Езици за програмиране

1. Групи езици за програмиране

а) езици от ниско ниво – всеки процесор който управлява даден компютър разглежда числовите данни като управляващи команди и данни. Управляващите команди се наричат машинен код (машинен език). Всяка операция в дадения процесор си има числов еквивалент наречен код на операцията. Всяко поколение процесори има различни кодове за операции. Поради тази причина машинния код се нарича машинно зависим език за програмиране. За улеснение на програмистите кодовете на операции биват заместени с буквени означения. Това се нарича език assembler. Тъй като е необходимо преобразуването в машинен код се налага преминаването през заместваща програма наречена assembler. Основен недостатък на асемблера е че той също се съобразява с решените задачи на процесора

б) Езици от високо ниво – езици близки до човешкия език (английски) които не се съобразяват с конкретен процесор. За тези езици се изискват специални превеждащи програми които да генерират адекватен машинен код. Тези програми се наричат транслатори.  
Съществуват два вида транслатори при първия всеки ред програмен код се превежда и веднага се изпълнява, наричат се интерпретатори.  
Вторият начин е целия код се превежда и съхранява в машинен код, след което може многократно да бъде изпълняван, наричат се компилатори. Интерпретатора не изисква допълнителна памет за съхранение.

в) Процес на компилиране – класическия процес на компилация преминава по следния път -   
- писане на програмата на съответния език от високо ниво source code   
- компилиране на програмата – в резултат на тази стъпка се получава т.нар. обектен код (.обу). този код има две характеристики не съдържа външни библиотечни обекти и няма специфичен код за изпълнение в конкретна операционна среда.  
- процес на свързване – в този процес към един или няколко обектни файла се добавят библиотечните елементи и специфичния за дадена операционна система код. В резултат се получава изпълним файл.

В съвременните операционни системи в един и същи момент се изпълняват няколко програми. Голяма част от тях използват едни и същи библиотечни елементи. Вместо тези елементи да се намират във всеки изпълним файл статично се използват динамични библиотеки, които се свързват след стартиране на изпълнимия файл. Динамичните библиотеки имат разширение DLL, те трябва да присъстват по време на изпълнение на програмата. При интернет програмиране не се знае нито хардуера нито операционната система на която ще се изпълнява програмата. Има два подхода за решаване на този проблем първият е използване на интерпретатор, вторият е да се компилира кода за виртуална операционна система и платформа и да се до компилира кода при клиента. Компилирането при самия клиент се извършва от специални компилатори в реално време. Най-популярна е JAVA виртуалната машина.

1. Групи езици за програмиране   
   а) Процедурни езици за програмиране – това са езици състоящи се от данни (променливи) и програмен код обработващ тези данни.  
   б) Обектно ориентирани езици за програмиране – при тях данните и кода са обединени в едно цяло.  
   в) Визуални езици за програмиране – това са езици автоматизиращи процеса на изграждане на потребителския интерфейс   
   г) Езици с междинна (виртуална) платформа – върху операционната система се инсталира междинен слой наречен виртуална машина. Програмите се изпълняват използвайки тази машина. Visual Basic .net, c sharp са същите като java.   
   д) Интерпретаторни (скриптинг) -   
   е) Езици които се изпълняват на web сървъри – служат за създаване на динамични web страници.   
   ж) специфични езици за програмиране за програмни продукти  
   Visual Basic for Application за офис пакета  
   Action Script за Adobe
2. Основни понятия в обектно ориентираното програмиране (ООП) -   
   а) Class – класът представлява съвкупност от свойства (характеристики) и действия на обект от реалния свят. Тези неща са капсуловани   
   б) Object – конкретен представител (инстанция) на даден клас  
   в) Property – представлява конкретна характеристика на даден клас. Обектите имат стойности за свойствата   
   г) Метод (method) – конкретно действие изпълнявано от класа и обектите в класа.  
   д) Събития (Event) – това са действия които се активират от други обекти

