

**POLITECHNIKA ŚLĄSKA**  
**WYDZIAŁ INŻYNIERII MATERIAŁOWEJ I METALURGII**

**Kierunek: Informatyka Przemysłowa**

**Rodzaj studiów: stacjonarne**

**PROJEKT INŻYNIERSKI**  
**Michał Cwiękała**

**INTEGRACJA JAVA Z MS EXCEL**  
**INTEGRATING JAVA WITH MS EXCEL**

**Kierujący projektem:**

**Dr inż. Adrian Smagór**

**Recenzent:**

**Dr Adam Kachel**

**Promotor merytoryczny: Dr inż. Marcin Blachnik**

# Spis treści

1. Wstęp .....	3
2. Cel i zakres pracy .....	4
3. Możliwe narzędzia integracji MS Excel i Java .....	5
3.1. Możliwości rozbudowy .....	5
3.2. Dostępne rozwiązania: .....	5
3.2.1. VBA .....	5
3.2.2. .Net.....	6
3.2.3. Java .....	6
3.3. Wybór technologii. ....	7
3.3.1. Most DCOM - j-Interop.....	7
3.3.2. Biblioteka XLLoop.....	8
3.3.2.1. Zastosowanie.....	8
3.3.2.2. XLLoop składa się z dwóch głównych komponentów: .....	8
4. RapidMiner .....	10
5. Szczegóły implementacyjne projektu.....	12
5.1. Budowa projektu.....	12
5.2. Wykorzystane Biblioteki.....	13
5.3. Interfejs użytkownika - GUI.....	13
5.4. Tworzenie procesu w RapidMiner'ze.....	16
5.5. Sposób działania .....	16
5.6. Główne klasy .....	18
5.7. Sposób tworzenia.....	20
6. Wnioski i podsumowanie.....	21
7. Literatura.....	22
8. Spis rysunków .....	23

# 1. Wstęp

W dzisiejszym skomputeryzowanym świecie biznesowym praca z arkuszem kalkulacyjnym to podstawa w każdej firmie i instytucji, niezależnie od branży. Microsoft Excel to najpopularniejszy na rynku program służący do dokonywania obliczeń zestawionych w formie tabelarycznej.<sup>1</sup> Dostarcza on funkcje matematyczne, finansowe, bazodanowe oraz zapewnia półautomatyczne powtarzanie pisanych formuł.

Excel został napisany początkowo w języku C, a następnie wprowadzono do niego dużo modyfikacji z C++. Wspiera developerów korzystających z C#, który został również stworzony przez Microsoft. Jednak najbardziej popularnym językiem programowania od wielu lat pozostaje Java.

Jan 2016	Jan 2015	Change	Programming Language	Ratings	Change
1	2	▲	Java	21.465%	+5.94%
2	1	▼	C	16.036%	-0.67%
3	4	▲	C++	6.914%	+0.21%
4	5	▲	C#	4.707%	-0.34%
5	8	▲	Python	3.854%	+1.24%

**Rys. 1.1. - Ranking najbardziej popularnych języków programowania na świecie wg. serwisu TIOBE.<sup>2</sup>**

Zagadnienie integracji języka JAVA i MS Excel jest częstym problemem i trudno jednoznacznie określić który sposób jest najlepszy. Każde rozwiązanie oferuje inne możliwości. Niniejsza praca ma na celu zbadania możliwości integracji obu platform i zaproponować najlepsze rozwiązanie.

Technologia Java została stworzona przez firmę Sun Microsystems, jego podstawowe koncepcje zostały przejęte z języka C++ - składnia i słowa kluczowe oraz SamlItalk - maszyna wirtualna i zarządzanie pamięcią.<sup>3</sup>

---

<sup>1</sup> Computerworld: Polski Word 2.0 i Excel 4.0 [dostęp 2009-01-31]

<sup>2</sup> <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> [dostęp 2016-01-13]

<sup>3</sup> <https://www.java.com/pl/about/> [dostęp 2015-12-27]

## 2. Cel i zakres pracy

Niniejsza praca dyplomowa pod tytułem "Integracja Java z MS Excel" ma na celu zbadanie możliwości rozbudowy funkcjonalności programu Microsoft Excel 2007 poprzez wykorzystanie do tego celu modułów i bibliotek dostępnych dla platformy Java

Zakres pracy obejmuje:

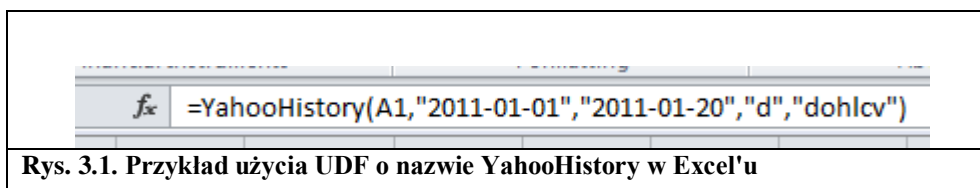
- dobór biblioteki pozwalającej na integrację platformy MS Excel z platformą języka Java,
- w oparciu o wybraną bibliotekę integracja platformy MS Excel z RapidMiner (biblioteką metod eksploracji danych)
- stworzenia wygodnego i przejrzystego interfejsu użytkownika wspierającego integrację procesów

## 3. Możliwe narzędzia integracji MS Excel i Java<sup>4</sup>

### 3.1. *Możliwości rozbudowy*

#### 3.1.1. Excel User-Defined Function (UDF)

UDF jest to funkcja excelowa zdefiniowana przez użytkownika, która wykorzystuje komórki excelowe jako argumenty i zwraca jakąś wartość jako wynik obliczenia.



#### 3.1.2. Makro

Excelowe makro zachowuje się inaczej - potrafi modyfikować wartości i style pomiędzy wieloma excelowymi komórkami i różnymi arkuszami. Używając makra możemy również tworzyć arkusze, menu, zmieniać kolor itd.



### 3.2. *Dostępne rozwiązania:*

#### 3.2.1. VBA

Najprostszą drogą do stworzenia UDF'a w MS Excel jest Visual Basic for Applications - w skrócie VBA. Jest to język programowania oparty na Visual Basic'u dodany do aplikacji w

---

<sup>4</sup> <http://computeraidedfinance.com/2011/11/26/what-shall-i-choose-for-implementing-an-excel-plugin-and-udf/> [dostęp 2015-08-12]

pakiecie Microsoft Office, ale także i innych np. AutoCAD. Służy przede wszystkim do automatyzacji pracy z dokumentem. VBA można łatwo używać, lecz nie nadaje się w większych projektach, gdzie naprawialność i debuggowanie jest mocno ograniczone.

