

ДОДАТОК

Процедури і функції Object Pascal

Процедури і функції для роботи з порядковими даними			
Підпрограми	Тип аргументу	Тип результату	Призначення
Ord (x)	Порядковий	Порядковий	Повертає порядковий номер x.
Pred (x)	Порядковий	Порядковий	Повертає попередній порядковий номер для x.
Succ (x)	Порядковий	Порядковий	Повертає наступний порядковий номер для x.
High (x)	Діапазон	Порядковий	Повертає максимальне значення типу – діапазон, до якого належить x.
Low (x)	Діапазон	Порядковий	Повертає мінімальне значення типу – діапазон, до якого належить x.
Процедури і функції для роботи з цілими і дійсними даними			
Підпрограми	Тип аргументу	Тип результату	Призначення
Abs (x)	Integer, Real	Integer, Real	Повертає модуль x.
Chr (b)	Byte	Char	Повертає символ за його кодом.
Dec (x [,i])	Integer	Integer	Зменшує значення x на i, а при відсутності i на 1.
Inc (x [,i]))	Integer	Integer	Збільшує значення x на i, а при відсутності i на 1.
Hi (w)	Word	Byte	Повертає старший байт аргументу
Lo (w)	Word	Byte	Повертає молодший байт аргументу

Hi (i)	Integer	Byte	Повертає старший байт аргументу
Lo (i)	Integer	Byte	Повертає молодший байт аргументу
Odd (i)	Integer	Boolean	Повертає true, якщо i непарне, інакше – false.
Random (i)	Integer	Integer	Повертає псевдовипадкове число з інтервалу (0, w-1).
Random	Real	Real	Повертає псевдовипадкове число з інтервалу (0, 1).
Randomize	–	–	Ініціалізація генератора псевдовипадкових чисел.
Swap (i)	Integer	Integer	Міняє місцями байти у слові.
Swap (w)	Word	Word	Міняє місцями байти у слові.
Sin (x)	Real	Real	Синус, кут у радіанах.
Cos (x)	Real	Real	Косинус, кут у радіанах.
Arctan (x)	Real	Real	Арктангенс, значення в радіанах.
Exp (x)	Real	Real	Експонента.
Ln (x)	Real	Real	Логарифм натуральний.
Pi	Real	Real	Число π .
Sqr (x)	Real	Real	Квадрат аргументу.
Sqrt (x)	Real	Real	Корінь квадратний.
Frac (x)	Real	Real	Дробова частина числа.
Int (x)	Real	Real	Ціла частина числа.
Процедури і функції для роботи з датами			
Підпрограми		Призначення	
Function Date : TDateTime;		Повертає поточну дату.	
Function Time : TDateTime;		Повертає поточний час.	
СИМВОЛЬНІ РЯДКИ			
Процедури і функції для роботи з символьними рядками			
Підпрограми		Призначення	

Function Concat (S1[,S2, ... SN]: String): String;	Повертає з'єднаний за порядком S1,S2,...,SN символний рядок.
Function Copy (St: String; Index, Count: Integer): String;	Копіює із рядка St, починаючи з символу з номером Index Count символів.
Procedure Delete (St: String; Index, Count: Integer);	Вилучає із рядка St, починаючи з символу з номером Index Count символів.
Procedure Insert (SubSt, St: String, Index: Integer);	Вставляє підрядок SubSt у рядок St, починаючи з символу з номером Index.
Function Pos (SubSt, St: String): Integer;	Повертає номер позиції першого входження підрядка SubSt у рядок St. Якщо підрядок не знайдено, то повертається нуль.
Function Length (St: String): Integer;	Повертає поточну довжину рядка St.
Procedure SetLength (St: String, NewLength: Integer);	Встановлює нову (меншу) довжину рядка St. Якщо NewLength більше за довжину рядка, то звернення до SetLength ігнорується.
Function LowerCase (Const St: String):String;	Повертає початковий рядок St, в якому всі великі латинські букви замінені на малі.
Function UpperCase (Const St: String):String;	Повертає початковий рядок St, в якому всі малі латинські букви замінені на великі.
Function AnsiLowerCase (Const St: String):String;	Повертає початковий рядок St, в якому всі великі букви замінені на малі (з врахуванням кирилиці).
Function AnsiUpperCase (Const St: String):String;	Повертає початковий рядок St, в якому всі малі букви замінені на великі (з врахуванням кирилиці).
Процедури і функції перетворення рядків до інших типів	
Підпрограми	Призначення
Function StrToFloat (St: String): Extended;	Перетворює символи рядка St у дійсне число. Рядок не повинен містити ведучих або ведених пробілів.
Function StrToInt (St: String): Integer;	Перетворює символи рядка St у ціле число. Рядок не повинен містити ведучих або ведених пробілів.