Език за програмиране Visual Basic .net

1. Версии на езика Basic – в настоящия момент съществуват следните версии на езика Basic от фирмата Microsoft. Това е версия на Basic която създава изпълними файлове които нямат нужда от междинна платформа или среда за изпълнение. VB.NET (2012) това е език или версия която използва .net framework. VB script – това е интерпретаторна версия на езика Basic. VB for application (VBA).
2. Азбука на езика – използват се латинските букви и някои специални знаци
3. Имена на обектите във VB.net - обектите в една програма биват:  
   - оператори представляващи действията върху самите данни или в/у изпълнението на програмата. Операторите са предефинирани в езика и потребителят неможе да създава дублиращи имена. Потребителски обекти – това са имена на променливи, константи, подпрограми. Тези имена трябва да отговарят на следните условия   
   - името трябва да започва с буква или подчертаващо тире  
   - името може да съдържа букви, цифри и подчертаващо тире  
   - ! езикът неправи разлика между малки и главни букви  
   - максималната дължина на едно име е 256 знака  
   Препоръчителни правила  
   - имената да са значещи   
   - за разделяне на думите в сложно име се използват или подчертаващи тирета или смесване на малки и главни букви   
   - в името на обекта да се постави и неговият тип (клас)   
   Функции на езика VB.net – служат за обработка на данните и връщане на определен резултат. Те също са част от езика VB.net   
   Готови класове от .net –
4. Структура на програмата на VB.net VS дефинира следните правила за изграждане на програма. Самата програма се нарича Solution (решение). Едно решение може да се състои от един или няколко проекта. Проектите могат да бъдат на различни езици за програмиране или проекти за инсталиране на решението на друг компютър. Структурата на един файл във VB .net е следната   
   - 1 не е задължителен оператор option   
   - 2 не е задължителни оператори Import  
   - 3 не е задължителен Namespace   
   - 4 един от операторите Class,Module, Structure, Interface   
   - 5 Function, SUB, оператори, Event  
   Вътре в тези блокове се разполагат съответните оператори, функции и потребителски обекти. Операторите се отделят или чрез знака за нов ред или чрез две точки. Ако трябва да се раздели дълъг оператор на няколко реда се използва интервал и подчертаващо тире. Забранява се прекъсването на заграден в кавички текст.
5. Коментари в програмния код – коментарите са незадължителен елемент улесняващ четящия програмния код. VB.net игнорира този текст. Оператор REN или ‘. Коментарът може да се използва за временно деактивиране на части от кода.

Типове данни

Всеки език за програмиране разполага с ясно дефинирана система от типове на данните които може да обработва. Съществуват езици при които автоматично се определя типът на данните. Във VB.net типовете данни трябва да се съобразят с типовете на .net

1. Целочислени типове данни – характеризират се с два параметъра. Колко байта оперативна памет заемат и дали са разрешени отрицателни числа.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип | Байт | Знак | Min | Max |
| Byte | 1 | не | 0 | 255 |
| Sbyte | 1 | да | -128 | 127 |
| Ushoat | 2 | не | 0 | 65535 |
| Short | 2 | да | -32768 | 32767 |
| Uinteger | 4 | не | 0 | 4297967295 |
| Integer | 4 | да |  |  |
| Ulong | 8 | не | 0 |  |
| Long | 8 | да |  |  |

1. Дробни числа - поради спецификата на съхранение дробните числа винаги имат определена точност (брой знаци след десетичната точка). Съществуват два типа -   
   Сингъл – заема 4 байта и има 8 знака точност   
   Double – заема 8 байта и има 16 знака точност  
   Съществува специален тип данни с обща дължина 28 знака.   
   Decimal – 16 байта 28 знака този тип е най-бавният тип данни и се препоръчва неговото използване при изключителни случай и сложни математически изчисления
2. Логически тип – това е тип съхраняващ две стойности истина и лъжа   
   Boolean.
3. Тип дата и час – от 00 : 00 на 01. 01.01 до 23:59 на 31.12.9999
4. Текст и символи – Char може да съхранява един символ   
   Code – за кодировка се използва Unicode (2b)  
   String – поредица от символи може да се съхранява „а,б,в”
5. Универсален тип данни – Object –

Константи

Константата е обект който не променя своята стойност по време на изпълнение на програмата.