### **3.2.2. .Net**

Najbardziej podstawowymi technologiami służącymi do komunikacji z MS Excelem z użyciem C# jest Visual Studio Tools for Office (VSTO) i Primary Interop Assemblies (PIA). Oba te narzędzia wymagają dodatkowego czasu wykonywania programu na komputerze użytkownika. Do tworzenia UDF najbardziej przydatny jest Excel DNA, który również potrafi obsługiwać makra.

### **3.2.3. Java**

Integracja Java z produktami Microsoft zawsze była problematyczna i wymagała zawiłych rozwiązań. W przypadku generowania plików excelowych przydatna jest biblioteka Apache POI, która pozwala na tworzenie nowych arkuszy, edycję stylów, kalkulacje itd. Niestety jest to komunikacja typu off-line, dostępna tylko z poziomu aplikacji Java. Natomiast aby osiągnąć komunikację on-line w arkuszu Excela, możemy wykorzystać:

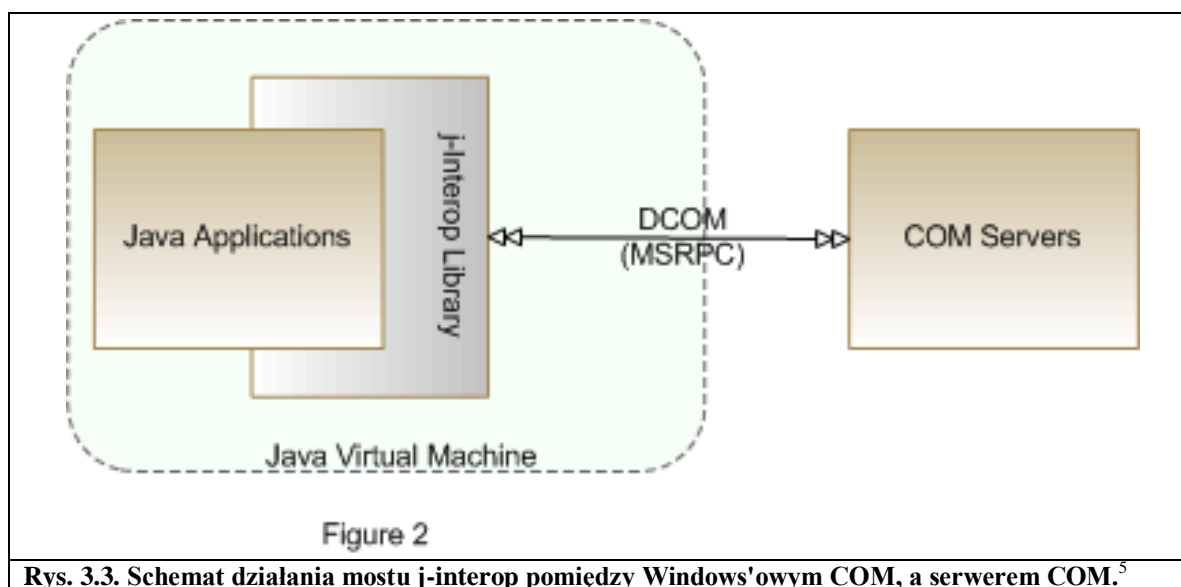
- integrację na poziomie makr - wykorzystuje Excel COM interface. W internecie dostępnych jest wiele bibliotek wspierających mosty COM np. JACOB, com4j i J-Interop. COM (Component Object Model) jest standardem do tworzenia i projektowania interfejsów programistycznych na poziomie binarnym dla każdego modułu wprowadzonym przez Microsoft.
- integrację na poziomie UDF - jest o wiele bardziej skomplikowana niż wezwanie COM. Dobrym open source'owym projektem jest XLLoop. Pozwala na integrację Excela z Java, C, Python, Ruby i wiele więcej. Minusem tego rozwiązania jest konieczność włączenia serwera XLLoop przez użytkownika.
- komunikację za pomocą web service'ów, w której platforma .Net pełni rolę pośrednika pomiędzy Java i Excel'em za pomocą protokołu SOAP

### 3.3. Wybór technologii.

#### 3.3.1. Most DCOM - j-Interop

Praca w początkowej fazie zakładała komunikację z MS Excelem za pomocą mostu COM, gdyż jest najprostszą i najbardziej "naturalną" metodą z możliwych. Do projektu wybrano najpopularniejszy i zarazem najnowszy most COM - j-Interop (<http://www.j-interop.org>). Standardowe mosty COM wykorzystują do komunikacji natywny interfejs Java JNI (Java Native Interface), co znacznie utrudnia komunikację między platformami. J-Interop wykorzystuje protokół DCOM umożliwiający efektywną, dwu-kierunkową i nie-natywną komunikację.<sup>5</sup>

Niestety po pierwszych próbach wystąpiła duża ilość problemów związanych z bezpieczeństwem. Komputer na którym tworzony był projekt wyposażony był w system Windows 7 Home Premium i pakiet Office 2007. Po próbach konfiguracji serwisów, komponentów, ustawień domeny i serwera oraz modyfikacji firewalla, prace zostały wstrzymane. Ze względu na Politykę Bezpieczeństwa Microsoft wykorzystanie tej technologii w tym środowisku jest bardzo problematyczne. Biblioteka była rozwijana najintensywniej w latach 2006-2007 i nie zakłada najnowszych zabezpieczeń. Kolejną fazą projektu było wypróbowanie możliwości biblioteki XLLoop.



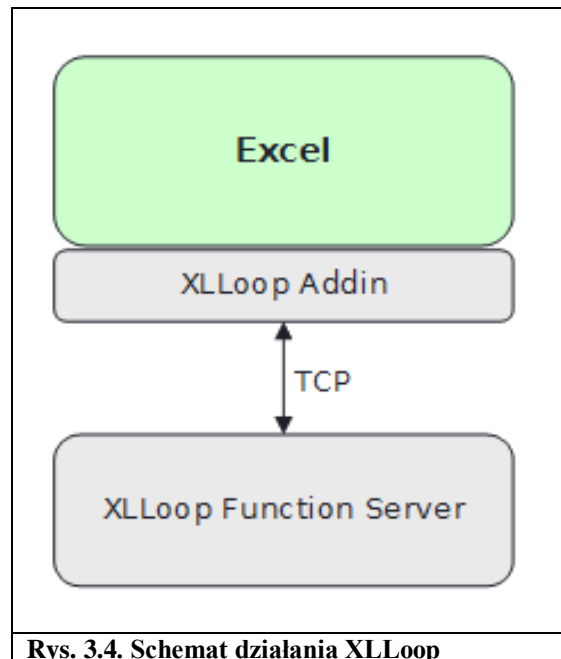
<sup>5</sup> <http://www.j-interop.org/introduction.html> [dostęp 2015-06-21]

### 3.3.2. Biblioteka XLLoop<sup>6</sup>

XLLoop to open sourceowa biblioteka pozwalająca implementować Excelowe funkcje zdefiniowane przez użytkownika ( UDF - User-Defined Functions) na zcentralizowanym serwerze (serwerze funkcyjnym).

