

(Cyclomatic Complexity) تحليل التعقيد الحلقي:

$$(CC) = D+1$$

في كود الوظيفة لدينا:

```
if (income < 2000)
  if (hasJob)
    if (creditScore >= 700)
      if (dependents == 0)
        else if (dependents <= 2)
          else if (creditScore >= 600)
            if (ownsHouse)
              if (creditScore >= 750 && income > 5000 && ownsHouse)
                ( هنا لدينا && تعتبر نقطتي قرار )
              else if (creditScore >= 650 && dependents == 0) (هنا لدينا
                && كمان تعتبر نقطتي قرار)
```

اصبح التعقيد:

$$(cc)=11+1=12$$

نتيجة كبيرة و الكود يعتبر معقد لهيكله بدنا نعالجه لان كل ما ارتفعت قيمة CC زادت صعوبة الاختبار و الفهم و قابلية الصيانة تتخفض كمان.

سوف نعمل على تقليل التعقيد الحلقي للتابع الرئيسي وزيادة قابلية القراءة وإعادة الاستخدام من خلال تقسيم المنطق إلى توابع فرعية أصغر تركز كل منها على مهمة واحدة.

التحسينات في ملف LoanEvaluation:

أصبح هذا التابع يركز على تحديد ما إذا كان المتقدم لديه وظيفة أم لا ثم يقوم بتفويض عملية **GetLoanEligibility**: التقييم الفعلية إلى توابع فرعية. انخفض التعقيد الحلقي له بشكل ملحوظ.

GetLoanEligibility: $D=2$ إذن $hasJob$ و $income < 2000$ نقاط القرار هي **GetLoanEligibility** الجديد لـ CC ، هذا يصنف على أنه مقبول (2-5)، مما يجعله سهل الاختبار والفهم. $CC=2+1=3$.

EvaluateEmployedApplicant: يعالج منطق المتقدمين الذين لديهم وظيفة.

$D=2$ ، $CC=2+1=3$ إذن $creditScore >= 600$ و $creditScore >= 700$ الجديد: نقاط القرار هي CC .

EvaluateHighCreditEmployed: يعالج منطق المتقدمين ذوي الانتماء العالي ولديهم وظيفة.

$D=2, CC=2+1=3$. إذن $dependents \leq 2$ و $dependents == 0$ الجديد: نقاط القرار هي CC

EvaluateMediumCreditEmployed: يعالج منطق المتقدمين ذوي الائتمان المتوسط ولديهم وظيفة.

$D=1, CC=1+1=2$. إذن $ownsHouse$ الجديد: نقاط القرار هي CC

EvaluateUnemployedApplicant: يعالج منطق المتقدمين الذين ليس لديهم وظيفة.

نقطة 2) $creditScore \geq 750 \ \&\& \ income > 5000 \ \&\& \ ownsHouse$ الجديد: نقاط القرار هي CC
($\&\&$ نقطة بسبب 2) $creditScore \geq 650 \ \&\& \ dependents == 0$ ($\&\&$ بسبب 2) $D=4$,
 $CC=4+1=5$.

طبعا الاختبار تم على التابع `GetLoanEligibility` بمدخلات مختلفة للدخل، الوظيفة، درجة الائتمان، عدد المعالين، امتلاك منزل

```
Administrator: Command Prompt
C:\Windows\System32>cd "C:\Users\XPRISTO\OneDrive\Desktop\software2"
C:\Users\XPRISTO\OneDrive\Desktop\software2>dotnet new sln -n LoanAppSolution
The template "Solution File" was created successfully.
C:\Users\XPRISTO\OneDrive\Desktop\software2>dotnet new classlib -n LoanApp.Core
The template "Class Library" was created successfully.
Processing post-creation actions...
Restoring C:\Users\XPRISTO\OneDrive\Desktop\software2\LoanApp.Core\LoanApp.Core.csproj:
Restore succeeded.
C:\Users\XPRISTO\OneDrive\Desktop\software2>dotnet new xunit -n LoanApp.Tests
The template "xUnit Test Project" was created successfully.
Processing post-creation actions...
Restoring C:\Users\XPRISTO\OneDrive\Desktop\software2\LoanApp.Tests\LoanApp.Tests.csproj:
Restore succeeded.
C:\Users\XPRISTO\OneDrive\Desktop\software2>dotnet sln add LoanApp.Core\LoanApp.Core.csproj
Project 'LoanApp.Core\LoanApp.Core.csproj' added to the solution.
C:\Users\XPRISTO\OneDrive\Desktop\software2>dotnet sln add LoanApp.Tests\LoanApp.Tests.csproj
Project 'LoanApp.Tests\LoanApp.Tests.csproj' added to the solution.
C:\Users\XPRISTO\OneDrive\Desktop\software2>dotnet add LoanApp.Tests\LoanApp.Tests.csproj reference LoanApp.Core\LoanApp.Core.csproj
Reference '..\LoanApp.Core\LoanApp.Core.csproj' added to the project.
C:\Users\XPRISTO\OneDrive\Desktop\software2>cd LoanApp.Tests
C:\Users\XPRISTO\OneDrive\Desktop\software2\LoanApp.Tests>dotnet test
Restore complete (0.6s)
  LoanApp.Core succeeded (5.0s) -> C:\Users\XPRISTO\OneDrive\Desktop\software2\LoanApp.Core\bin\Debug\net9.0\LoanApp.Core.dll
  LoanApp.Tests succeeded (0.9s) -> bin\Debug\net9.0\LoanApp.Tests.dll
[xUnit.net 00:00:00.00] xUnit.net VSTest Adapter v2.8.2+699d445a1a (64-bit .NET 9.0.5)
[xUnit.net 00:00:01.32] Discovering: LoanApp.Tests
[xUnit.net 00:00:01.48] Discovered: LoanApp.Tests
[xUnit.net 00:00:01.49] Starting: LoanApp.Tests
[xUnit.net 00:00:02.06] Finished: LoanApp.Tests
  LoanApp.Tests test succeeded (7.3s)
Test summary: total: 11, failed: 0, succeeded: 11, skipped: 0, duration: 7.2s
Build succeeded in 15.5s
```

تقديم:

آية نصر فئة (9)