1. Константи литерали – това е стойност директно записана в програмния код   
   а) Литерали цели числа – целите числа могат да се записват в десетична, осмична, и шестнадесетична бройна система. За осмична та бройна с/ма се използва префикса &, а за шестнадесетична &H.   
   Важно е да се знае типът на константата – в този случай се използват специални суфикси. Пример S – short US – Ushoat , I – Integer(% - непроменя стойността) , UI – Uinteger , L – Long , UL – Ulong

б)Дробни литерали – съществуват два начина за представяне -   
- Класически 10.2   
- Научен формат – използва се експоненциална стойност   
Използват се следните суфикси F – single – може да се замени с !, R (#) – double, D (@) – decimal

в) Дата и час загражда се в ## пр. #12:05# #26/10/12#

г) Символи и низове „!“ c- Char, string „АБЦ“ „ ! “   
д) Логически литерали – True False

1. Именувани константи – не се препоръчва използването на литерали в програмният код. Препоръчва се деклариране на имена на константите които да се използват в програмния код.   
   Const име [AS тип] = c – cm
2. Сложни константи – това са константи които обединяват повече от една целочислена стойност.   
   Enum име   
    име 1 [= c – cm]  
    име 2 [= c – cm]   
   end enum   
   Ако не се посочи стойност към стойността на предходния елемент се добавя единица.  
   Ако първият елемент няма стойност стойността е 0.

Променливи

Променливата е обект които може да съхранява различни стойности по време на изпълнение на програмата. Заемат определено място в паметта на компютъра в зависимост от техния тип.

1. Декларация на променливата – DIM име на променлива 1,2 – AS тип [= c.cm – начална стойност]. DIM A, B as Byte. Начална стойност може да се посочва само на единична променлива.
2. Масиви – масивът е структура при която всички елементи са от един и същи тип. Всички елементи имат номер наречен индекс на елемента.   
   DIM M1 (10) – номер на последният елемент AS Byte – декларира се масив от 11 елемента от тип Byte.   
   DIM M2 (2) AS Long = {20, 30, 40} – декларира масив от три елемента с начални стойности 20, 30, 40.   
   DIM M3 (2, 3) AS Double – Декларира матрица с три реда и четири колонки. Двумерен масив.   
   DIM M4 (1, 2) AS Byte = {{1, 2, 3}, {4, 5, 6} }   
   DIM M5 (1, 2, 3) AS Byte = {{1, 2, 3}, {4, 5, 6}}  
   Съществува възможност да се декларира масив без да се заделя памет за него. Това се извършва чрез оператора Redim – позволяващ да се променя размера на масив по време на изпълнение.   
   Redim [preserve] име (нов размер). Ако не е посочено preserve се губят старите стойности на масива.   
   DIM M6 () AS Boolean   
   Redim M6 (99)   
   M(0) = true   
   M(1) = false   
   Redim preserve M6 (199) – 200 елемента като запазваме старите стойности
3. Сложни типове данни – това е тип които не заема място в паметта. Може да бъде използван за декларация на променливи или масиви.   
   Structure име   
    DIM поле1 AS тип   
    DIM поле2 AS тип   
   End structure   
   Structure Grad   
   DIM Oblast AS String   
   DIM IME AS String   
   DIM PK AS UShort  
   END Structure   
     
   Structure Student   
    DIM IME AS String   
    DIM SU AS Double   
    DIM Mestor AS Grad   
    DIM Mestoj AS Grad   
   END Structure   
     
   DIM P1 AS Grad   
   P1.PK = 5250  
   P1.Ime = „Свищов“  
   DIM P2 AS Student   
    P2.Ime = „Иван“   
   P2.mestor.Ime = „Свищов“

Условни оператори

Оператори които позволяват програмния код да се разклони в зависимост от някакви условия

1. Едноредов оператор IF – IF условие THEN оператор 1 [ ELSE оп 1,2]. Ако условието е истина се изпълняват операторите в THEN частта, ако е лъжа се изпълнява ELSE частта.