#### 3.3.2.1. Zastosowanie

- Funkcje mogą być napisane w każdym języku (w tym Java, języki skryptowe itd.),
- Funkcje mogą być dodane szybko i dynamicznie bez konieczności restartowania Excela,
- Funkcje mogą być zarządzane oddzielnie i centralnie, co zapobiega kosztów zarządzania wieloma XLL'ami i gwarantuje wszystkim użytkownikom używanie tych samych funkcji,
- Dane mogą być dzielone pomiędzy excelowymi sesjami/użytkownikami ( np. obecne ceny giełdowe mogą być przechowywane na pojedynczym serwerze i wszystkie sesje Excela mogą uzyskiwać te dane za pomocą przykładowej funkcji GetStock).



#### 3.3.2.2. XLLoop składa się z dwóch głównych komponentów:

- Excelowy dodatek (XLL napisany w C++)
- Serwer i biblioteka napisana w Java ( i w wielu innych językach)

---

<sup>6</sup> <http://xlloop.sourceforge.net/> [dostęp 2015-11-12]



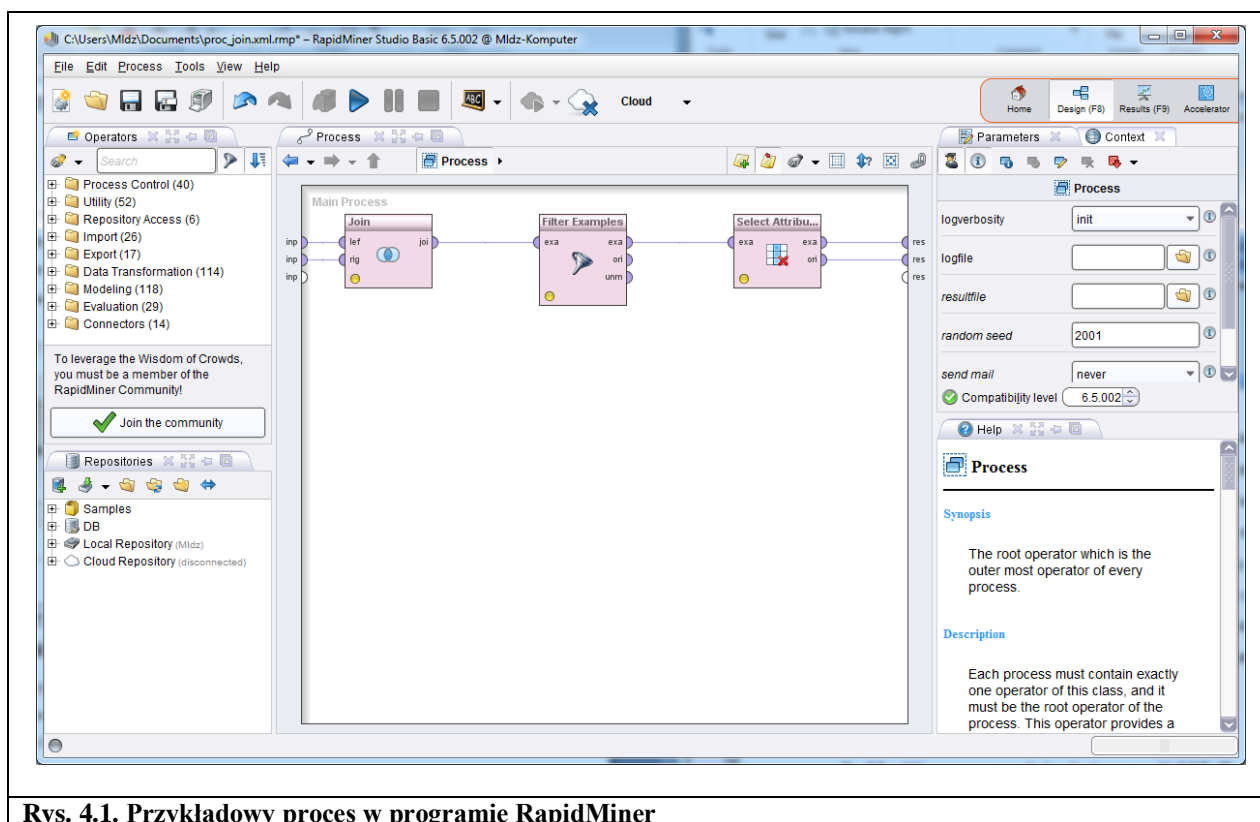
Dodatek i serwer komunikują się poprzez zwykłe oparte o schowek protokoły, wysyłające i odbierające (binarnie) zserializowane excelowe obiekty (nazywane xloper).

W modyfikacji JSON dodatek i serwer komunikują się ponad protokołami http(s), wysyłając i otrzymując JSON'owe zserializowane excelowe obiekty.

## 4. RapidMiner

RapidMiner to najczęściej używana open source'owa platforma do predykcyjnej analizy oraz eksploracji danych. Tworzony w nim kod generuje się za pośrednictwem intuicyjnego graficznego interfejsu użytkownika, który pomaga zaprojektować analizowany proces. RapidMiner posiada upublicznione biblioteki służące do rozszerzenia jego funkcjonalności, można je znaleźć na serwerze GitHub.com. Platforma jest otwarta do współpracy z mnóstwem źródeł danych w tym: Excel, Access, Oracle, IBM DB2, Microsoft SQL, Netezza Teradata, MySQL, Postgres, SPSS, Salesforce.com. RapidMiner umożliwia nowoczesną analizę w każdej skali, jest wręcz idealne dla Big Data.

