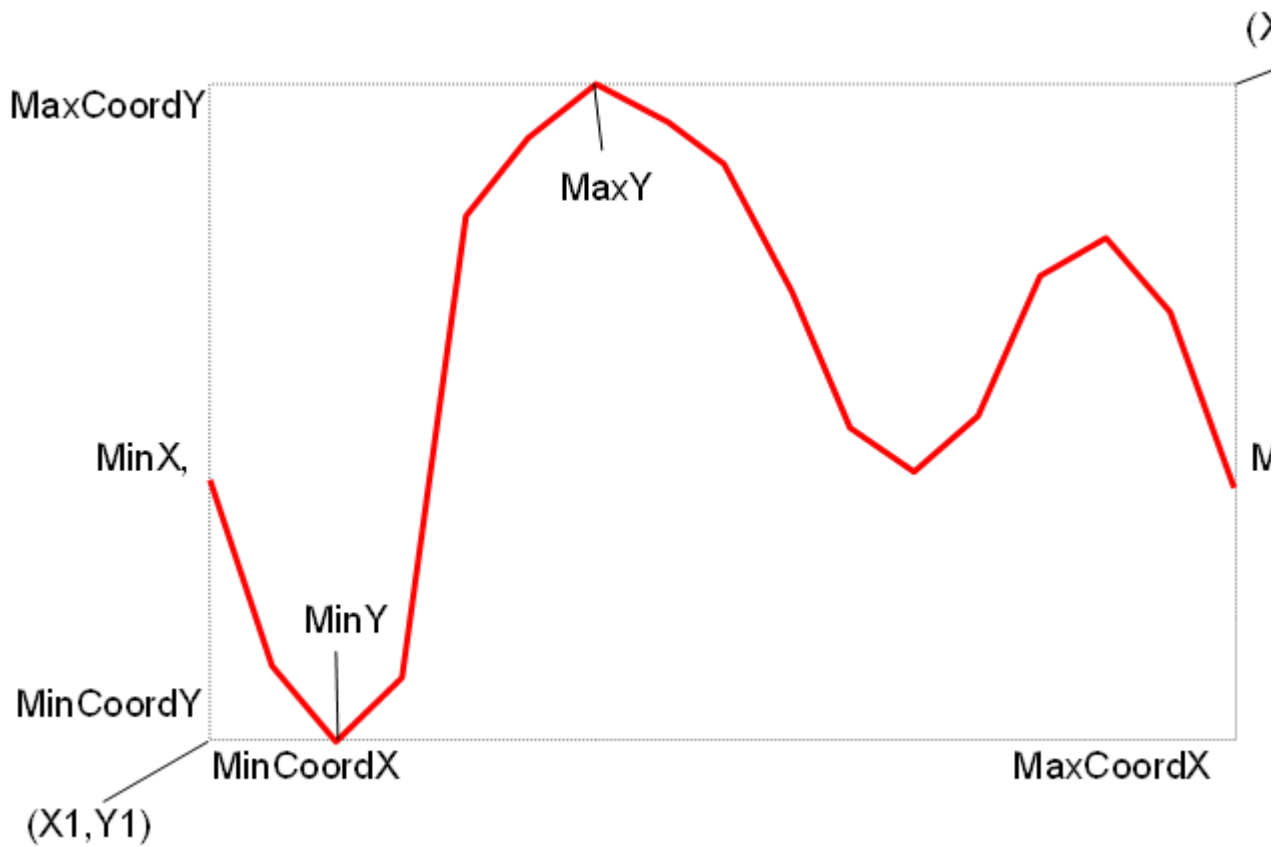
X1, Y1 - координати лівого нижнього кута зони виведення графіка;  
 X2, Y2 - координати правого верхнього кута зони виведення графіка; Повинні повністю співпадати з тими ж параметрами процедури BuildCoord;  
 NumbPoint - кількість вузлових точок графіка для яких визначені масиви числові значення (ціле число не більше NumbOfPoint+1);  
 MinX, MaxX - мінімальне і максимальне значення аргументу. Ці параметри можна передавати як значення і як елементи масиву;  
 MinCoordX, MaxCoordX - мінімальне і максимальне значення осі абсцис. Ці значення повинні співпадати з параметрами MinX, MaxX процедури BuildCoord;  
 MinY, MaxY - мінімальне і максимальне значення функції;  
 MinCoordY, MaxCoordY - мінімальне і максимальне значення осі ординат. Ці значення повинні співпадати з параметрами MinY, MaxY процедури BuildCoord;  
 ColorChart - колір, яким буде виводитись лінія графіка;  
 LineStyle - стиль лінії (суцільна, пунктирна тощо). Цей параметр має значення тільки при невеликій кількості точок, з яких складається графік;  
 Thickness - товщина лінії;  
 ArrX, ArrY - масиви у яких зберігаються відповідні значення аргументу і функції.

