

بسمه تعالی

دانشگاه ملی مهارت

(دانشکده امام خمینی قاین)

گروه مهندسی مکانیک (ساخت و تولید / ماشین افزار)

جزوه آموزشی و پروژه محور:

برنامه نویسی رایانه (کاربرد VBA در محاسبات ماشین کاری و CNC)

مدرس:

رضاپور

وبسایت مرجع و پشتیبانی:

www.Hrezapour.ir

نیمسال تحصیلی ۱۴۰۳-۱۴۰۴

کتابچه آموزشی برنامه‌نویسی رایانه (پروژه محور)

محیط توسعه: Microsoft Excel (VBA)

مقدمه مؤلف: پیوند ماشین کاری و برنامه‌نویسی

در دنیای مدرن مهندسی ساخت و تولید، دوران محاسبات دستی و تنظیمات سنتی ماشین آلات به سر آمده است. امروزه دستگاه‌های CNC (کنترل عددی کامپیوتری)، روبات‌های جوشکاری و سیستم‌های اندازه‌گیری CMM تماماً بر اساس کدهای برنامه‌نویسی کار می‌کنند. نرم‌افزارهای قدرتمند CAD/CAM مانند SolidWorks، Mastercam و PowerMill نیز در هسته مرکزی خود چیزی جز هزاران خط کد برنامه‌نویسی نیستند که معادلات پیچیده هندسی را حل می‌کنند.

هدف از این درس، تبدیل کردن شما به یک مهندس نرم‌افزار نیست؛ بلکه هدف این است که یاد بگیرید چگونه به عنوان یک «مهندس مکانیک»، از ابزار برنامه‌نویسی برای حل سریع‌تر، دقیق‌تر و ارزان‌تر مشکلات کارگاه یا کارخانه خود استفاده کنید.

ما در این دوره از محیط Excel VBA استفاده می‌کنیم. اکسل نرم‌افزاری است که در هر کارگاهی یافت می‌شود و زبان برنامه‌نویسی پشت آن (Visual Basic for Applications) به حدی قدرتمند است که می‌تواند پیچیده‌ترین محاسبات سینماتیک، طراحی قید و بند، انتخاب ابزار برشی و حتی تولید مستقیم G-Code برای دستگاه CNC را به صورت خودکار انجام دهد.

فصل اول: مبانی منطق، الگوریتم و فلوچارت

پیش از آنکه دست به کیبورد ببریم و کدی بنویسیم، باید یاد بگیریم چگونه مانند یک کامپیوتر فکر کنیم. کامپیوترها به شدت سریع، اما کاملاً بی منطق هستند. آنها فقط دستوراتی را اجرا می کنند که ما با دقت و مرحله به مرحله به آنها داده باشیم.

۱-۱. الگوریتم چیست؟

الگوریتم (Algorithm) مجموعه ای از دستورالعمل های متناهی، دقیق و بدون ابهام است که برای حل یک مسئله خاص طراحی شده است.

در مهندسی ساخت، ما هر روز از الگوریتم ها استفاده می کنیم. به عنوان مثال، “دستورالعمل راه اندازی دستگاه تراش” یک الگوریتم است:

۱. شروع

۲. بررسی سطح روغن دستگاه.

۳. بستن قطعه کار در سه نظام و آچار کشی.

۴. انتخاب ابزار مناسب و بستن روی تارت.

۵. روشن کردن اسپیندل با دور مناسب.

۶. نزدیک کردن ابزار به قطعه و مماس کردن (Zero Point).

۷. پایان.

ویژگی‌های یک الگوریتم خوب:

- نقطه شروع و پایان مشخص: هیچ برنامه‌ای نباید تا بی‌نهایت در خلأ رها شود.
- دقت: دستور «کمی بار بده» برای کامپیوتر بی‌معنی است. باید بگوییم «مختصات Z را ۲ میلی‌متر کاهش بده».
- ترتیب منطقی: نمی‌توان قبل از روشن کردن اسپیندل، باربرداری را شروع کرد.

۲-۱. مثال مهندسی: الگوریتم محاسبه سرعت برشی

فرض کنید می‌خواهیم برنامه‌ای بنویسیم که قطر قطعه کار و دور بر دقیقه (RPM) را بگیرد و سرعت برشی (Cutting Speed) را حساب کند.

فرمول ریاضی:

$$V_c = \pi \times D \times N \quad V_c = \frac{\pi \times D \times N}{1000} \quad V_c = 1000 \times \pi \times D \times N$$

در این فرمول V_c سرعت برشی (متر بر دقیقه)، D قطر قطعه (میلی‌متر) و N سرعت دوران (دور بر دقیقه) است.

الگوریتم برنامه:

۱. شروع.

۲. مقدار عدد پی (π) را برابر ۳.۱۴۱۵ قرار بده.

۳. مقدار قطر (DDD) را از کاربر دریافت کن.

۴. مقدار دور (NNN) را از کاربر دریافت کن.

۵. حاصل ضرب $N \times D \times \pi \times D \times \pi \times N \times D \times \pi$ را محاسبه کن و بر ۱۰۰۰ تقسیم کن.

۶. جواب را در متغیری به نام VcV_cVc ذخیره کن.

۷. مقدار VcV_cVc را روی صفحه نمایش بده.

۸. پایان.

۳-۱. فلوجارت (روندنما)

فلوجارت نمایش تصویری الگوریتم است. برنامه‌نویسان حرفه‌ای قبل از نوشتن پروژه‌های بزرگ، ابتدا فلوجارت

آن را رسم می‌کنند. در فلوجارت از اشکال هندسی استاندارد استفاده می‌شود:

بیضی: برای شروع و پایان.

متوازی‌الاضلاع: برای دریافت اطلاعات (ورودی) یا نمایش اطلاعات (خروجی).

مستطیل: برای انجام محاسبات ریاضی و عملیات اجرایی.