RapidMiner jest programem, który nadaje się do współpracy z MS Excel i jest w stanie znacząco rozszerzyć jego możliwości. Jest napisany w Java i posiada możliwość wczytywania arkuszy Excel. Jednak w niniejszym projekcie funkcje Java mają być wykonywane z poziomu Excela - który potrafi obsługiwać przeciętny użytkownik.<sup>7</sup>

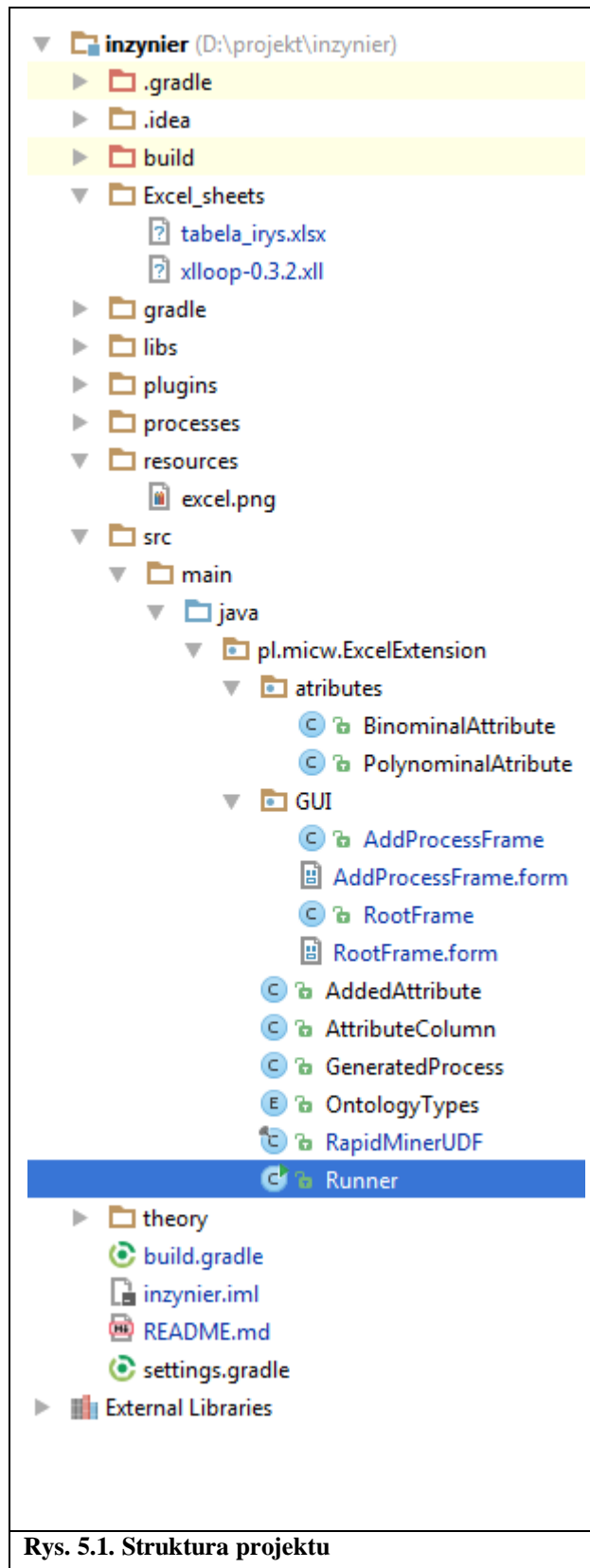


Rys. 4.1. Przykładowy proces w programie RapidMiner

<sup>7</sup> <https://rapidminer.com/> [dostęp 2015-11-29]

Procesy w RapidMinerze, można eksportować i importować w formacie RMP (format dedykowany) oraz XML co umożliwia znaczną uniwersalność jak również możliwość wykorzystania w różnych projektach.

## 5. Szczegóły implementacyjne projektu



Rys. 5.1. Struktura projektu

### 5.1. Budowa projektu

Projekt został stworzony w środowisku IntelliJ IDEA 15. Wykorzystuje Java SE Development Kit 8. Składa się z 9 klas i 1 typu wyliczeniowego. Posiada również 2 okna dialogowe:

- RootFrame - odpowiada, za główne menu programu, zawiera instrukcję obsługi, przyciski służące do dodawania i usuwania procesów oraz wyświetla aktualnie załadowane procesy,
- AddProcessFrame - odpowiada za dodanie nowego atrybutu przez użytkownika. Należy w nim podać nazwę projektu, lokalizację do procesu w formacie XML lub RMP oraz atrybuty (o tej samej nazwie i typie co atrybuty w projekcie RapidMiner'a).

Projekt zawiera 922 linie kodu napisanego w Java, 279 linii wygenerowanych przez pliki FORM i 269 linii w plikach Gradle. Aby zachować wysoką jakość kodu w projekcie użyto plugin'u CheckStyle i FindBugs, a analizy został użyty plugin Statistic.

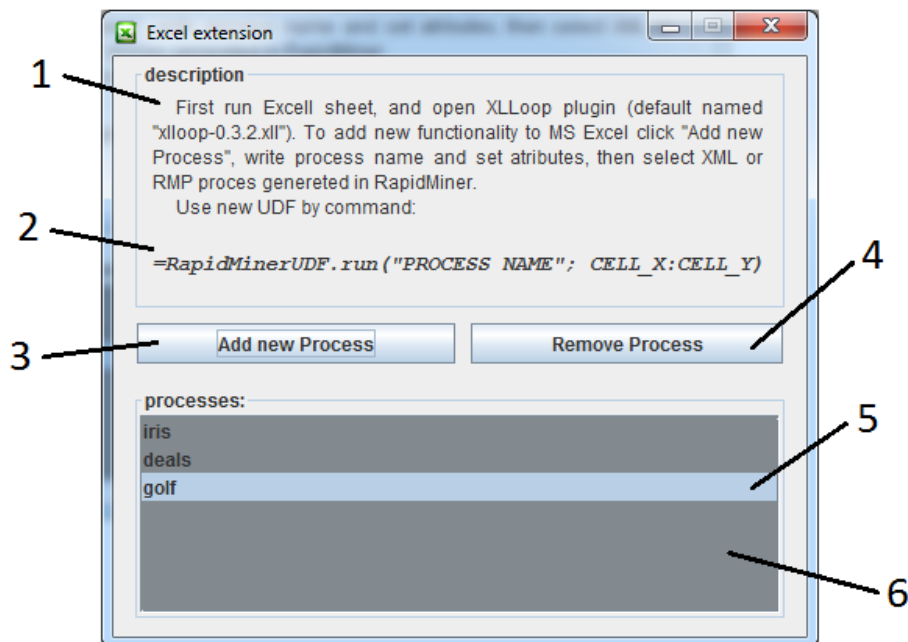
## 5.2. Wykorzystane Biblioteki

Do zbudowania projektu zostało użytych kilka plugin'ów napisanych w Gradle. Dzięki temu większość bibliotek została pobrana ze zdalnego repozytorium. Jedynie biblioteka core'owa do RapidMinera, jak również XLLoop musiała zostać ręcznie ściągnięta i ustawiona w repozytorium lokalnym. Pluginy Gradle'owe zostały pobrane ze strony projektu RapidMiner na Github.com. Główne biblioteki:

- XLLoop 0.3.2,
- Rapidminer-studio-core 6.5.2,
- Swing - interfejs graficzny,
- Awt - interfejs graficzny i obsługa zdarzeń,