Procedure Val (St: String; Var X; Code: Integer);	Перетворює рядок символів St у внутрішнє представлення цілої або дійсної змінної X, яке визначається типом цієї змінної. При успішному перетворенні параметр Code містить нуль.
Function StrToDate (St: String): TdateTime;	Перетворює символи рядка St у дату. Рядок повинен містити два або три числа, розділених символом “ . ”. Перше число – дата, друге – місяць. Якщо задано третє число – рік, то воно повинно бути у форматі XX або XXXX. Якщо рік не задано, то дата доповнюється поточним роком.
Function StrToTime (St: String): TdateTime;	Перетворює символи рядка St у дату. Рядок повинен містити два або три числа, розділених символом “ : ”. Числа задають години, хвилини і, можливо, секунди. За останнім числом через пробіл можуть іти символи “ am ” “ pm ”, які вказують на 12-годинний формат часу.
Function StrToDateTime (St: String): TdateTime;	Перетворює символи рядка St у дату і час. Рядок повинен містити дату і час, у правильному форматі, розділених пробілом.
Процедури і функції перетворення даних різних типів до символьних рядків	
Підпрограми	Призначення
Function FloatToStr (Value: Extended): String;	Перетворює дійсне значення Value у рядок символів.
Function IntToStr (Value: Integer): String;	Перетворює ціле значення Value у рядок символів.
Procedure Str (X[:Width[:Decimals]]; var St: String);	Перетворює число X будь-якого цілого або дійсного типів у рядок символів. Параметри Width і Decimals визначають формат перетворення. Width – загальна ширина поля символьного представлення, а Decimals – кількість десяткових знаків для дійсних чисел.

Function DateToStr (Value: TdateTime): String;	Перетворює дату із параметра Value у рядок символів.
Function TimeToStr (Value: TdateTime):String;	Перетворює час із параметра Value у рядок символів.
Function DateTimeToStr (Value: TdateTime): String;	Перетворює дату і час із параметра Value у рядок символів.
ДИНАМІЧНІ СТРУКТУРИ ДАНИХ	
Процедури і функції для роботи з динамічною пам'яттю	
Підпрограми	Призначення
Function Addr (x): Pointer;	Повертає адресу аргументу x. Аналогічний результат повертає операція @.
Procedure New (Var P: Pointer);	Резервує фрагмент купи динамічної пам'яті для розміщення змінної і поміщає в типізований вказівник P адресу першого байта.
Procedure Dispose (Var P: Pointer);	Повертає в купу фрагмент динамічної пам'яті, який був зарезервований за типізованим вказівником P.
Procedure GetMem (Var P: Pointer; Size: Integer);	Резервує за нетипізованим вказівником P фрагмент динамічної пам'яті розміру Size (в байтах) і присвоює йому адресу цієї області.
Procedure FreeMem (Var P: Pointer; Size: Integer);	Повертає в купу фрагмент динамічної пам'яті, який був зарезервований за нетипізованим вказівником P.
Function SizeOf (x): Integer;	Повертає довжину в байтах внутрішнього представлення об'єкта x.
ФАЙЛИ	
Процедури і функції для роботи з усіма типами файлів	
Підпрограми	Призначення
Procedure AssignFile (Var F: File; FileName: String);	Зв'язує файлову змінну F з іменем файла FileName,
Procedure Reset (Var F: File [; RecSize: Word]);	Відкриває існуючий файл. Параметр RecSize, має зміст тільки для нетипізованих файлів, вказує розмір блоку даних.
Procedure Rewrite (Var F: File [; RecSize: Word]);	Створює новий файл. Параметр RecSize, має зміст тільки для нетипізованих файлів, вказує розмір