لوزی: برای شرط‌ها و تصمیم‌گیری‌ها (آیا قطر از ۵۰ بیشتر است؟ بله/خیر).

(تمرین کلاسی: دانشجویان فلوجارت فرآیند قلاویزکاری دستی را با استفاده از اشکال هندسی رسم کنند).

فصل دوم: آشنایی با محیط Excel VBA و طراحی رابط کاربری (UI)

اکنون که با منطق حل مسئله آشنا شدیم، وارد محیط نرم‌افزار می‌شویم. ما قصد داریم به جای استفاده از سلول‌های ساده اکسل، نرم‌افزارهای کوچک با پنجره‌های اختصاصی تولید کنیم.

۱-۲. فعال‌سازی تب Developer (توسعه‌دهنده)

به صورت پیش‌فرض، منوی برنامه‌نویسی در اکسل پنهان است تا کاربران عادی کدهای سیستمی را تغییر ندهند. برای فعال‌سازی آن:

۱. نرم‌افزار Excel را باز کنید.

۲. از منوی File به بخش Options بروید.

۳. در پنجره باز شده، از سمت چپ گزینه Customize Ribbon را انتخاب کنید.

۴. در لیست سمت راست، تیک گزینه Developer را فعال کنید و OK را بزنید.

[در اینجا اسکرین‌شات از پنجره تنظیمات اکسل و تیک Developer قرار داده شود]

۲-۲. ورود به آزمایشگاه برنامه‌نویسی (VBA IDE)

حالا در منوی بالای اکسل تب Developer را می‌بینید. روی آیکون Visual Basic کلیک کنید (یا کلیدهای میانبر $Alt + F$ را فشار دهید).

پنجره‌ای باز می‌شود که به آن IDE (محیط توسعه یکپارچه) می‌گویند. این محیط کارگاه ماشین‌کاری ما در دنیای کامپیوتر است!

بخش‌های مهم محیط VBA:

پنجره Project Explorer: (سمت چپ بالا) مانند درختواره فایل‌هاست. شیت‌های اکسل و فرم‌هایی که می‌سازیم اینجا قرار می‌گیرند.

پنجره Properties (خصوصیات): (سمت چپ پایین) یکی از مهم‌ترین بخش‌هاست. هر چیزی که در برنامه می‌سازیم (دکمه، فرم، کادر) یک سری خصوصیات دارد مثل رنگ، اندازه و نام.

محیط Code Window: (پنجره بزرگ وسط) جایی که کدهای برنامه را تایپ می‌کنیم.

۲-۳. ساخت اولین فرم (UserForm)

ما می‌خواهیم پنجره‌ای بسازیم که کاربر بتواند اعداد را در آن وارد کند.

۱. در محیط VBA، از منوی بالا روی Insert کلیک کرده و UserForm را انتخاب کنید.

۲. یک پنجره خاکستری رنگ ظاهر می‌شود. همچنین جعبه ابزاری به نام Toolbox نمایش داده می‌شود.

[در اینجا اسکرین‌شات از یک UserForm خالی و Toolbox در کنار آن قرار داده شود]

معرفی ابزارهای Toolbox:

Label (برچسب - نماد A): برای نوشتن متن‌های راهنما. کاربر نمی‌تواند این متن را تغییر دهد. (مثلاً نوشتن

کلمه "شعاع را وارد کنید").

TextBox (جعبه متن - نماد ab): کادری سفید رنگ که کاربر نرم افزار، اعداد یا حروف را داخل آن تایپ می کند. این پل ارتباطی ما با کاربر است.

CommandButton (دکمه فرمان - نماد مستطیل برجسته): دکمه ای که کاربر پس از وارد کردن اعداد روی آن کلیک می کند تا محاسبات آغاز شود.

۴-۲. اصول نام گذاری استاندارد (بسیار مهم)

وقتی یک TextBox روی فرم می گذارید، کامپیوتر به صورت خودکار نام آن را TextBox ۱ می گذارد. اگر ۱۰ کادر داشته باشیم، نامها TextBox ۱ تا TextBox ۱۰ خواهند بود. هنگام برنامه نویسی گیج می شویم که کدام کادر مربوط به سرعت بود و کدام مربوط به قطر!

لذا از قانون نام گذاری پیشوندها استفاده می کنیم. روی هر ابزار کلیک کنید و در پنجره Properties، خاصیت (Name) را تغییر دهید:

برای کادرهای متن پیشوند txt بگذارید. مثال: txtDiameter (کادر قطر)

برای دکمه ها پیشوند btn بگذارید. مثال: btnCalculate (دکمه محاسبه)

برای برچسبها پیشوند lbl بگذارید. مثال: lblResult (برچسب نتیجه)

(نکته: خاصیت Name با خاصیت Caption تفاوت دارد. Name نام مخفی ابزار برای برنامه نویس است، اما Caption متنی است که کاربر روی آن ابزار می بیند).

فصل سوم: متغیرها (Variables) و مدیریت حافظه (RAM)

فرض کنید به عنوان انباردار یک کارخانه انتخاب شده‌اید. قطعات مختلفی (پیچ، مهره، ورق فولادی، روغن) به انبار می‌آیند. شما نمی‌توانید روغن را در قفسه توری، یا ورق فولادی را در بشکه بریزید! هر ماده‌ای به محفظه مخصوص خود و برجسب نام نیاز دارد.

در برنامه‌نویسی، حافظه RAM کامپیوتر همان انبار ماست. متغیرها، ظرف‌هایی هستند که ما در این انبار می‌سازیم تا داده‌ها را در طول اجرای برنامه در آن‌ها نگه داریم.

۱-۳. انواع داده‌ها (Data Types)

بسته به اینکه چه چیزی را می‌خواهیم ذخیره کنیم، باید نوع متغیر را به درستی انتخاب کنیم تا حافظه کامپیوتر هدر نرود.

