

# Abacus Formula Compiler (AFC)

- *Alle kennen Excel - jetzt sogar Ihre Java-Applikation!*
- Bringt Tabellenkalkulationen auf die JVM
- <http://formulacompiler.org/>
  
- Peter Arrenbrecht für Abacus Research AG
- <http://abacus.ch/>

# Agenda

- Einführung mit Demo
- Was kann AFC Ihnen und Ihren Kunden helfen?
- Aufbau von AFC
- Wie bauen Sie AFC geschickt ein?
- Wie prüfen Sie, ob AFC zu bestehenden Kalkulationen passt?
- Roadmap

# Motivation

- Ausgehend von
  - einem Basisbetrag,
  - dem Alter des Angestellten, und
  - einer Tabelle mit Faktoren pro Altersgruppe,
- bestimmen Sie
  - den passenden Faktor für diesen Angestellten und
- berechnen Sie
  - den effektiven Betrag als Produkt des Basisbetrags und des Faktors.

# Abacus früher

**Payroll type**

Calculation Selection Settings Entries/leavings Transfer Assignment Layout

Validity: Generally valid

**Number**

No.	Line	Op.	Reference value
1		...	Bases ( Basis   Current settlement   16, Basis 16 ) ...
2		...	...

Operand: \*

**Rate**

No.	Line	Op.	Reference value
1		...	Calculate table ( Valid for current client   120, ) ...
2		...	...

anderes Programm!

Table number: 120 Name:

**Define tables**

Search criterion 1

Name: Alter bis

Reference value: Employee age ( Age current year ...

Search function: Greater or equal

Columns

Name: Prozent Amount: 1

Reference value:

Amount type: Percent

ltv.key_1	ltv.value_1
20	10.64
99	8.33

# Abacus heute

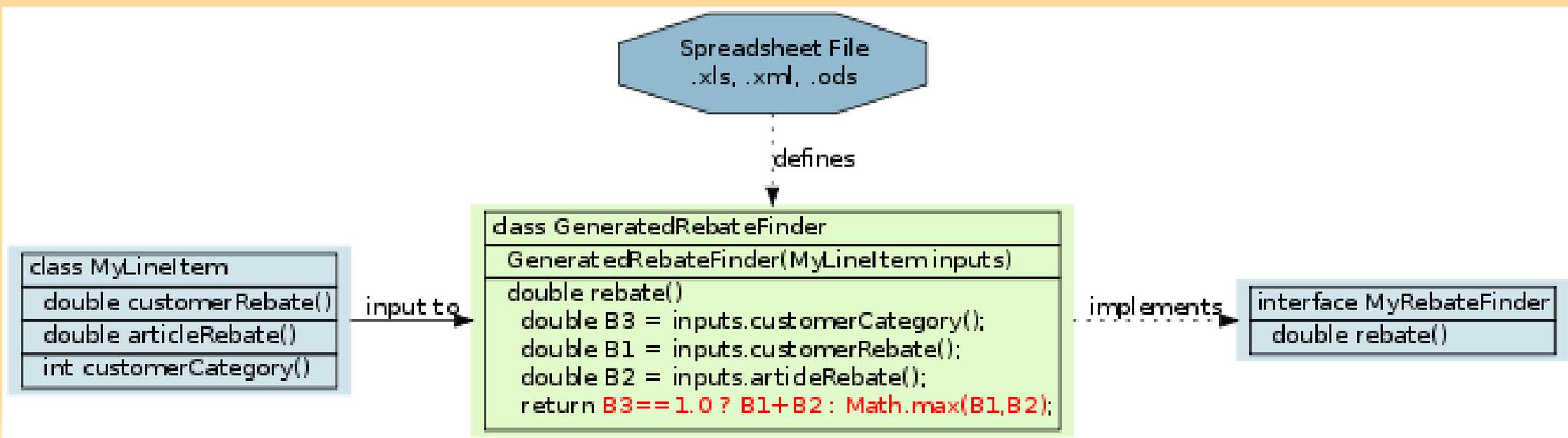
	A	B	C	D	E
1	<b>Payroll type: 3500</b>				
2	<b>Designation: Vacation pay entitlement</b>				
3					
4	<b>Payroll input values:</b>				
5					
6	Bases( Basis; Current settlement; 16, Basis 16 )	120.0000			
7	Employee age ( Age current year )	20			
8					
9	<b>Intermediate values:</b>				
10				<b>From age</b>	<b>Rate</b>
11	Rate	=VLOOKUP(B7,D11:E12,2,1)		0	10.64
12				21	8.33
13	<b>Total:</b>				
14					
15	<b>Result total</b>	1,276.8000			