## 5.3. Interfejs użytkownika - GUI

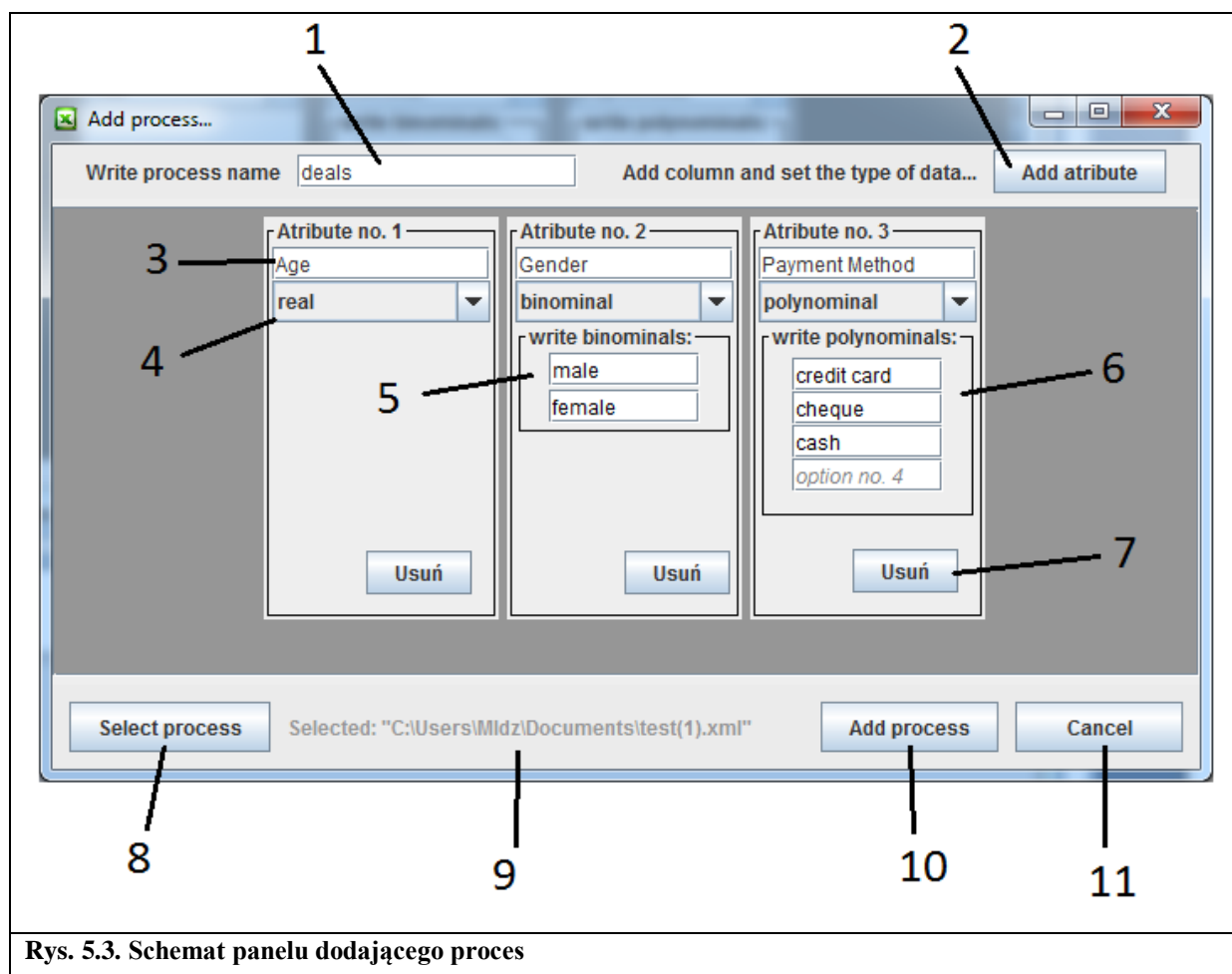
Poniżej w sposób schematyczny przedstawiono podstawowe elementy konfiguracyjne stworzonego interfejsu użytkownika.



Rys. 5.2. Schemat głównego panelu

Legenda do rys. 5.1

- 1) opis działania programu
- 2) funkcja wzorowa wywołująca dany skrypt w arkuszu Excel
- 3) przycisk służący do dodania nowego procesu
- 4) przycisk służący do usunięcia zaznaczonego procesu
- 5) zaznaczony proces
- 6) lista z aktualnie obsługiwanymi procesami