	блоку даних.
Procedure CloseFile (Var F);	Закриває файл, а зв'язок файлової змінної з іменем файла, встановлений раніше процедурою AssignFile , зберігається. При створенні нового або розширенні старого файла процедура забезпечує збереження у файлі всіх нових записів і реєстрацію файла в каталозі. Процедура CloseFile виконується автоматично по відношенню до всіх відкритих файлів при нормальному завершенні програми. Оскільки зв'язок файлової змінної з іменем файла зберігається, то файл можна повторно відкрити без додаткового використання процедури AssignFile .
Function Eof (Var F): Boolean;	Тестує кінець файла, якщо покажчик вказує на кінець файла повертає <i>true</i> , інакше <i>false</i> .
Function IOResult : Integer;	Повертає код завершення останньої операції введення-виведення. При успішному завершенні операції код дорівнює нулеві.
Procedure Flush (Var F);	Очищає внутрішній буфер файла, що гарантує збереження всіх останніх змін файла на диску.
Function FileExists (Const FileName: String): Boolean;	Повертає <i>true</i> , якщо файл з іменем (або маршрутом) FileName існує, інакше <i>false</i> .
Procedure GetDir (D: Byte; Var S: String);	Повертає у змінній S ім'я поточного каталогу. D – номер пристрою (0 – пристрій за замовчуванням, 1 – диск А, 2 – диск В і т. д.).
Procedure ChDir (Path: String);	Змінює поточний каталог. Path – шлях до поточного каталогу, який встановлюється за замовчуванням.
Procedure MkDir (Dir: String);	Створює новий каталог. Dir – шлях до створюваного каталогу.
Procedure RmDir (Dir: String);	Вилучає каталог Dir . Каталог що вилучається повинен бути порожнім.

Procedure Erase (Var F);	Вилучає файл F. Перед вилученням файл потрібно закрити.
Procedure Rename (Var F; NewName: String);	Змінює ім'я файл F. NewName – нове ім'я файла. Перед переіменуванням файл потрібно закрити.
Function DateTimeToFileDate (DateTime: TdateTime): Integer;	Перетворює значення DateTime у системний формат часу створення файла.
Function FileDateToDateTime (FileDate: Integer): TdateTime;	Перетворює системний формат FileDate часу створення файла у формат дата-час.
Function FileGetDate (Handle: Integer): Integer;	За заданим дескриптором файла Handle повертає дату і час його створення у системному форматі.
Function FileSetDate (Handle: Integer; Age: Integer): Integer;	Для файла з дескриптором Handle встановлює нову дату і час Age у системному форматі.
Function FileAge (Const FileName: String; Attr: Integer): Integer;	Для файла FileName повертає час його останнього оновлення (у системному форматі) або -1, якщо такого файла не існує.
Function FindFirst (Const Path: String; Attr: Integer; Var F : TsearchRec): Integer;	Повертає атрибути першого із файлів зареєстрованих у вказаному каталозі: Path – маршрут пошуку і маска вибору файлів; Attr – атрибути файлів, що вибираються; F – змінна типу TsearchRec, в якій повертається ім'я першого вибраного файла.
Function FindNext (Var F : TsearchRec): Integer;	Повертає у змінній F ім'я наступного файла у каталозі. Змінна F повинна попередньо ініціюватися зверненням до функції FindFirst.
Procedure FindClose (Var F : TsearchRec): Integer;	Звільнює пам'ять, виділену для пошуку файлів функціями FindFirst або FindNext.
Процедури і функції для роботи з текстовими файлами	
Підпрограми	Призначення
Procedure Append (Var F: TextFile);	Відкриває текстовий файл для розширення, при цьому покажчик встановлюється на кінець файла.
Procedure Read (Var F:	Читає із текстового файла

