

# Blunter 3.0 - C# to Python Transpiler

## Über yWorks GmbH

Die yWorks GmbH ist weltweit führender Anbieter für Softwarebibliotheken zur Visualisierung und Bearbeitung von Graphen, Diagrammen und Netzwerken. Die größten Softwareunternehmen weltweit setzen unsere Bibliotheken ein, um ihrerseits Programme zu erstellen, mit denen strukturierte Daten automatisch erstellt, bearbeitet, angeordnet und dargestellt werden. Unsere kommerziellen Lösungen bieten die umfassendsten Algorithmen zur automatischen Anordnung von strukturierten Daten in Diagrammform. Die Softwarebibliotheken sind für verschiedene Programmierplattformen und -sprachen erhältlich und werden aufgrund ihrer Vielseitigkeit in sämtlichen Industrie- und Wirtschaftsbereichen genauso eingesetzt wie in der Forschung.

## Hintergrund

Unser Kernprodukt „yFiles“ ist eine Programmierbibliothek, die Softwareentwicklern für ihre eigenen Applikationen einsetzen, um komplexe Diagramme und Graphen einfach darzustellen. Ein Schwerpunkt des Produkts sind die automatischen Layoutalgorithmen, die dafür sorgen, dass große strukturierte Daten automatisch in lesbare Diagramme umgewandelt werden können. Algorithmen aus dem Bereich Graphenzeichnen werden hierfür von yWorks originär lediglich in einer einzigen Programmiersprache implementiert. Um diese Algorithmen auch auf anderen Plattformen direkt verfügbar zu machen, besitzt yWorks Hilfsmittel, um den Quelltext dafür automatisch in Quelltext anderer Programmiersprachen zu überführen. Eine besondere Herausforderung bei diesem Übersetzungsvorgang ist, dass das entstehende Resultat nicht nur lauffähig ist, sondern handgeschriebenem Code möglichst nahekommt. Das bedeutet unter anderem, dass es möglich sein soll, den Code nach der Übersetzung weiter manuell zu bearbeiten und direkt zu debuggen. Viel wichtiger ist aber, dass Konventionen, Normen, Konzepte, und möglichst auch Basisbibliotheken weitestgehend auf der Zielseite eingehalten bzw. wiederverwendet werden, damit Programmierer, die mit dem Resultat auf der Zielplattform arbeiten, das Gefühl mit einer „nativen“ Bibliothek zu arbeiten vermittelt wird.

Ein solches Übersetzungstool (interner Codename „Blunter“) setzen wir ein, um Code der in C# geschrieben wird sowohl nach Java, als auch alternativ nach JavaScript zu übersetzen. Dieses Werkzeug ist vollständig bei uns mithilfe der Sprache C# entwickelt worden. Es nutzt die Open-Source .NET Compiler Plattform „Roslyn“ um C# Code zu parsen und zu analysieren. Es ist sowohl unter Windows als auch unter Linux und Mac lauffähig.

Im Zuge dieses Projektes soll das Übersetzungstool „Blunter“ derart erweitert werden, dass es als dritte Sprache Python bei der Ausgabe unterstützt. Das Ziel hierbei ist es eine Klassenbibliothek zu erzeugen, die komfortabel von Entwicklern unter Python 3 genutzt werden kann. Dazu muss in einem ersten Schritt festgelegt werden, wie die jeweiligen Plattform- und Sprachunterschiede aufeinander abgebildet werden sollen. Für yWorks speziell ist es hier nicht notwendig, dass sämtliche Eigenschaften der Ursprungssprache C# abgedeckt werden. Für unseren Anwendungsfall haben wir uns im Ausgangsprodukt auf eine Untermenge der vielfältigen Möglichkeiten von C# beschränkt und weniger häufig genutzte, sehr spezielle Fähigkeiten der Sprache bewusst ausgelassen um die Übersetzung zu vereinfachen.

Im folgenden Schritt kann die Konvertierungslogik implementiert und mit Tests die Ergebnisse überprüft werden. Für die Konvertierung bietet sich eine schrittweise Vorgehensweise an, bei der z.B. initial nur einige Eigenheiten der Quellsprache abgedeckt werden oder Teile zwar lauffähig aber nicht den Plattformnormen entsprechend implementiert werden.

Für die Entwicklung konkret kann bevorzugt JetBrains Rider auf allen der drei großen Plattformen oder Visual Studio unter Windows eingesetzt werden. Während der Entwicklung sollte continuous testing und continuous integration eingesetzt und eine hohe Testabdeckung durch Integrationstest erreicht werden.

Das Projekt wird mit den agilen Methoden der Softwareentwicklung durchgeführt (SCRUM). Dies umfasst alle Schritte von der Festlegung der Roadmap, über die Sprintplanung bis zur Durchführung der Implementierung und der Tests.

Wir erhoffen uns aufgrund des anspruchsvollen aber spannenden Themas eine aktive Mitarbeit. Die Möglichkeit, gestaltend und mit viel Selbstbestimmung an modernster Technologie mitzuwirken verspricht gleichzeitig Spaß und Herausforderungen.

Der Kick-Off wird in den Räumen der yWorks GmbH in Tübingen stattfinden. Mitarbeiter der yWorks GmbH, welche das bestehende Projekt implementiert und begleitet, werden euch hier die Geschichte hinter dem Projekt und das Produkt selbst vorstellen. Sie stehen nachher während des ganzen Projekts beratend als Ansprechpartner zur Verfügung.

## Überblick

### Zielsetzung

- Entwicklung einer Erweiterung des bestehenden C# Sourcecodetranspilers „Blunter“ von der Sprache C# nach Python 3
- Bestimmung der syntaktischen und semantischen Unterschiede zwischen den Sprachen und den Grundbausteinen der jeweiligen Kernbibliotheken
- Festlegung einer Abbildung der unterschiedlichen Funktionalitäten der Sprache C# auf die Sprache Python
- Implementierung eines Ausgabesyntaxbaumes für Python
- Implementierung der Übersetzerfunktionalität
- Überprüfung der Übersetzung mittels Integrationstests

### Anforderungen

- Gute Erfahrungen in objektorientierter Programmierung
- Erfahrungen mit C# und/oder Python
- Interesse an Programmiersprachen und gutes analytisches Denken sind von Vorteil. Erfahrung in Graphentheorie oder mit speziellen Algorithmen ist nicht notwendig.
- Implementierungssprache und vorhandener Code: C#
- Sourcecodeversionierung: Git
- Abdeckung der Anforderungen durch NUnit Tests
- Entwicklungssysteme: Windows, Linux, Mac je nach Vorliebe
- Entwicklungsumgebungen: bevorzugt JetBrains Rider oder Visual Studio mit Erweiterungen
- Continuous Integration: Jenkins oder Atlassian Bamboo

- Teamkommunikation: Slack oder Ähnliches je nach Vorliebe
- Sprintplanung und Durchführung mit Atlassian JIRA
- Begleitende Dokumentation mit Atlassian Confluence