۱. نوع Integer (عدد صحیح):

برای ذخیره اعداد بدون اعشار در محدوده $-32,768$ تا $32,767$ استفاده می‌شود. ۲ بایت از حافظه را اشغال می‌کند.

کاربرد در مهندسی: تعداد دندانه‌های یک چرخ‌دنده، تعداد سوراخ‌های فلنج، تعداد پاس‌های ماشین کاری.

۲. نوع Long (عدد صحیح بزرگ):

اگر عدد صحیح ما بزرگتر از ۳۲ هزار بود، از این نوع استفاده می‌کنیم. ۴ بایت اشغال می‌کند.

کاربرد: شمارش تعداد قطعات تولیدی در طول یک سال.

۳. نوع Single و Double (اعداد اعشاری):

برای اعدادی که نیاز به دقت اعشاری دارند. Double دقت بسیار بالاتری دارد (تا ۱۵ رقم اعشار) و در محاسبات مهندسی ما همیشه از Double استفاده می‌کنیم. ۸ بایت اشغال می‌کند.

کاربرد: مختصات دقیق XXX و YYY برای دستگاه CNC، محاسبه تکرانس‌ها، سرعت‌ها.

۴. نوع String (رشته متنی):

برای ذخیره حروف، کلمات و جملات. این نوع داده نمی‌تواند در عملیات ریاضی شرکت کند.

کاربرد: نام اپراتور دستگاه، کدهای G-Code مانند "G۰۱ X۲۰ Y۳۰".

۵. نوع Boolean (منطقی):

فقط می‌تواند دو حالت داشته باشد: True (درست/روشن) یا False (غلط/خاموش).

کاربرد: آیا پمپ آب صابون روشن است؟ (بله/خیر).

۲-۳. نحوه تعریف متغیرها (دستور Dim)

برای ساختن یک متغیر در زبان بیسیک، از کلمه کلیدی Dim (مخفف Dimension) استفاده می‌کنیم.

ساختار کلی:

Dim [نام متغیر] As [نوع داده]

مثال:

```
vba
Dim ToolNumber As Integer ' متغیری برای ذخیره شماره ابزار
Dim SpindleSpeed As Double ' متغیری برای ذخیره سرعت دوران
Dim OperatorName As String ' متغیری برای ذخیره نام اپراتور
```

۳-۳. انتساب مقدار به متغیر

پس از ساخت ظرف، حالا باید درون آن چیزی بریزیم. این کار با علامت مساوی = انجام می‌شود. دقت کنید که در برنامه‌نویسی، علامت مساوی به معنای "برابری ریاضی" نیست، بلکه به معنای "ریختن مقدار سمت راست به درون متغیر سمت چپ" است.

```
vba
ToolNumber = 5
SpindleSpeed = 1250,75
OperatorName = "Ali Rezaei" ' قرار بگیرند
```

🌸 پروژه عملی ۱: نرم‌افزار محاسبه حجم و وزن شمش استوانه

شرح مسئله: می‌خواهیم برنامه‌ای بنویسیم که قطر (DDD) و طول (LLL) یک شمش استوانه‌ای از جنس فولاد را به میلی‌متر بگیرد، حجم آن را حساب کرده و با توجه به چگالی فولاد (۷۸۵ g/cm^۳) را به میلی‌متر بگیرد، حجم آن را حساب کرده و با توجه به چگالی فولاد (۷۸۵ g/cm^۳) وزن نهایی قطعه را به کیلوگرم به ما اعلام کند. این برنامه برای محاسبه وزن مواد خام در کارگاه ریخته‌گری یا تراشکاری بسیار حیاتی است.

فرمول‌ها:

$$\text{شعاع: } \frac{D}{2} = R \quad \text{حجم: } \frac{D^2}{4} \times L = V$$

حجم استوانه به سانتی متر مکعب:

$$\text{Volume} = \pi \times (R \cdot 10)^2 \times (L \cdot 10) \quad \text{Volume} = \pi \times \left(\frac{R}{10}\right)^2 \times \left(\frac{L}{10}\right)$$

وزن به کیلوگرم:

$$\text{Weight} = \text{Volume} \times 7,85 \cdot 1000 \quad \text{Weight} = \frac{\text{Volume} \times 7,85}{1000}$$

مراحل ساخت رابط کاربری (UI):

۱. یک UserForm ایجاد کنید.

۲. دو Label برای راهنمایی کاربر (متن: "قطر شمش (mm)" و "طول شمش (mm)") قرار دهید.

۳. دو TextBox مقابل آنها قرار دهید. نام اولی را txtD و دومی را txtL بگذارید.

۴. یک CommandButton با نام btnCalc و متن "محاسبه وزن" در پایین فرم قرار دهید.

کدنویسی:

روی دکمه "محاسبه وزن" دبل کلیک کنید تا وارد محیط کدنویسی شوید. کدهای زیر را با دقت و خط به خط بنویسید (متن های سبز رنگ که با آپوستروف ' شروع شده اند، توضیحات برنامه هستند و کامپیوتر آنها را اجرا نمی کند. نوشتن توضیحات، عادت مهندسان حرفه ای است).

```
vba
Private Sub btnCalc_Click()
    -- مرحله ۱: تعریف متغیرهای مورد نیاز --
    Dim D As Double, L As Double
    Dim R As Double
```

```
Dim Vol As Double
Dim Weight As Double
Dim Pi As Double

' مقداردهی ثابت پی
Pi = ۳,۱۴۱۵۹۲

' -- مرحله ۲: دریافت اطلاعات از رابط کاربری --
' متن داخل کادر را به عدد خالص تبدیل میکند تا از خطا جلوگیری Val دستور
شود
D = Val(txtD.Text)
L = Val(txtL.Text)

' -- مرحله ۳: انجام محاسبات هندسی و فیزیکی --
R = D / ۲

' محاسبه حجم (چون ابعاد به میلی‌متر گرفته شده، برای تبدیل به سانتی‌متر
مکعب تقسیم بر ۱۰ میکنیم)
Vol = Pi * (R / ۱۰) ^ ۲ * (L / ۱۰)

' محاسبه وزن فولاد به کیلوگرم
Weight = (Vol * ۷,۸۵) / ۱۰۰۰

' -- مرحله ۴: نمایش خروجی به کاربر --
' خروجی را تا ۲ رقم اعشار گرد میکند Round دستور
MsgBox "کیلوگرم " & Round(Weight, ۲) & " :وزن شمش فولادی برابر است با",
vbInformation, "نتیجه محاسبه"

End Sub
```

