

METODY I NARZĘDZIA GENEROWANIA KODU WYKONYWALNEGO

ppor. mgr inż. Norbert Waszkowiak

METODY I NARZĘDZIA GENEROWANIA KODU WYKONYWALNEGO

ppor. mgr inż. Norbert Waszkowiak

nr tel.: 261 839 060

e-mail: norbert.waszkowiak@wat.edu.pl

pomieszczenie: bud. 100 pok. 254B

(zapraszam również do kontaktu przez MS Teams)

PRZEPLANOWANIE ZAJĘĆ

ZASADY OCENIANIA

1. Każde przedstawione zadanie podlega ocenie 2-5;
2. W przypadku stwierdzenia niesamodzielności w wykonaniu pracy zostanie ona wliczona do średniej jako 0;
3. Ostateczna ocena z laboratoriów będzie wynikała ze średniej z wszystkich zadań laboratoryjnych;
4. Warunkiem zaliczenia jest oddanie wszystkich zadań laboratoryjnych w ustalonym terminie dla każdego zadania.

ZADANIE 1: WYRAŻENIA REGULARNE

A KOMU TO POTRZEBNE? A DLACZEGO?

Wyrażenia regularne - (ang. regular expression, w skrócie regex lub regexp) - wzorzec opisujący łańcuch symboli. Teoria wyrażeń regularnych jest związana z teorią języków regularnych.

Wyrażenia regularne mogą określać zbiór pasujących łańcuchów, jak również wyszczególniać istotne części łańcucha.

W praktyce znalazły bardzo szerokie zastosowanie, pozwalają bowiem w łatwy sposób opisywać wzorce tekstu, natomiast istniejące algorytmy w efektywny sposób określają, czy podany ciąg znaków pasuje do wzorca lub wyszukują w tekście wystąpienia wzorca. Wyrażenia regularne w praktycznych zastosowaniach są zapisywane za pomocą bogatszej i łatwiejszej w użyciu składni niż ta stosowana w rozważaniach teoretycznych.

POKAZ Z OBJAŚNIENIEM

Przykład z grupami dziekańskimi

REGULAR EXPRESSION

v1 ~ 143 matches (14.7ms)

" (?<group>\w{3})(?<year>\d{2})((?<id>\w{2,5})(?<type>(N|S))?(?<degree>\d)|(\w*)(?<chase>\w{2,5})) " gm

