**Reflection**

Reflection means given an **object or assembly**, we can investigate its Type (GetType()), and then Attributes, Properties, Methods… **at runtime** to determine the appropriate actions.

The System.Reflection namespace gives you classes for **informations** that we get from Reflection (e.g., PropertyInfo, MethodInfo, MemberInfo, etc.)

Example:

public abstract class Character

{

public abstract void Attack();

}

public class Wizard : Character

{

public override void Attack() { … }

public void GoBruhBruh() { … }

}

public class Human : Character

{

public override void Attack() { … }

// Human cannot GoBruhBruh()

}

public DoSomething(Character char)

{

|  |  |
| --- | --- |
| **Without Reflection** | **With Reflection** |
| if (char as Wizard != null || char as XX != null …)  {  GoBruhBruh();  }  else if (char as Human != null || char as ….)  {  Attack();  } | var CharType = char.GetType();  MethodInfo XXX = CharType.GetMethod(“GoBruhBruh”);  if (XXX == null) char.Attack();  else XXX.Invoke(char, \_param);   * Much cleaner! |

}