تحلیل فنی کد (کالبدشکافی کد):

Val(): کاربران ممکن است به جای عدد ۵۰، اشتباهاً بنویسند "۵۰mm". اگر بخواهیم حروف را ضرب و تقسیم کنیم برنامه Crash می‌کند (هنگ می‌کند). تابع Val حروف اضافه را حذف کرده و فقط عدد را استخراج می‌کند.

عملگر $^{\wedge}$: در محیط VBA برای به توان رساندن اعداد از علامت $^{\wedge}$ (Shift+) استفاده می‌شود. $2R^{\wedge}$ یعنی شعاع به توان ۲.

عملگر &: این عملگر برای «چسباندن» استفاده می‌شود. در خط آخر، ما یک متن ثابت («وزن شمش...») را به یک متغیر عددی (Weight) و سپس به کلمه ("کیلوگرم") چسبانده‌ایم تا جمله کاملی ساخته شود.

✂ خطایابی (Debugging):

اگر هنگام اجرای برنامه با خطای Type Mismatch مواجه شدید، به این معنی است که سعی کرده‌اید متن را در ظرفی که برای اعداد ساخته شده (Dim ... As Double) بریزید. حتماً چک کنید که از دستور Val استفاده کرده باشید.

تمرینات پایان فصل سوم:

۱. برنامه‌ای بنویسید که طول، عرض و ضخامت یک مکعب مستطیل را بگیرد و حجم آن را محاسبه کند.
۲. فرمول محاسبه سرعت پیشروی ماشین کاری به این صورت است: $ZF = N \times f_z \times fz \times N = F$ که در آن NNN دور اسپیندل، fzf_zfz بار به ازای هر دندانه و ZZZ تعداد دندانه‌های ابزار است. برنامه‌ای با سه TextBox بنویسید که این مقادیر را گرفته و سرعت پیشروی (FFF) را محاسبه کند. برای هر متغیر چه نوع داده‌ای (Data Type) انتخاب می‌کنید؟

فصل چهارم: ساختارهای تصمیم‌گیری (هوش‌بخشیدن به برنامه)

در فصل‌های قبل، برنامه‌هایی نوشتیم که خط به خط از بالا به پایین اجرا می‌شدند. اما در دنیای واقعی مهندسی، ما همیشه نیاز به تصمیم‌گیری داریم.

به عنوان یک بازرس کنترل کیفیت (QC)، وقتی قطعه‌ای را با میکرومتر اندازه می‌گیرید، از خود می‌پرسید: “آیا این اندازه در محدوده تolerانس مجاز هست یا خیر؟” اگر بله، قطعه تایید می‌شود؛ اگر خیر، قطعه یا باید دوباره ماشین‌کاری شود (Rework) و یا کاملاً ضایعات (Scrap) است.

برای اینکه کامپیوتر هم بتواند چنین تصمیماتی بگیرد، از دستورات شرطی استفاده می‌کنیم. معروف‌ترین دستور شرطی در تمام زبان‌های برنامه‌نویسی، دستور If (اگر) است.

۴-۱. عملگرهای مقایسه‌ای (Relational Operators)

برای ساختن یک شرط، ما باید دو چیز را با هم مقایسه کنیم. در زبان VBA از علامت‌های زیر استفاده می‌شود:

- = : مساوی است با
- <> : نامساوی است (در ریاضی \neq)
- < : بزرگتر است از
- > : کوچکتر است از
- <= : بزرگتر یا مساوی است با
- >= : کوچکتر یا مساوی است با

۴-۲. عملگرهای منطقی (Logical Operators)

گاهی اوقات شرط ما ترکیبی است. مثلاً: “قطر باید از ۲۴.۹۸ بزرگتر و از ۲۵.۰۲ کوچکتر باشد.”

- And (و): زمانی شرط برقرار است که هر دو طرف درست باشند.
- Or (یا): زمانی شرط برقرار است که حداقل یکی از طرفین درست باشد.
- Not (نقیض): نتیجه شرط را برعکس می‌کند (درست را به غلط و بالعکس تبدیل می‌کند).

۴-۳. ساختار دستور If - Then - Else

قالب کلی دستور شرطی به شکل زیر است:

```
vba
If [شرط یا مقایسه] Then
    کارهایی که اگر شرط درست بود باید انجام شود '
ElseIf [شرط دوم در صورت غلط بودن شرط اول] Then
    کارهایی که اگر شرط دوم درست بود انجام شود '
Else
    کارهایی که اگر هیچکدام از شروط بالا درست نبود انجام شود '
End If
```

🌸 پروژه عملی ۲: دستیار هوشمند کنترل کیفیت (تلرانس سنج)

شرح مسئله: یک شفت فولادی باید با قطر اسمی $25 \text{ mm} \pm 0.05 \text{ mm}$ تولید شود. نقشه مهندسی،

تلرانس را $0.0525 \pm 0.05 \text{ mm}$ تعیین کرده است. ما می‌خواهیم نرم‌افزاری بسازیم که اندازه

اندازه‌گیری شده توسط اپراتور را بگیرد و به او بگوید که قطعه در چه وضعیتی است.

