Shape and Letter are Abstract classes
to promote GRASP/SOLID principles

Little Line

Shape

+ Texture : String

+ Color: String

+ method(type): type
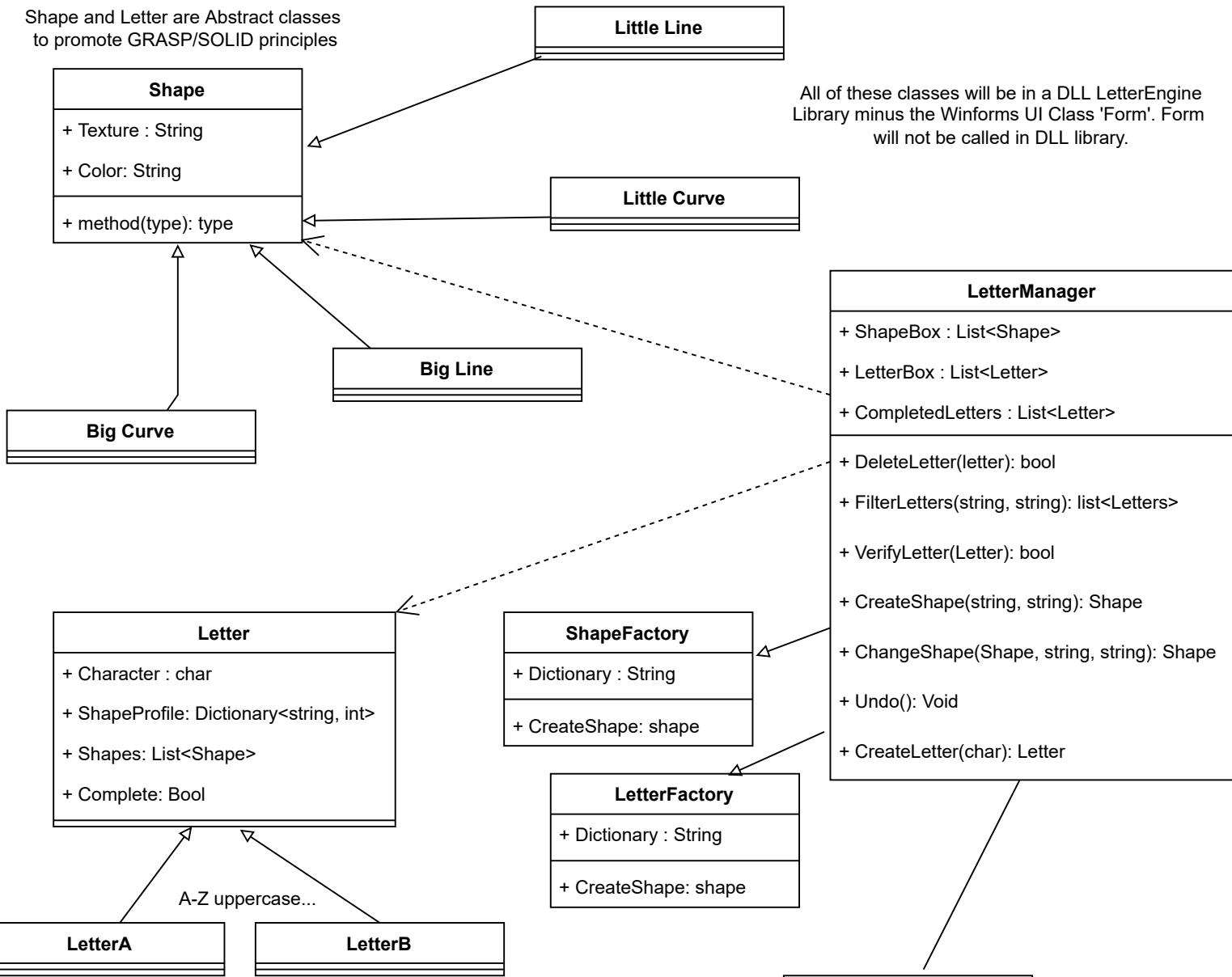
All of these classes will be in a DLL LetterEngine
Library minus the Winforms UI Class 'Form'. Form
will not be called in DLL library.

Little Curve

Big Line

LetterManager

+ ShapeBox : List<Shape>

+ LetterBox : List<Letter>

+ CompletedLetters : List<Letter>

+ DeleteLetter(letter): bool

+ FilterLetters(string, string): list<Letters>

+ VerifyLetter(Letter): bool

+ CreateShape(string, string): Shape

+ ChangeShape(Shape, string, string): Shape

+ Undo(): Void

+ CreateLetter(char): Letter

Big Curve

Letter

+ Character : char

+ ShapeProfile: Dictionary<string, int>

+ Shapes: List<Shape>

+ Complete: Bool

ShapeFactory

+ Dictionary : String

+ CreateShape: shape

LetterFactory

+ Dictionary : String

+ CreateShape: shape

A-Z uppercase...

LetterA

LetterB

Form

SOLID

GRASP:
Information Expert - Have Letter manager deal withall tasks as it has
all the required information to do so
Low Coupling - Classes have little to do with each other
Protection Variations - Used via abstraction
Indirection - LetterManager acting as mediator among Shapes,Letters and UI
Polymorphism - Using Abstraction, each letter and shape will have a different
constructor
High Cohesion - All methods and fields relevant to their class

S: Each class only has one responsibility
O:
L
I - no unnecessary interfaces
D: Depending on abstractions, not concretions