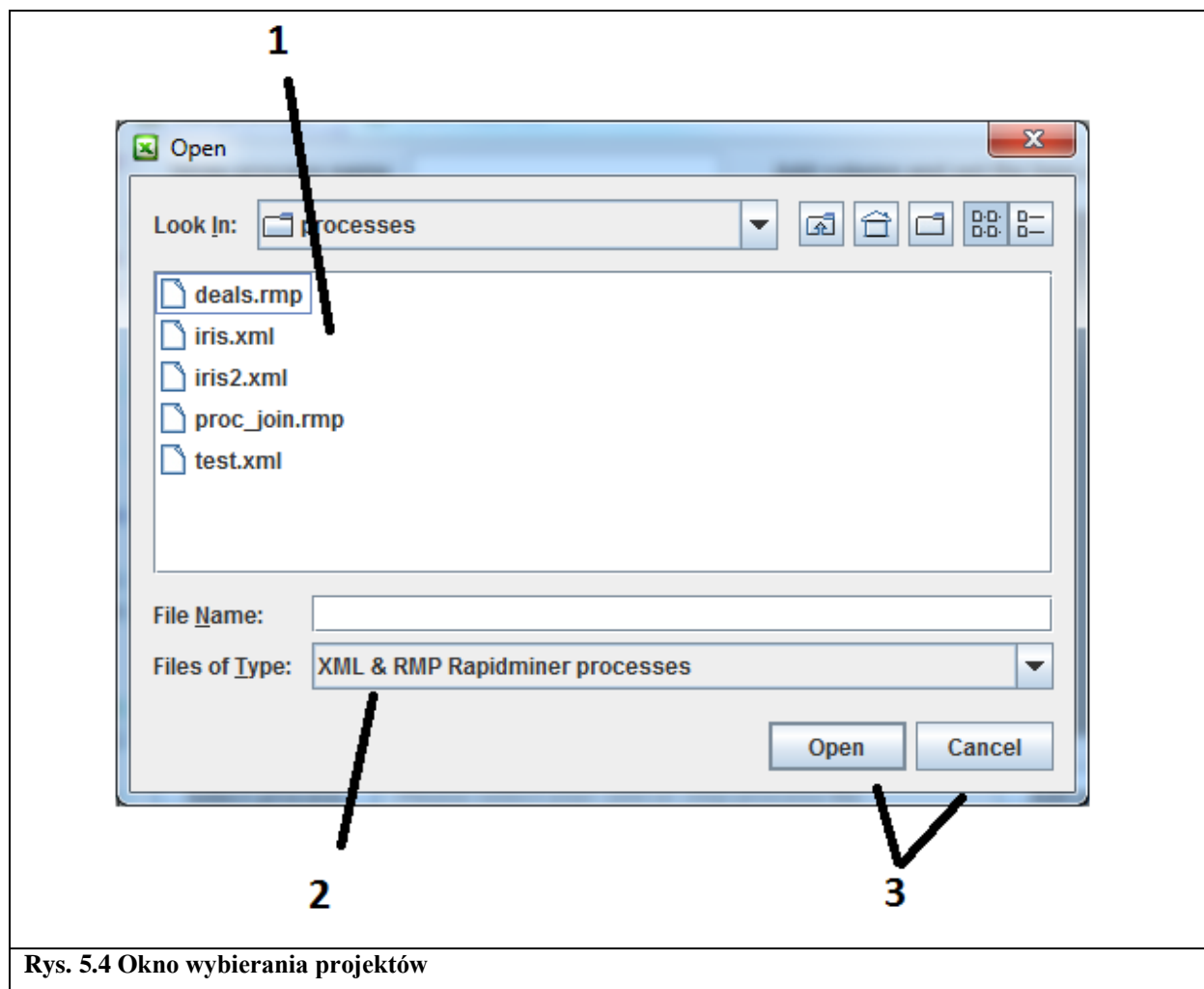


Rys. 5.3. Schemat panelu dodającego proces

Legenda do rys. 5.2:

- 1) nazwa procesu,
- 2) przycisk generujący kolejny atrybut,
- 3) nazwa atrybutu (musi pokrywać się z nazwą w procesie),
- 4) rodzaj danych
- 5) zdefiniowane wartości binominal,

- 6) zdefiniowane wartości polynominal (opcje kolejno się generują),
- 7) przycisk umożliwiający usunięcie całego atrybutu,
- 8) przycisk pozwalający wybrać lokalizację
- 9) ścieżka do wybranego procesu
- 10) przycisk dodający proces
- 11) przycisk anulujący



Rys. 5.4 Okno wybierania projektów

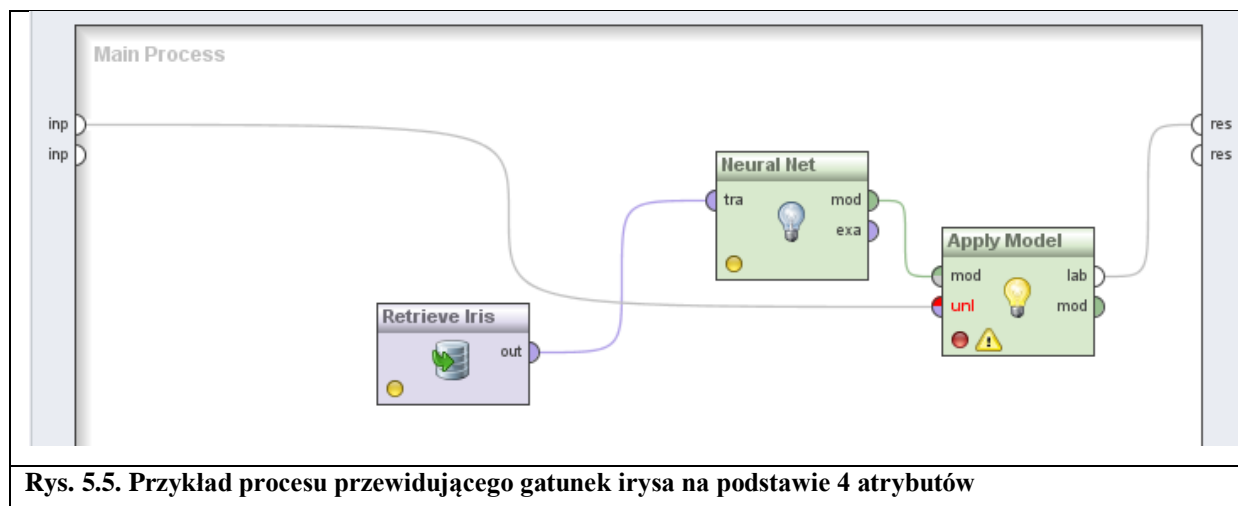
Legenda do rys 5.3:

- 1) dostępne procesy do wyboru
- 2) możliwe rozszerzenia procesu .xml oraz .rmp
- 3) przyciski wyboru

## 5.4. Tworzenie procesu w RapidMiner'ze

Tworzone procesy w RapidMiner'ze mogą posiadać wiele wejść. Dzięki bogatemu interfejsowi graficznemu użytkownik jest w stanie w łatwy sposób analizować przekazywane dane.

W poniższym przykładzie procesu sieć neuronowa jest uczona za pomocą zbioru danych "Retrieve Iris". Następnie dane przychodzące na wejście są analizowane przez sieć i funkcja "Apply Model" decyduje o wyniku predykcji.



Rys. 5.5. Przykład procesu przewidującego gatunek irysa na podstawie 4 atrybutów

ExampleSet (150 examples, 2 special attributes, 4 regular attributes)						
Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200

Rys. 5.6. Przykładowy zbiór danych uczących sieć neuronową

## 5.5. Sposób działania

Klasa Runner posiada statyczną metodę główną main która odpowiedzialna jest za uruchomienie programu. W początkowej fazie wykorzystywana jest biblioteka XLLoop, aby utworzyć serwer nasłuchujący przesyłane pakiety protokołu TCP przez dodatek uruchomiony w Excel'u.



W międzyczasie odpalany jest kolejny wątek odpowiedzialny za wywołanie głównego okna - RootFrame - w którym użytkownik może zdefiniować własne procesy. Należy pamiętać, że atrybuty w ustawianym procesie muszą być identyczne z atrybutami definiowanymi w procesie RapidMiner'a. Możliwe są 3 typy danych:

- real - wartości liczbowe,
- binominal - wartości dwuargumentowe np. typu prawda/fałsz,
- polynominal - typ wyliczeniowy np. rodzaje kolorów

Po zdefiniowaniu procesów, należy uruchomić arkusz MS Excel z danymi do przetworzenia i uruchomić dodatek "xlloop-0.3.2.xll". W tym momencie przesyłana jest do Excela metoda run z klasy RapidMinerUDF. Można ją wywołać przykładowo poprzez wywołanie:



Metoda run w pierwszej kolejności sprawdza nazwę procesu i pobiera ją z Listy wygenerowanych procesów. Dzięki danym w liście możliwe jest utworzenie ExampleSet'a, który obsługuje zbiór wejściowych danych do procesów. W przypadku wartości binominal i polynominal tworzone są mapy zamieniające wartość Integer na dany typ String. Po wygenerowaniu ExampleSet'a pobierane są wejścia procesu, a następnie pobieramy wygenerowany wynik. W przypadku wyniku w wielu komórkach należy użyć w arkuszu Excel kombinacji "ctrl + shift + enter" odpowiadającą za rozbudowane funkcje.