- حالت اول: اندازه بین $24.9524/9524/95$ و $25.0525/0525/05$ است → قطعه تایید است.
- حالت دوم: اندازه بزرگتر از $25.0525/0525/05$ است → قطعه گوشت اضافه دارد و قابل اصلاح (ماشین‌کاری مجدد) است.
- حالت سوم: اندازه کوچکتر از $24.9524/9524/95$ است → قطعه لاغر شده و ضایعات است.

مراحل ساخت رابط کاربری (UI):

۱. یک UserForm جدید بسازید.

۲. یک TextBox برای دریافت اندازه قطعه (با نام txtMeasure) قرار دهید.

۳. یک CommandButton برای بررسی وضعیت (با نام btnCheck) قرار دهید.

۴. یک Label بزرگ در پایین فرم با نام lblResult قرار دهید تا نتیجه را با رنگ‌های مختلف در آن نشان

دهیم.

[اسکرین شات از فرم طراحی شده دستیار کنترل کیفیت]

کدنویسی:

```
vba
Private Sub btnCheck_Click()
    Dim Measure As Double
    Dim UpperLimit As Double, LowerLimit As Double

    ' تعریف حدود مجاز بر اساس نقشه
    UpperLimit = ۲۵,۰۵
    LowerLimit = ۲۴,۹۵

    ' دریافت اندازه قطعه از اپراتور
    Measure = Val(txtMeasure.Text)

    ' بررسی شرطها
    If Measure >= LowerLimit And Measure <= UpperLimit Then
        ' اگر قطعه تایید بود
        lblResult.Caption = " (OK) قطعه تایید شد"
        lblResult.BackColor = vbGreen ' تغییر رنگ پسزمینه برجسب به سبز

    ElseIf Measure > UpperLimit Then
        ' اگر قطعه گوشت اضافه داشت
        lblResult.Caption = " (Rework) نیاز به ماشینکاری مجدد"
        lblResult.BackColor = vbYellow ' تغییر رنگ به زرد

    ElseIf Measure < LowerLimit Then
        ' اگر قطعه لاغر شده بود
        lblResult.Caption = " (Scrap) ! قطعه ضایعات است"
        lblResult.BackColor = vbRed ' تغییر رنگ به قرمز

    End If
End Sub
```

کالبدشکافی کد:

- **vbRed ,vbGreen**: در محیط VBA، رنگ‌های پایه به صورت کلمات از پیش تعریف شده (Constants) وجود دارند. با تغییر خاصیت `BackColor` (رنگ پس‌زمینه) توانستیم یک آلام بصری عالی برای اپراتور دستگاه طراحی کنیم تا با یک نگاه متوجه وضعیت قطعه شود.
- **اهمیت ترتیب شروط**: نرم‌افزار خطوط را از بالا به پایین می‌خواند. به محض اینکه یکی از شرطها درست باشد، وارد آن می‌شود و بقیه `ElseIf` ها را نادیده می‌گیرد و مستقیماً به `End If` می‌پرد.

تمرینات پایان فصل چهارم:

۱. الگوریتم و فلوچارت برنامه بالا (کنترل تلرانس) را روی کاغذ رسم کنید.

۲. برنامه‌ای بنویسید که سرعت اسپیندل (RPM) را دریافت کند. اگر سرعت بالای ۳۰۰۰۳۰۰۰۳۰۰۰ بود، پیام دهد “خطر: دور بحرانی! آیا از سیستم آب‌صابون فشار قوی استفاده می‌کنید؟” در غیر این صورت پیام دهد “دور نرمال است.”

فصل پنجم: حلقه‌های تکرار (اتوماسیون کارهای خسته‌کننده)

یکی از بزرگترین مزیت‌های کامپیوترها این است که هرگز خسته نمی‌شوند! فرض کنید می‌خواهید روی یک صفحه فلزی، ۱۰۰ سوراخ به فاصله مساوی مته‌کاری کنید. اگر بخواهید مختصات مرکز این ۱۰۰ سوراخ را با ماشین حساب حساب کنید و روی کاغذ بنویسید، حداقل یک ساعت زمان می‌برد و احتمال خطای انسانی بسیار بالاست.

در برنامه‌نویسی، ابزاری به نام حلقه (Loop) وجود دارد که یک فرمول یا یک کار را هزاران بار در کسری از ثانیه تکرار می‌کند.

۱-۵. ساختار حلقه For - Next

پربکارترین حلقه در محاسبات مهندسی، حلقه For است. ما از این حلقه زمانی استفاده می‌کنیم که دقیقاً می‌دانیم یک کار چند بار باید تکرار شود (مثلاً ۵۰ پاس تراشکاری).

قالب کلی:

```
vba
[مقدار شروع] = [متغیر شمارنده] To [مقدار پایان] Step [گام حرکت]
    کدهایی که باید تکرار شوند
Next [متغیر شمارنده]
```

- **متغیر شمارنده:** یک متغیر از نوع Integer است که تعداد دفعات چرخش حلقه را می‌شمارد.
- **Step:** مشخص می‌کند که شمارنده در هر چرخش چقدر اضافه یا کم شود. اگر نوشته نشود، نرم‌افزار به صورت پیش‌فرض یکی یکی بالا می‌رود.

۲-۵. کار با سلول‌های اکسل (Cells)

تا الان ما خروجی‌ها را روی فرم (داخل Label یا MsgBox) نشان می‌دادیم. اما در کارهای مهندسی، ما نیاز داریم جداول بزرگی از داده‌ها بسازیم. اینجاست که قدرت اکسل نمایان می‌شود.

ما می‌توانیم از طریق کدهای VBA، مستقیماً داخل سلول‌های اکسل بنویسیم یا از آن‌ها بخوانیم:

دستور: `Cells(Row, Column).Value\Sheet`.

Row: شماره سطر (۱، ۲، ۳...)