Да се намери по малкото от две числа   
IF a<b THEN min = a ELSE min = b

1. Многоредов IF -   
   IF условие THEN оп1:оп2   
   ELSEIF условие THEN   
   оп1:оп2  
   оп3  
   -----  
   -----  
   ELSE   
    оп  
    оп  
   END IF

Да се намери по малкото от две числа  
IF A<B THEN  
 MIN = A  
ELSE   
 MIN = B   
END IF

Променливата S съдържа сума на продажбите. Променливата О трябва да съдържа отстъпката която се изчислява по следната таблица

IF S < 10000 THEN   
 O=0

ELSEIF S < 100 000 THEN   
 O = 0, 02

ELSEIF S < 1 000 000 THEN  
 O = 0, 03

ELSE   
 O = 0, 04

END IF

Променливата М съдържа номера на месеца. Да се запише в променливата S сезона

IF (M = 3) or (M = 4) or (M = 5) THEN  
IF (M>=3) and (M<= 5)THEN   
S = “Пролет“   
ELSEIF (M > = 6) AND (M < = 8) THEN  
S = „Лято“  
ELSEIF (M > = 9) AND (M < = 11) THEN   
S = “ Есен“   
ELSEIF (M = 1) or (M = 2) or (M = 12) THEN  
S = „ЗИМА“  
ELSE

S = “Грешка “   
END IF

Компонента RBJENA съдържа информация дали пола на даден човек е жена. Да се запише пола в променлива форма. Когато свойството е логически тип проверката е излишна

1. Оператор Select – това е условен оператор които се базира на стойността на един израз или променлива. Стойността на израза се проверява последователно за CASE блоковете. Изпълнява се първият блок които отговаря на условието, след което се излиза от Select оператора.

Стойностите се задават по три начина   
- списък от стойности – 1,2,3  
- затворен интервал от стойности -1 tu 10   
- отворен интервал – IS > 10 IS < = 5

Select Case M  
 Case 3 to 5   
 S = “Prolet”  
 Case 6 to 8  
 S = “Lqto”  
 Case 9 to 11   
 S = “Esen”  
 Case 1,2,12   
 s = “Zima”  
 Case ELSE   
 S = “Грешка“   
END Select

Оператори за цикъл

Цикълът представлява един или няколко оператора наречени тяло на цикъла които могат да се изпълнят повече от един път.

1. Цикъл с точен брой повторения -   
   FOR пром [ AS тип ] = начална с-ст To крайна с-ст [Step с-ст]   
    Body   
   [ Exit FOR ]   
   Next   
   - ако променливата не е декларирана предварително трябва да се декларира с AS частта   
   - управляващата променлива последователно приема стойности от началната до крайната, и за всяка от тях се изпълнява тялото на цикъла  
   - Ако е пропусната STEP частта управляващата променлива се увеличава с 1   
   - Стъпката може да бъде и отрицателна  
   - Променливата може да участва в тялото на цикъла, но нетрябва да се променя нейната стойност

Да се отпечата върху екрана даден текст 10 пъти

For BR AS Byte = 1 to 10   
 MSG BOX (“HELLO”)  
Next

Да се отпечатат четните числа от 1 до 100

For BR AS Byte = 2 to 100 Step +2

MSG BOX (BR)

Next

Д

Да се намери сумата на числата от 1 до 100

Dim Suma AS Short   
Suma = 0

For BR AS Byte = 1 to 100

Suma = Suma + BR

Next

Даден е масив Х които съдържа N цели числа. Да се намери тяхната сума.   
Dim Suma AS Ulong = o

For BR AS Ulong = 0 to N – 1

Suma += X (BR)