SUMA									
=RapidMinerUDF.run("iris";B8:E8)									
	A	B	C	D	E	F	G	H	I
1	id	a1	a2	a3	a4	Iris-setosa	Iris-versicolor	Iris-virginica	Prediction
2	id 1	4.4	2.9	1.4	0.2	0.99222224	0.007777447	3.1278E-07	Iris-setosa
3	id 2	4.9	3.1	1.5	0.1	0.993543304	0.006456431	2.6478E-07	Iris-setosa
4	id 3	5.4	3.7	1.5	0.2	0.994661609	0.005338168	2.2381E-07	Iris-setosa
5	id 4	4.8	3.4	1.6	0.2	0.993870395	0.006129351	2.5435E-07	Iris-setosa
6	id 5	6.3	2.9	5.6	1.8	8.49089E-05	0.004418658	0.99549643	Iris-virginica
7	id 6	5.6	2.5	3.9	1.1	0.007990046	0.990929293	0.00108066	Iris-versicolor
8	id 7	5.9	3.2	4.8	1.8	=RapidMinerUDF.run("iris";B8:E8)			
9	id 8	5.3	3.7	1.5	0.2				
10	id 9	5	3.3	1.4	0.2				
11	id 10	7	2.2	4.7	1.4				

Rys. 5.8. Przykład wywołania procesu iris w MS Excel 2007

```

1  Listening on port 5454...
2  pl.micw.ExcelExtension.GeneratedProcess{ pathToProcess="D:\projekt\inzynier\processes\iris2.xml",
3  attributes={ a1 - real a2 - real a3 - real a4 - real }, processName='iris'}
4  org.boris.xlloop.Initialize("Mldz", "Mldz-Komputer")
5  org.boris.xlloop.GetFunctions()
6  RapidMinerUDF.run
7  RapidMinerUDF.run("iris", [5.9, 3.2, 4.8, 1.8])

18 #5: confidence(Iris-setosa) (real/single_value):
19 confidence(Iris-setosa): 0.00
20 #6: confidence(Iris-versicolor) (real/single_value):
21 confidence(Iris-versicolor): 0.52
22 #7: confidence(Iris-virginica) (real/single_value):
23 confidence(Iris-virginica): 0.48
24 #4: prediction(label) (nominal/single_value)/values=[Iris-setosa, Iris-versicolor, Iris-virginica]:
25 Iris-versicolor

```

**Rys. 5.9. Kod programu przekazywany do konsoli.**

Linia 1 - program nasłuchuje portu nr 5454. Oczekuje na przesłanie pakietów TCP.

Linie 2-3 - został dodany proces przez użytkownika.

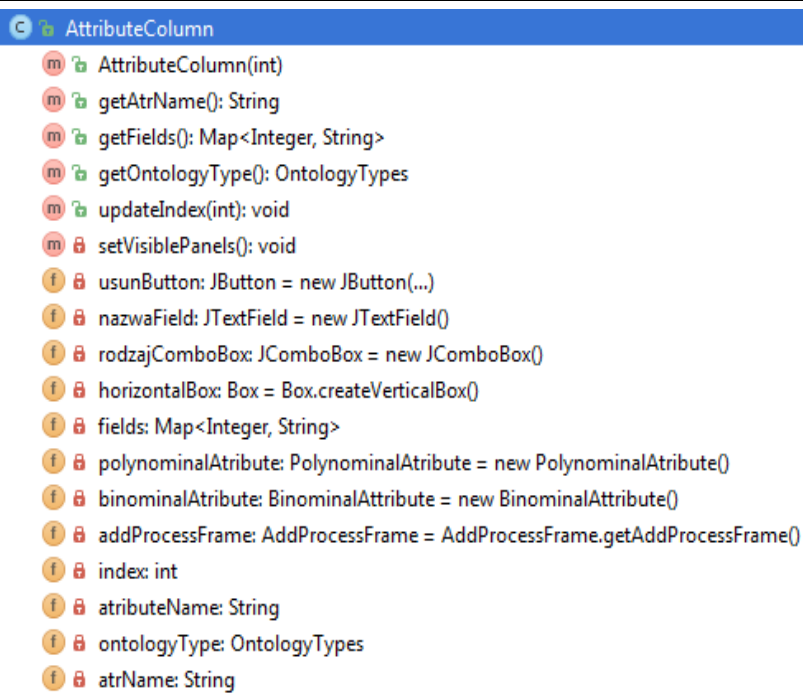

Linie 4-6 - Użytkownik włączył dodatek do Excela. Następuje inicjalizacja biblioteki xl-loop, a następnie przekazywana jest statyczna metoda RapidMinerUDF.run

Linia 7 - Została wywołana nowa funkcjonalność. Dane są przekazywane do aplikacji napisanej w Java i są poddane analizie

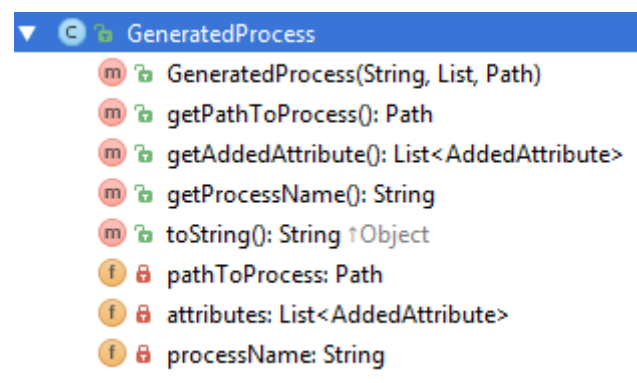
Linie 18-25 - dane zostają zwrócone wraz z predykcją

## 5.6. Główne klasy

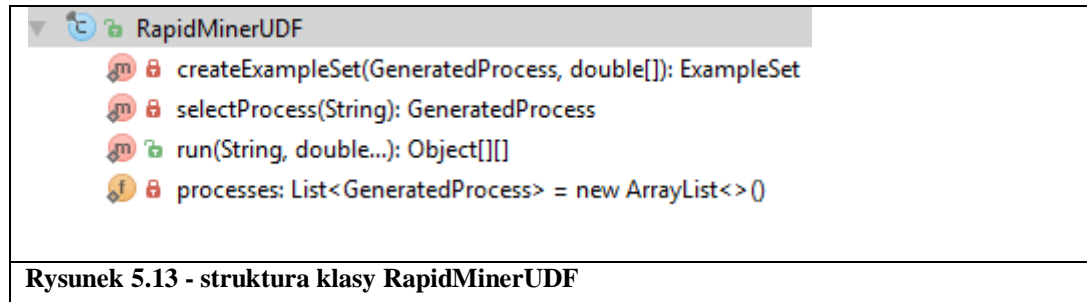
- Attribute column - klasa odpowiedzialna za tworzenie obiektu poszczególnego atrybutu. Posiada unikalny index i może przybierać różne typy (użyty został enum OntologyType). Jest częścią Listy w klasie AddProcessFrame. Przy ontologii binominal i polinomial uzupełniana jest mapa fields, aby poszczególnym opcjom przyporządkować odpowiedni index. Natomiast przy ontologii real mapa jest domyślnie pusta.