Column: شماره ستون (۱ برای A، ۲ برای B، ۳ برای C...)

مثال: `Cells(۱Sheet, ۱).Value = ۲` (عدد ۵۰ را در سطر اول و ستون دوم، یعنی سلول B۱ بنویس).

🌀 پروژه عملی ۳: تولید مختصات پاس‌های کف‌تراشی (Face Milling)

شرح مسئله: ما یک بلوک فولادی داریم و می‌خواهیم با دستگاه فرز CNC سطح آن را کف‌تراشی کنیم تا

ضخامت آن کاهش یابد. ابزار ما روی سطح قطعه ($Z=0$) مماس شده است. می‌خواهیم در مجموع 10 mm

10 mm از قطعه باربرداری کنیم و عمق مجاز باربرداری در هر پاس (Depth of Cut) یا

(apa_pap) برابر با 2 mm باشد.

برنامه‌ای بنویسید که مختصات محور Z را برای تمام پاس‌های ماشین‌کاری تولید کرده و در یک جدول اکسل چاپ کند تا به عنوان Setup Sheet به اپراتور داده شود.

الگوریتم:

۱. از $Z = 0$ شروع کن.

۲. در هر مرحله، ۲ میلی‌متر به سمت منفی حرکت کن (- Step ۲).

۳. این کار را تا زمانی که به $Z = -10$ بررسی تکرار کن.

۴. در هر مرحله، شماره پاس و مختصات Z را در سطرهای اکسل چاپ کن.

مراحل ساخت رابط کاربری:

۱. روی یک UserForm جدید، یک دکمه با نام btnGeneratePath قرار دهید و نام آن را "تولید مسیر ابزار" بگذارید.

[در اینجا اسکرین‌شات از فرم و نمای کنار هم اکسل و محیط VBA قرار داده شود تا مشخص شود خروجی کجا چاپ می‌شود]

کدنویسی:

```
vba
Private Sub btnGeneratePath_Click()
    Dim Z As Integer
    Dim RowNum As Integer
    Dim PassNumber As Integer

    ' ایجاد سرستون‌ها در شیت اکسل
    Sheet1.Cells(1, 1).Value = "شماره پاس"
    Sheet1.Cells(1, 2).Value = "(عمق) Z مختصات"

    ' سطر اول را برای سرستون‌ها اشغال کردیم، پس چاپ داده‌ها از سطر ۲ شروع می‌شود
    RowNum = ۲
    PassNumber = ۱ ' شماره پاس از ۱ شروع می‌شود
```

```
شروع حلقه تکرار: از ۰ تا منفی ۱۰ با گام منفی ۲  
For Z = ۰ To -۱۰ Step -۲  
  
    ' (A) چاپ شماره پاس در ستون اول  
    Sheet\Cells(RowNum, ۱).Value = "Pass " & PassNumber  
  
    ' همراه با واحد (B) چاپ مختصات عمق در ستون دوم  
    Sheet\Cells(RowNum, ۲).Value = "Z" & Z & " mm"  
  
    ' آماده سازی متغیرها برای چرخش بعدی حلقه  
    RowNum = RowNum + ۱ ' یک سطر در اکسل پایین می آیم  
    PassNumber = PassNumber + ۱ ' شماره پاس را یکی اضافه می کنیم  
  
Next Z  
  
MsgBox "مسیر ابزار با موفقیت تولید و در اکسل ثبت شد", vbInformation  
End Sub
```

کالبدشکافی کد:

- چرا متغیر RowNum لازم است؟ حلقه For مقادیر ۰، ۲-، ۴- و... را تولید می کند. ما نمی توانیم این اعداد منفی را به عنوان شماره سطر اکسل استفاده کنیم (اکسل سطر منفی ۲ ندارد!). بنابراین یک متغیر کمکی به نام RowNum ساختیم که از عدد ۲ شروع می شود و مستقل از Z، در هر چرخش یک واحد به آن اضافه می شود تا سطرها به ترتیب (۲، ۳، ۴، ۵ و...) پر شوند.
- حلقه چگونه متوقف می شود؟ به محض اینکه مقدار Z به عدد ۱۰- برسد، کدهای داخل حلقه یک بار دیگر اجرا می شوند و سپس نرم افزار به صورت خودکار از حلقه خارج شده و به خط بعد از Next Z (یعنی پیام MsgBox) می رود.

خطایابی (Debugging) - خطر حلقه بی نهایت (Infinite Loop):

یکی از بدترین خطاهایی که در برنامه نویسی رخ می دهد، حلقه بی نهایت است. اگر شما شرط پایان حلقه را اشتباه بنویسید (مثلاً بنویسید For Z = ۰ To -۱۰ Step -۲) کامپیوتر چون با گام منفی هرگز به عدد مثبت ۱۰ نمی رسد، تا ابد به شمردن ادامه می دهد و نرم افزار اکسل هنگ می کند! در این حالت برای متوقف کردن اجباری برنامه باید کلیدهای Ctrl + Break را روی کیبورد فشار دهید.

۳-۵. حلقه های تو در تو (Nested Loops)

گاهی نیاز داریم یک حلقه داخل حلقه دیگر قرار بگیرد. مثلاً فرض کنید می خواهیم روی یک صفحه مستطیلی، ۳ ردیف سوراخ بزنیم که در هر ردیف ۵ سوراخ قرار دارد (مجموعاً ۱۵ سوراخ).

در این حالت از دو حلقه استفاده می‌کنیم: حلقه بیرونی برای حرکت در محور Y (ردیف‌ها) و حلقه درونی برای حرکت در محور X (ستون‌ها). این مفهوم پایه و اساس تولید الگوهای سوراخ کاری (Hole Patterns) در ماشین کاری CNC است که در فصل‌های آینده به آن خواهیم پرداخت.