# Es geht um Anpassbarkeit

- Formeln für Schlüsselwerte unterscheiden sich
  - Von Kunde zu Kunde
  - Von Jahr zu Jahr
- Beispiele
  - Preise und Bedingungen
  - Lieferkonditionen und -spesen
  - Lohnbestandteile (regional und je nach Modell)
  - Bewertungen von Beständen oder Risiken
  - Versicherungsprämien

# Einführung

- Compiler für Tabellenkalkulationen
  - Formeln und Konstanten
  - Eingaben aus Ihren Interfaces
  - Ausgaben in Ihre Interfaces
  - Erzeugt direkt JVM Byte Code (wie .class-Dateien)



# Rahmenbedingungen

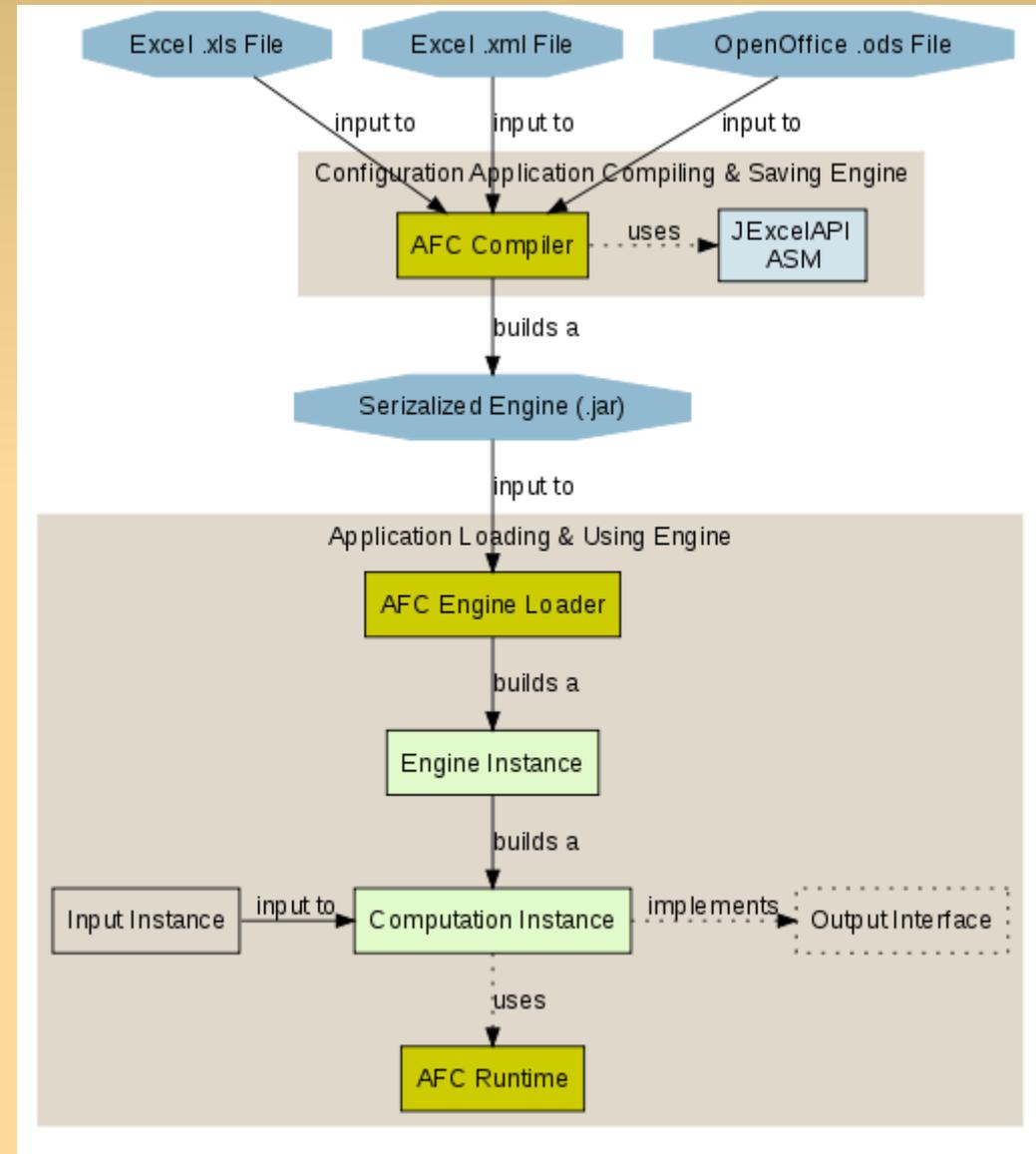
- Compiler und erzeugte Klassen laufen auf normaler JRE
  - Benötigt kein installiertes Excel
  - Benötigt kein JDK
- Erzeugte Klassen sind schnell
- Gut getestet und dokumentiert
- Unter GPL und kommerziell verfügbar

