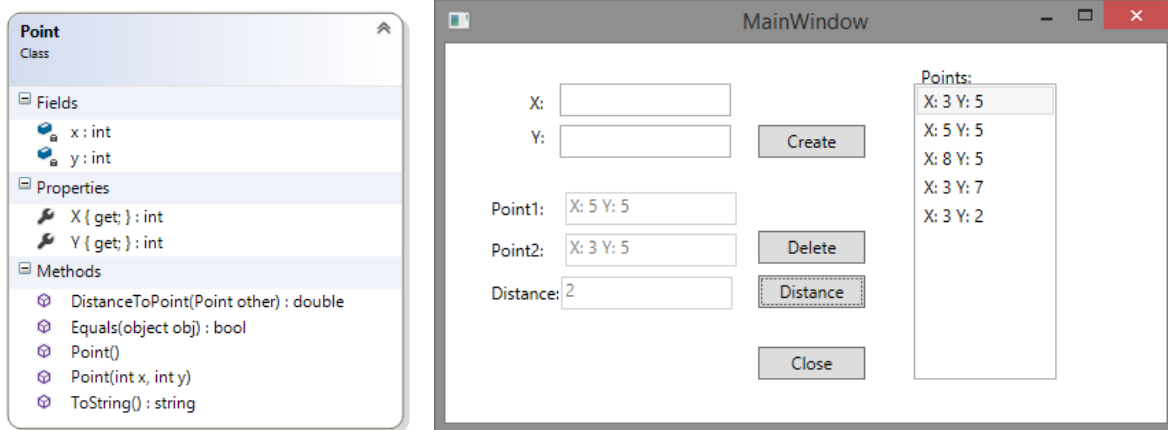


11_PointsWithCollectionClass

Erstelle eine Kopie von deinem Projekt 04_PointsWithList und nenne dein Ordner unter dem Namen 04_PointsWithCollectionClass.

Klasse Point und MainWindow

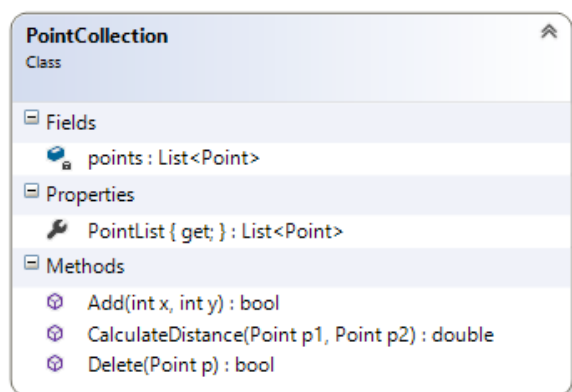


Erweitere die bestehende Klasse Point mit einer *Equals* Methode. Die Methode liefert dann *true*, wenn die x und y Koordinaten gleich sind wie die x und y Koordinaten des Parameters.

Erweitere deine *MainWindow* mit einem Button *Delete*. Dieser soll es ermöglichen, die in der Listbox zuletzt ausgewählte Point-Objekt aus der Liste zu löschen. (**Achtung: aktualisiere auch die entsprechende TextBox Point1 bzw. Point2.**)

Ändere die Implementierung in *MainWindow* so, dass du die Liste der Point-Objekte und die dazugehörige Logik (Operationen) in eine eigene Klasse (*PointCollection*) „auslagerst“. Die GUI soll nur noch für die Ein- und Ausgabe zuständig sein.

Klasse PointCollection



- Feld *points* - speichert die Liste der Pointobjekte.
- Property *PointList* - liefert die Liste *points* für die Anzeige in der GUI.
 - Add – fügt ein neues Point-Objekt in die Liste *points* nur dann ein, wenn dieser Punkt nicht bereits in der Liste gespeichert wurde. Wenn der Punkt bereits vorhanden ist und daher nicht eingefügt werden konnte, liefert die Methode *false* als Rückgabewert, ansonsten *true*.
 - Deletet – entfernt den Punkt, der als Parameter mitgegeben wurde, aus der Liste *points*. Der Rückgabewert informiert darüber, ob der Punkt entfernt werden konnte.
- CalculateDistance – berechnet die Entfernung der beiden Punkte, die als Parameter mitgegeben wurden.

Gutes Gelingen!