تمرینات پایان فصل پنجم:

۱. در پروژه عملی ۳، کدی را به برنامه اضافه کنید که قبل از تولید مسیر جدید، اطلاعات قبلی روی شیت اکسل را پاک کند. (راهنمایی: می‌توانید از دستور `ClearContents` در `Range("A1:Sheet100:B2")` در ابتدای برنامه استفاده کنید).

۲. برنامه‌ای بنویسید که با استفاده از حلقه `For`، جدول ضرب ۱ تا ۱۰ را در ستون A و B اکسل چاپ کند (مثال: در ستون اول بنویسد "۱ * ۵ = ۵" و در ستون دوم جواب آن یعنی ۵ را بنویسد).

فصل ششم: توابع ریاضی و تولید G-Code (پروژه نهایی)

در دنیای ماشین کاری و برنامه‌نویسی CNC، هندسه و ریاضیات حرف اول را می‌زنند. محاسبه مختصات دقیق برخورد ابزار با قطعه کار، نیازمند استفاده از توابع مثلثاتی (سینوس و کسینوس) و جبر است. زبان VBA توابع ریاضی قدرتمندی را در اختیار ما قرار می‌دهد که در این فصل با آن‌ها آشنا می‌شویم.

۱-۶. توابع پایه ریاضی در VBA

پرکاربردترین توابعی که در رشته ماشین‌افزار به آن‌ها نیاز دارید عبارتند از:

- **قدر مطلق (Abs):** علامت منفی را از بین می‌برد. در محاسبه فاصله بین دو نقطه کاربرد دارد.
- **مثال:** `Abs(-۱۰/۵)` خروجی `۱۰/۵` می‌دهد.
- **جذر یا ریشه دوم (Sqr):** برای محاسبه وتر مثلث یا استفاده در قضیه فیثاغورس ($C = \sqrt{2B + 2A}$)
- **مثال:** `Sqr(۲۰)` خروجی `۴.۴۷` می‌دهد.

۲-۶. توابع مثلثاتی و «تله رادیان»

توابع Sin (سینوس)، Cos (کسینوس) و Tan (تانژانت) در VBA به صورت پیش فرض وجود دارند. اما یک نکته بسیار مهم و حیاتی مهندسی در اینجا وجود دارد:

⚠ نکته طلایی (تله رادیان): تمام زبان‌های برنامه‌نویسی (از جمله VBA)، زاویه را بر حسب رادیان (Radian) دریافت می‌کنند، نه درجه (Degree)! اگر شما بنویسید Sin(۳۰)، کامپیوتر سینوس ۳۰ رادیان را حساب می‌کند نه ۳۰ درجه!

راه‌حل: ما باید همیشه زاویه را قبل از دادن به تابع، از درجه به رادیان تبدیل کنیم. فرمول تبدیل بسیار ساده است:

$$\text{Radian} = \text{Degree} \times \left(\frac{\pi}{180} \right)$$
$$\text{Radian} = \text{Degree} \times (\pi / 180)$$

برای استفاده از عدد پی (π) در VBA، بهترین راه استفاده از تابع داخلی اکسل است:

```
Pi = WorksheetFunction.Pi()
```

۳-۶. چسباندن رشته‌ها (String Concatenation)

در دستگاه‌های CNC، فرمت کدها ترکیبی از حروف و اعداد است (مثل X۲۰۵Y-۱۰۰۷). برای اینکه بتوانیم در VBA یک حرف ثابت (مثل "X") را به یک متغیر عددی بچسبانیم، از علامت & (آمپرسند) استفاده می‌کنیم.

- مثال: اگر متغیر X_Pos برابر با ۱۲/۵۱۲/۵۱۲/۵ باشد، عبارت "X" & X_Pos خروجی متن X۱۲/۵ را تولید می‌کند.

🌀 پروژه نهایی (Master Project): تولید G-Code سوراخ کاری روی دایره (BHC)

شرح مسئله: یکی از رایج‌ترین و زمان‌برترین کارها در کارگاه، سوراخ کاری روی یک مسیر دایره‌ای (Bolt Hole Circle) است (مثلاً سوراخ‌های روی فلنج). اپراتور باید مختصات XXX و YYY مرکز هر سوراخ را با سینوس و کسینوس حساب کند.

ما می‌خواهیم نرم‌افزاری بسازیم که شعاع دایره (RRR)، تعداد سوراخ‌ها (NNN) و زاویه شروع (θ) را از کاربر بگیرد و با زدن یک دکمه، تمام G-Code های لازم برای سیکل سوراخ کاری (G1G1G1G) را در اکسل چاپ کند!

تحلیل ریاضی مسئله:

۱. زاویه بین هر سوراخ برابر است با: $\Delta\theta = \frac{360}{N}$

۲. مختصات هر سوراخ از روابط زیر به دست می‌آید:

$$X = R \times \cos(\text{Angle}) \quad X = R \times \cos(\text{Angle})$$

$$Y = R \times \sin(\text{Angle}) \quad Y = R \times \sin(\text{Angle})$$

مراحل ساخت رابط کاربری (UI):

۱. یک UserForm جدید باز کنید.

۲. سه عدد TextBox برای دریافت اطلاعات قرار دهید و نام آن‌ها را txtRadius (شعاع)، txtHoles (تعداد سوراخ) و txtStartAngle (زاویه شروع) بگذارید.

۳. یک CommandButton با نام btnGenerateGCode (تولید برنامه CNC) قرار دهید.