TextFile; v1 [,v2, ... ,vn]);	послідовність символів, ігноруючи ознаку EOLN і враховуючи типи змінних v_i . Змінні v_i можуть бути типу Char або String, а також будь-якого цілого або дійсного типів.
Procedure Readln (Var F: TextFile; v1 [,v2, ... ,vn]);	Читає із текстового файла послідовність символів з урахуванням ознаки EOLN і типу змінних v_i . Змінні v_i можуть бути типу Char або String, а також будь-якого цілого або дійсного типів.
Procedure Write (Var F: TextFile; p1 [,p2, ... ,pn]);	Записує в символьній формі параметри $p_1, \dots, p_n$ у текстовий файл.
Procedure Writeln (Var F: TextFile; p1 [,p2, ... ,pn]);	Записує в символьній формі параметри $p_1, \dots, p_n$ і ознаку кінця рядка EOLN у текстовий файл.
Function Eoln (Var F: TextFile); boolean;	Тестує маркер кінця рядка. Якщо досягнутий кінець рядка повертає <i>true</i> , інакше <i>false</i> .
Function SeekEoln (Var F: TextFile); boolean;	Тестує маркер кінця рядка, пропускаючи всі пробіли або знаки табуляції до признаку EOLN або до першого символу. Якщо досягнуто кінець рядка повертає <i>true</i> , інакше <i>false</i> .
Function SeekEof (Var F: TextFile); boolean;	Тестує маркер кінця файла, пропускаючи всі пробіли, знаки табуляції і маркери кінця рядка EOLN, до маркера кінця файла EOF або до першого символу. Якщо досягнуто кінець файла повертає <i>true</i> , інакше <i>false</i> .
Процедури і функції для роботи з типізованими файлами	
Підпрограми	Призначення
Procedure Read (Var F; v1,v2, ... ,vn);	Читає (без перетворення) дані із типізованого файла F. V_i – змінні того ж типу, що і компоненти файла.
Procedure Write (Var F; p1,p2, ... ,pn);	Записує (без перетворення) дані у типізований файл F. P_i – вирази того

	ж типу, що і компоненти файла.
Procedure Seek (Var F; N: Longint);	Встановлює покажчик файла F на N-ий компонент файла (перший компонент файла має номер 0).
Function FileSize (Var F): Longint;	Повертає кількість компонентів файла.
Function FilePos (Var F): Longint;	Повертає поточну позицію покажчика в файлі, тобто номер компонента, який буде оброблятися наступною операцією введення-виведення.
Процедури і функції для роботи з нетипізованими файлами	
Підпрограми	Призначення
Procedure BlockRead (Var F:File; Var Buf; Count: Integer [; Var AmtTransferred: Integer]);	Читає (без перетворення) дані із нетипізованого файла F. Buf – ім'я змінної, яка приймає участь в обміні даними. Count – кількість записів, які повинні бути прочитані за одне звернення. AmtTransferred – необов'язковий параметр, який містить при виході із процедури кількість фактично оброблених записів.
Procedure BlockWrite (Var F:File; Var Buf; Count: Integer [; Var AmtTransferred: Integer]);	Записує (без перетворення) дані у нетипізований файл F. Buf – ім'я змінної, яка приймає участь в обміні даними. Count – кількість записів, які повинні бути записані за одне звернення. AmtTransferred – необов'язковий параметр, який містить при виході із процедури кількість фактично оброблених записів.

ЛІТЕРАТУРА

1. Сурков К.А., Сурков Д.А Программирование в среде DELPHI 2.0. – Мн.: Попури, 1997. – 640 с.
2. Епанешников А., Епанешников В. Программирование в среде DELPHI 2.0. Учебное пособие: В 4-х частях. – М.: ДИАЛОГ-МИФИ. 1998.– 1236 с.
3. Фаронов В.В. Delphi 4. Учебный курс. М.: Нолидж, 1998.-464 с.
3. Фаронов В.В. Delphi 3. Учебный курс. М.: Нолидж, 1998.-400 с.
4. Р. Баас, М. Фервай, Х. Гюнтер. Delphi 5: для пользователя – К. Издательская группа BNV, 2000.–496 с.
6. П.В.Шумаков, Фаронов В.В. Delphi 4. Руководство разработчика баз данных. – М.: Нолидж,1999.-476 с.
7. Калверт Ч. Delphi 2. Энциклопедия пользователя: Пер. с англ. – К.: НИПФ ДиаСофтЛтд, 1996.– 736 с.