Next

Функции за работа с текст

1. Len – връща дължината на текста Len(“Иван“) > 4  
   Len (tbIme.Text)  
   a) Функции за извличане на части от текста  
    Left(text, брой знаци)  
   Right(текст, брой)  
   Mid(текст, старт,брой)  
   left (“ABCD”, 2)  
   Right(“ABCD”, 2)  
   Mid(“ABCD”, 2,1) – номерацията започва от 1, от коя позиция колко знака взема  
   MID (Right(x,2),1,1)  
   Left (Right(x,2),1)  
   MID (x, LEN(x) – 1, 1)  
   Съществува свойство Left, което се подразбира в редактора. Затова се налага функцията Left да бъде по-точно дефинирана Microsoft.VisualBasic. Left
2. Търсене на текст  
   InStr ([старт], къде, какво, [как сравняваме]) – начинът на сравнение може да бъде binary (подразбира се), при който се прави разлика между малки и главни букви. Text – не се прави разлика между малки и главни букви  
   InStr (“ABCD”, “B”) 2  
   InStr (3, “ABCDABCDB”, “B”) 6   
   InStr(2, “ABC”, “A”) 0   
   InStr( “ABC”, “a”) 0   
   InStr( “ABC”, “a”, Text) 1  
   InStr( “ABa”, “a”) 3  
   Функцията връща номера на първото срещане на намерения текст или 0 ако текста не е намерен.   
   InStrRev (къде, какво, [старт], [сравнение]) – извършва търсене от дясно на ляво. Началната позиция може да бъде отрицателно число  
   Променливата х съдържа трите имена на човека. Да се извади само първото име на човека  
   InStr (x,” “)  
   MID (x,1,InStr(x,” “) -1)   
   Б) Подмяна на части от текста –   
   Replace (къде, какво, с какво, [старт], [брой], [сравнение])  
   Replace ( “ABCA”, “A”, “\*”) \*BC\*  
   Replace ( “ABCABCABC”, “A”, “\*”, 2,1) “ABC\*BCABC”  
   В) Разделяне и обединяване на текст  
   Split ( какво, [разделител – по подразбиране „ „ ], [брой], [сравнение]) връща масив съдържащ частите на текста спрямо разделителя  
   Dim M() As String  
   М = Split (“ABC XYZ KLM”)   
   M(0) = “ABC”  
   M(1) = “XYZ”  
   M(2) = “KLM”  
   M = Split (“ABC KLM”)   
   M(0) = “ABC”  
   M(1) = “”  
   M(2) = “”  
   M(3) = “KLM”  
   г) Други функции -   
   LCase – преобразува текста в малки букви (Т)  
   UCase – преобразува текста в главни букви (Т)  
   LTrim – премахва интервалите от ляво на текста (Т)  
   RTrim – премахва от дясно интервалите (Т)  
   Trim – премахва от ляво и от дясно (Т)  
   StrReverse – обръща текста отзад напред (Т)  
   StrDup – повторение на даден символ определен брой пъти (число, „символ“)   
   Променливата Х съдържа една дума да се покаже думата с първа буква главна и останалите малки  
   Х = LCase (x)   
   X = UCase(MID(x,1,1)) & MID (x,2,Len (x) -1)  
   Променливата ЕГН съдържа ЕГН във вид на текст. В променливите   
   d, m, g да се получат денят, месеца и годината във вид на числа  
   Dim d, m As Byte  
   Dim g As Long /(short)   
   Dim EGN As String   
   d = CByte (MID(EGN, 5, 2 ))  
   m = CByte(MID(EGN, 3, 2))  
   g = CLng (MID (EGN, 1,2))
3. Функция за генериране на псевдо случайни числа  
   Числата при компютрите се генерират чрез формула, за да се добави случаен елемент трябва да се изпълни оператора Randomize   
   RND () – генерира случайно число между 0 и 1. За генериране на цяло число в определени граници (мин. Макс. ) се използва следната формула INT((max – min + 1 ) \* RND() + min)