[در اینجا اسکرین‌شات از فرم زیبای پروژه نهایی قرار داده شود]

کدنویسی پروژه نهایی:

روی دکمه دوبار کلیک کنید و کدهای زیر را با دقت وارد کنید:

```
vba
Private Sub btnGenerateGCode_Click()
    ' ۱. تعریف متغیرها
    Dim R As Double, StartAngle As Double
    Dim NumHoles As Integer
    Dim AngleStep As Double, CurrentAngle As Double
    Dim X As Double, Y As Double
    Dim Radian As Double
    Dim Pi As Double
    Dim i As Integer
```

```
Dim RowNum As Integer

' پاک کردن شیت اکسل برای برنامه جدید
Sheet\Cells.ClearContents

' دریافت اطلاعات از فرم ۲.
R = Val(txtRadius.Text)
NumHoles = Val(txtHoles.Text)
StartAngle = Val(txtStartAngle.Text)

' محاسبات اولیه ۳.
Pi = WorksheetFunction.Pi() ' دریافت عدد پی از اکسل
AngleStep = ۳۶۰ / NumHoles ' زاویه بین هر سوراخ
CurrentAngle = StartAngle ' تنظیم اولین زاویه روی زاویه شروع

' در اکسل G-Code چاپ سرستون‌های
Sheet\Cells(۱, ۱).Value = "شماره بلوک"
Sheet\Cells(۱, ۲).Value = "G-Code"
RowNum = ۲

' حلقه تولید مختصات برای هر سوراخ ۴.
For i = ۱ To NumHoles

    ' تبدیل زاویه فعلی از درجه به رادیان
    Radian = CurrentAngle * (Pi / ۱۸۰)

    ' محاسبه مختصات X و Y
    X = R * Cos(Radian)
    Y = R * Sin(Radian)

    ' CNC استاندارد دستگاه) گرد کردن اعداد تا ۳ رقم اعشار
    X = Round(X, ۳)
    Y = Round(Y, ۳)

    ' در شیت اکسل CNC چاپ کدهای ۵.
    Sheet\Cells(RowNum, ۱).Value = "N" & (i * ۱۰)

    If i = ۱ Then
        ' و سطح Z-۵ با عمق فرض) را روشن می‌کنیم G۸۱ برای سوراخ اول، سیکل
        R۲)
        Sheet\Cells(RowNum, ۲).Value = "G۹۰ G۸۱ X" & X & " Y" & Y & "
        Z-۵,۰ R۲,۰ F۱۰۰"
    Else
        ' برای سوراخ‌های بعدی فقط مختصات جدید را می‌دهیم
        Sheet\Cells(RowNum, ۲).Value = "X" & X & " Y" & Y
    End If

    ' آماده‌سازی برای سوراخ بعدی ۶.
    CurrentAngle = CurrentAngle + AngleStep ' زاویه را به اندازه گام
    بعدی می‌چرخانیم
    RowNum = RowNum + ۱ ' یک سطر در اکسل پایین می‌رویم

Next i

' لغو سیکل سوراخ‌کاری در پایان
Sheet\Cells(RowNum, ۱).Value = "N" & ((i) * ۱۰)
Sheet\Cells(RowNum, ۲).Value = "G۸۰"

MsgBox "پایان عملیات", vbInformation, "با موفقیت تولید شدند CNC کدهای"
```

End Sub

کالبدشکافی کد (نکات طلایی):

Round(X, ۳): دستگاه‌های فرز CNC معمولاً تا ۳ رقم اعشار (میکرون) را می‌خوانند. اگر عدد را گرد نکنیم، کامپیوتر ممکن است عددی مثل ۱۲.۵۴۳۲۹۸۷۱ را تولید کند که باعث خطای Syntax در کنترلر CNC می‌شود. تابع Round این مشکل را حل می‌کند.

تولید شماره بلوک (N-Codes): در استاندارد برنامه‌نویسی CNC، خطوط برنامه با N مشخص می‌شوند و معمولاً ۱۰ تا ۱۰ تا بالا می‌روند (N۱۰N, N۲۰, N۳۰). ما با فرمول "N" & "i" (i * ۱۰) این شماره‌ها را به صورت خودکار و بسیار هوشمندانه تولید کردیم!

منطق G۸۱: در خط اول سوراخ‌کاری، ما کل فرمان G۸۱ (سیکل سوراخ‌کاری) را نوشتیم. طبق استاندارد CNC (Modal Codes)، تا زمانی که دستور G۸۰ صادر نشود، سیکل فعال می‌ماند؛ بنابراین در سوراخ‌های دوم به بعد (بخش Else) فقط کفایت مختصات XXX و YYY جدید را به کنترلر بدهیم و در خط آخر G۸۰ را چاپ کردیم.

تمرینات پایان ترم:

۱. در پروژه بالا، اگر کاربر به جای نقطه (مبدأ مختصات در G۰X۰, G۰Y۰, G۰X۰, G۰Y۰) بخواهد دایره سوراخ‌ها را در نقطه G۵۰X۵۰, G۵۰Y۵۰ ایجاد کند، کد را چگونه تغییر می‌دهید؟ (راهنمایی: دو TextBox جدید برای آفست مرکز دایره اضافه کنید و مقادیر آنها را با خروجی سینوس و کسینوس جمع کنید).

۲. برنامه‌ای بنویسید که قطر میلگرد (DDD) و سرعت برشی (VcV_cVc) را از هندبوک ماشین‌کاری بگیرد و دور اسپیندل (RPM) ماشین تراش را با فرمول $N = \frac{V_c \times 1000}{\pi \times D}$ محاسبه کند.

(پایان کتابچه آموزشی)

🎓 سخن پایانی با دانشجویان:

شما اکنون دیگر فقط یک اپراتور یا برنامه‌نویس ساده G-Code نیستید؛ شما یاد گرفتید که چگونه برای کارهای مهندسی خود **ابزارسازی** کنید. ترکیب دانش ماشین کاری با برنامه‌نویسی VBA در اکسل، از شما یک مهندس ساخت و تولید بسیار متمایز و ارزشمند در صنعت می‌سازد که می‌تواند پیچیده‌ترین محاسبات کارگاه را در چند ثانیه اتوماتیک کند. موفق و پیروز باشید!