 <pre> AttributeColumn   m AttributeColumn(int)   m getAtrName(): String   m getFields(): Map&lt;Integer, String&gt;   m getOntologyType(): OntologyTypes   m updateIndex(int): void   m setVisiblePanels(): void   f usunButton: JButton = new JButton(...)   f nazwaField: JTextField = new JTextField()   f rodzajComboBox: JComboBox = new JComboBox()   f horizontalBox: Box = Box.createVerticalBox()   f fields: Map&lt;Integer, String&gt;   f polynominalAttribute: PolynominalAttribute = new PolynominalAttribute()   f binominalAttribute: BinominalAttribute = new BinominalAttribute()   f addProcessFrame: AddProcessFrame = AddProcessFrame.getAddProcessFrame()   f index: int   f attributeName: String   f ontologyType: OntologyTypes   f atrName: String </pre>	
<p><b>Rys. 5.10. Struktura klasy Attribute Column</b></p>	<p><b>Rys. 5.11. Attribute Column wygląd w GUI</b></p>

- GeneratedProcess - klasa odpowiedzialna za tworzenie obiektów skonfigurowanych projektów. Zawiera wszystkie kluczowe informacje niezbędne do uruchomienia danego procesu.

 <pre> GeneratedProcess   m GeneratedProcess(String, List, Path)   m getPathToProcess(): Path   m getAddedAttribute(): List&lt;AddedAttribute&gt;   m getProcessName(): String   m toString(): String   f pathToProcess: Path   f attributes: List&lt;AddedAttribute&gt;   f processName: String </pre>	<p><b>Rysunek 5.12 - struktura klasy GeneratedProcess</b></p>
---	---

- RapidMinerUDF - klasa, która przekazuje wszystkie swoje metody publiczne, statyczne do klasy ReflectFunctionHandler w XLLoop. Jediną taką metodą jest run. Reszta służy do obsługi tej metody.



Rysunek 5.13 - struktura klasy RapidMinerUDF

## 5.7. Sposób tworzenia

Biblioteka XLLoop wykorzystuje w dużej mierze programowanie proceduralne - najważniejsze metody są statyczne i są wywoływane w klasie Runner. Podobnie biblioteka Swing i AWT opierają się na programowaniu proceduralnym, gdzie poszczególne parametry określa się metodami typu get i set, a wartości przekazuje się przez metody statyczne.

W związku z tym podczas tworzenia projektu utrudnione było projektowanie obiektowe oraz wykorzystywanie powszechnie znanych wzorców projektowych.

## 6. Wnioski i podsumowanie

W informatyce bardzo często mamy do czynienia z kilkoma rozwiązaniami dotyczącymi tych samych problemów. Autorzy danych produktów, używając często własnych implementacji, co może dodatkowo utrudnić proces integracji. Z punktu widzenia developera, który musi korzystać z rozwiązań wielu technologii konieczna jest wiedza i doświadczenie w tej dziedzinie. Kluczowym zagadnieniem są protokoły służące do ujednolicenia komunikowanych danych.

Microsoft wprowadzając na rynek coraz nowsze wersje pakietu Office wciąż nie wspiera technologii Java, z powodów ekonomicznych. Jednak ich odpowiednik - C# nie jest w stanie w znaczący sposób wyróżnić się na tle innych języków wysokiego poziomu.

Każdy projekt różni się wytycznymi i w zależności od wymagań, architektury, funkcjonalności czy trendów wymaga zastosowania odpowiedniej technologii. Każdy projekt jest unikalny i należy do niego podchodzić indywidualnie szukając możliwie najprostszych i najbardziej efektywnych rozwiązań. Dlatego warto rozwijać się w wielu dziedzinach oraz poznawać alternatywne rozwiązania. Wraz z postępem czasu potrzeba integracji wielu systemów staje się coraz większa.

## 7. Literatura

- 1) Computerworld: Polski Word 2.0 i Excel 4.0 wydanie 2009-01-31
- 2) <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>, 2016-01-13
- 3) <https://www.java.com/pl/about/> , 2015-12-27
- 4) <http://computeraidedfinance.com/2011/11/26/what-shall-i-choose-for-implementing-an-excel-plugin-and-udf/> , 2015-08-12
- 5) <http://www.j-interop.org/introduction.html> , 2015-06-21
- 6) <http://xlloop.sourceforge.net/> , 2015-11-12
- 7) <https://rapidminer.com/> , 2015-11-29

## 8. Spis rysunków

- Rys. 1.1. - Ranking najbardziej popularnych języków programowania na świecie wg. serwisu TIOBE. Strona 3
- Rys. 3.1. Przykład użycia UDF o nazwie YahooHistory w Excel'u. Strona 4
- Rys. 3.2. Przykład wykorzystania makra. Strona 5
- Rys. 3.3. Schemat działania mostu j-interop pomiędzy Windows'owym COM, a serwerem COM. Strona 7
- Rys. 3.4. Schemat działania XLLoop. Strona 7
- Rys. 4.1. Przykładowy proces w programie RapidMiner. Strona 9
- Rys. 5.1. Struktura projektu. Strona 10
- Rys. 5.2. Schemat głównego panelu. Strona 11
- Rys. 5.3. Schemat panelu dodającego proces. Strona 12
- Rys. 5.4 Okno wybierania projektów. Strona 13
- Rys. 5.5. Przykład procesu przewidującego gatunek irysa na podstawie 4 atrybutów. Strona 14
- Rys. 5.6. Przykładowy zbiór danych uczących sieć neuronową. Strona 14
- Rys. 5.7. Przykład wywołania funkcji iris dla argumentów B8:E8. Strona 15
- Rys. 5.8. Przykład wywołania procesu iris w MS Excel 2007. Strona 15
- Rys. 5.9. Kod programu przekazywany do konsoli. Strona 16
- Rys. 5.10. Struktura klasy Attribute Column. Strona 17
- Rys. 5.11. Attribute Column wygląd w GUI. Strona 17
- Rys. 5.12. struktura klasy GeneratedProcess. Strona 17
- Rys. 5.13. struktura klasy RapidMinerUDF. Strona 18