# Demo

- Kurzdemo aus Doku
- Kalkulation aus der Abacus ERP Software
  - Integriert in ABEA und Projektrechnung
  - Ist selber Preisfindung für Abacus-Module
- Highlights
  - Einfaches Setup
  - BigDecimal
  - Einsicht in generierten Code

# Architektur

- Modular
- Kompakte Laufzeitumgebung
- Verschiedene Input-Formate
- Verschiedene numerische Typen
- Wiederholende Daten
- In-memory Verwendung



# Benutzersicht (1)

- Schneller Einstieg
  - AFC kann helfen mit generierten Kalkulationen
- Komfort der Integration
  - Wie binde ich Zellen an verfügbare Eingaben und erwartete Ausgaben?
- Fehlersuche
  - Wie finde ich heraus, warum ein Wert nicht stimmt?
    - AFC kann Berechnungen protokollieren
    - Ausblick: generierte Kalkulationen mit effektiven Daten

# Benutzersicht (2)

- Verwaltung der verwendeten Kalkulationen
  - Wo gespeichert? Versioniert?
- Verhalten über die Zeit
  - Wie viel Anpassungsaufwand habe ich?
    - Input-/Output-Interfaces sind ein API!
    - AFC hat versionierte Runtimes
  - Wie teste ich neue Releases der Software mit meinen Anpassungen?

# Entwicklersicht (1)

- **Verlässlichkeit**
  - Compiler und Runtime sind systematisch getestet
  - Kalkulationen sind funktional. Halten Sie Ihre Eingabeschnittstellen auch so!
- **Dokumentation**
  - Ausführliches Tutorial zu allen Features
  - Referenz aller unterstützten Funktionen
- **Performance**
  - Lazy Evaluation und optionales Caching
  - Bezieht Eingabedaten nach Bedarf

# Entwicklersicht (2)

- Debugging
  - Tracing-Modus
  - Sicht auf erzeugten Code als Java-Source
- Flexibilität
  - Kann auch als Backend für eigene Formelsprachen dienen

# Interna

- Constant-folding zur Compile-Zeit
- Template-Methoden für Code-Generator
  - Von Interpreter und Compiler verwendet
  - Werden bei einem Build von AFC decompiled und daraus Compiler-Code erzeugt
- High-level funktionale Sprache
  - SUM als Fold mit + definiert
  - Vom Interpreter interpretiert
  - Vom Compiler nach JVM abgebildet
- Und ja, es gibt dazu Entwickler-Dokumentation!

# Roadmap

- Demnächst
  - Tool zum Prüfen bestehender Kalkulationen auf Kompatibilität
- Geplant
  - Speichern von Tabellen mit effektiven Werten für Debugging
  - Transparenter Recompile mit javac für Debug-Infos zum Durchsteppen

# Roadmap (2)

- In Diskussion
  - Basis für visuelles Binden von Zellen an Ein-/Ausgaben
  - Basis für automatisierte Tests von Benutzer-Formeln
- Aktuell nicht geplant
  - Array-Formeln
  - Typinferenz für numerischen Typ

# Zusammenfassung

- Anwender wollen Formeln anpassen können
- Sie verwenden Tabellenkalkulationen als UI
  - Verbreitet, interaktiv, nachvollziehbar
  - Benötigt gute Integration in Anwendung
- Abacus Formula Compiler ist dafür geschaffen
  - Direkte Integration, schneller Code
  - BigDecimal für Finanzanwendungen
  - Kein Excel nötig auf App-Server
  - Wird produktiv verwendet
  - GPL oder kommerziell verfügbar
- <http://formulacompiler.org/>