Наступний рисунок ілюструє використання процедури, яка викликана приблизно так:

```

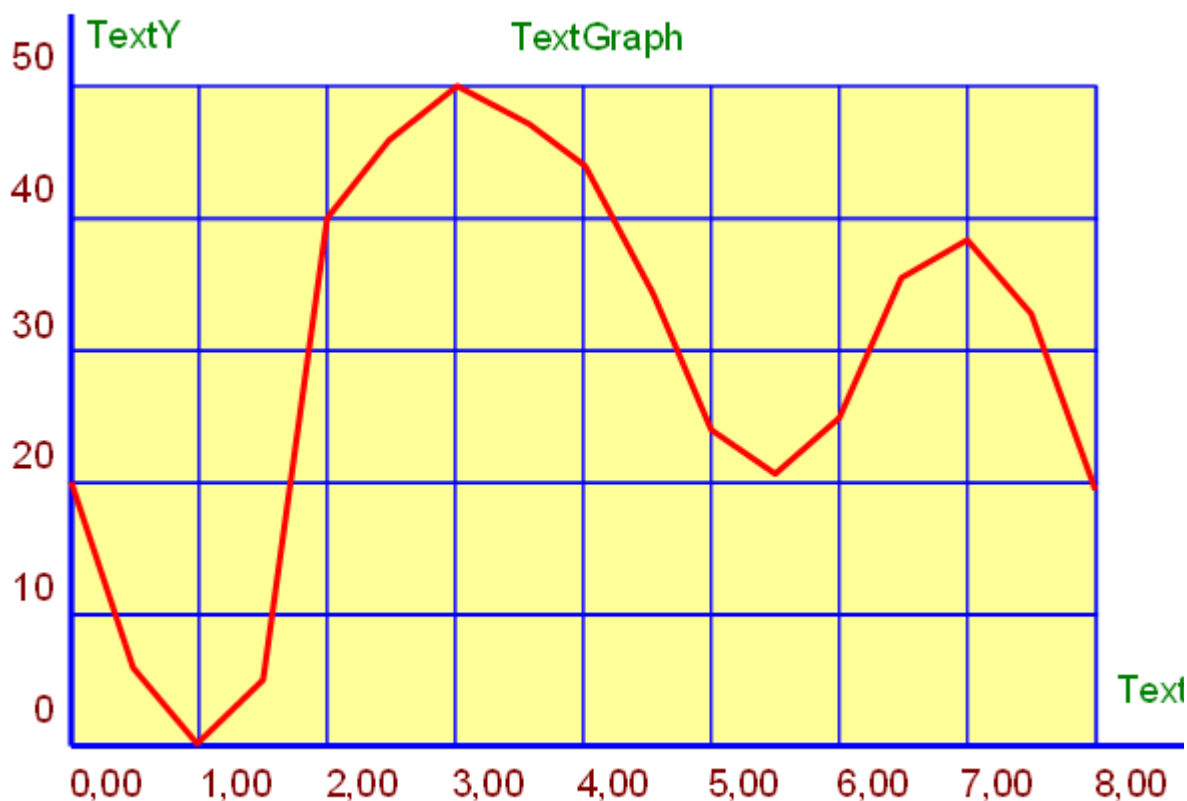
BuildChart(0.15, 0.85, 0.15, 0.25, 18, 0, 8, 0, 8, 0, 0, 50, 50,
Red, SolidLn, 3, ArrX, ArrY);

```



Все, що відображається на рисунку чорним кольором слугує тільки поясненням.

Якщо послідовно викликати дві вищезазначені процедури, то на екрані можна отримати приблизно таке зображення.



## Модуль для проектів Delphi

При створенні проектів Delphi можна скористатися модулем `Eskiz`, який містить процедури виведення окремих ланок механізмів.

Модуль `Eskiz` складається із процедур із тими самими іменами, що й модуль [PVMech](#).

Призначення процедур і схеми повністю відповідають одна одній. Єдина відмінність полягає у тому, що всі процедури модулю `Eskiz` мають першим параметром `Image: TImage` - ім'я канви того елемента, на якому повинні відобразитись відповідні елементи кінематичної схеми.

Далі наведемо лише заголовки процедур:

```
procedure Opora (Image: TImage; x, y, Orient: integer);
```

[\(перейти до однойменної процедури модулю PVMech\)](#)

```
procedure Uzel (Image: TImage; x, y: integer; Name: string);
```

[\(перейти до однойменної процедури модулю PVMech\)](#)

```
procedure RazmerLinear (Image: TImage; x1, y1, x2, y2: integer;
```

```
Text: string);
```

[\(перейти до однойменної процедури модулю PVMech\)](#)

```
procedure RazmerAngular (Image: TImage; x1, y1, x2, y2, x3, y3: integer;
```

```
Text: string);
```

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** PolzunHor (Image:TImage;x,y:integer;osn:boolean);

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** PolzunVer (Image:TImage;x,y:integer;osn:boolean);

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** PolzunAng (Image:TImage;x1,y1,x2,y2:integer);

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** Kulisa (Image:TImage;x1,y1,x2,y2,l:integer);

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** StoykaHor (Image:TImage;x,y,lUp,LDn:integer);

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** NapravHor (Image:TImage;x,y,l:integer);

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** StoykaVer (Image:TImage;x,y,lRt,LLf:integer);

[\(перейти до однойменної процедури модулю PVMech\)](#)

**procedure** NapravVer (Image:TImage;x,y,l:integer);

[\(перейти до однойменної процедури модулю PVMech\)](#)