TEST STRING

WCY18IX1S3 ; -WCY19IG1S1 ; -WCY19IJ1N1 ; -WCY19IJ1S1 ; -WCY19IJ2S1 ; -WCY19IJ3S1 ; -WCY19IJ4S1 ; -WCY19IL1S0 ; -WCY19IT1S0 ; -WCY19IW1S0 ; -WCY19KA1S0 ; -WCY19KB1S0 ; -WCY19KC1S0 ; -WCY19KC1S1 ; -WCY19KS1S0 ; -WCY19KS1S1 ; -WCY20IG1S1 ; -WCY20IJ1N1 ; -WCY20IJ1S1 ; -WCY20IJ2S1 ; -WCY20IJ3S1 ; -WCY20IM1S1 ; -WCY20IX1S0 ; -WCY20IX2S0 ; -WCY20IX3S0 ; -WCY20IX4S0 ; -WCY20IX5S0 ; -WCY20IX6S0 ; -WCY20IY1S1 ; -WCY20IY2S1 ; -WCY20IY3S1 ; -WCY20IY4S1 ; -WCY20IY5S1 ; -WCY20KA1S1 ; -WCY20KC1S1 ; -WCY20KX1S0 ; -WCY20KX2S0 ; -WCY20KX3S0 ; -WCY20KX4S0 ; -WCY20KY1S1 ; -WCY20KY2S1 ; -WCY20KY3S1 ; -WCY21E7IN5 ; -WCY21I*BDa ; -WCY21I*BDb ; -WCY21I*S0 ; -WCY21I*Tp1 ; -WCY21IB1S4 ; -WCY21IB2S4 ; -WCY21IE1S4 ; -WCY21IH1S4 ; -WCY21IM1S4 ; -WCY21IV1S4 ; -WCY21IX1N1 ; -WCY21IX1S0 ; -WCY21IX2N1 ; -WCY21IX2S0 ; -WCY21IX3S0 ; -WCY21IX4S0 ; -WCY21IX5S0 ; -WCY21IX6S0 ; -WCY21IY1S1 ; -WCY21IY2S1 ; -WCY21IY3S1 ; -WCY21IY4S1 ; -WCY21K*SAa ; -WCY21K*SAb ; -WCY21KB1S4 ; -WCY21KB2S4 ; -WCY21KB3S4 ; -WCY21KC1S4 ; -WCY21KS1S4 ; -WCY21KS2S4 ; -WCY21KT1S4 ; -WCY21KX1S0 ; -WCY21KX2S0 ; -WCY21KX3S0 ; -WCY21KX4S0 ; -WCY21KX5S0 ; -WCY21KX6S0 ; -WCY21KY1S1 ; -WCY21KY2S1 ; -WCY22IA2aS6 ; -WCY22IA2bS6 ; -WCY22IB1N2 ; -WCY22IB2N2 ; -WCY22IB3aS6 ; -WCY22IB3bS6 ; -WCY22IB0S6 ; -WCY22IE3aS6 ; -WCY22IE3bS6 ; -WCY22IX1N1 ; -WCY22IX1N5 ; -WCY22IX1S0 ; -WCY22IX2N1 ; -WCY22IX2N5 ; -WCY22IX2S0 ; -WCY22IX3N1 ; -WCY22IX3S0 ; -WCY22IX4S0 ; -WCY22IX5S0 ; -WCY22IY1S1 ; -WCY22IY2S1 ; -WCY22IY3S1 ; -WCY22IY4S1 ; -WCY22IY5S1 ; -WCY22IZ3aS6 ; -WCY22IZ3bS6 ; -WCY22KX1S0 ; -WCY22KX2S0 ; -WCY22KX3S0 ; -WCY22KX4S0 ; -WCY22KX5S0 ; -WCY22KX6S0 ; -WCY22KX7S0 ; -WCY22KY1S1 ; -WCY22KY2S1 ; -WCY22S03S6 ; -WCY22X*BBDa ; -WCY22X*BDa ; -WCY22X*Fiz2a ; -WCY22X*Fiz2b ; -WCY22X*M1a ; -WCY22X*M2a ; -WCY22X*M2b ; -WCY22X*Md1a ; -WCY22X*Tiika ; -WME19BC1S1 ; -WME19BM1S1 ; -WME19BO1S1 ; -WME20BE1S1 ; -WME20BM1S1 ; -WME20BO1S1 ; -WME21BM1S4 ; -WME21BX1S1 ; -WME21BX2S1 ; -WME21BX3S1 ; -WME21BX4S1 ; -WME22B1N5 ; -WME22BX1S1 ; -WME22BX2S1 ; -WME22BX3S1 ; -WME22BX4S1

EXPLANATION

" (?<group>\w{3})(?<year>\d{2})((?<id>\w{2,5})(?<type>(N|S))?(?<degree>\d)|(\w*)(?<chase>\w{2,5})) " gm

Named Capture Group group (?<group>\w{3})

\w matches any word character (equivalent to [a-zA-Z0-9_])

{3} matches the previous token exactly 3 times

Named Capture Group year (?<year>\d{2})

\d matches a digit (equivalent to [0-9])

{2} matches the previous token exactly 2 times

3rd Capturing Group

(?<id>\w{2,5})(?<type>(N|S))(?<degree>\d)|(\w*)(?<chase>\w{2,5}))

1st Alternative

(?<id>\w{2,5})(?<type>(N|S))(?<degree>\d)

Named Capture Group id (?<id>\w{2,5})

\w matches any word character (equivalent to [a-zA-Z0-9_])

{2,5} matches the previous token between 2 and 5 times, as many times as possible, giving back as

MATCH INFORMATION

Match 1

0-10

WCY18IX1S3

Group group

0-3

WCY

Group year

3-5

18

Group 3

5-10

IX1S3

Group id

5-8

IX1

Group type

8-9

S

Group 6

8-9

S

Group degree

9-10

3

Match 2

13-23

WCY19IG1S1

Group group

13-16

WCY

Group year

16-18

19

Dane:

WCY18IX1S3 ; WCY19IG1S1 ; WCY19IJ1N1 ; WCY19IJ1S1 ; WCY19IJ2S1 ; WCY19IJ3S1 ; WCY19IJ4S1 ; WCY19IL1S0 ; WCY19IT1S0 ; WCY19IW1S0 ; WCY19KA1S0 ; WCY19KB1S0 ; WCY19KC1S0 ; WCY19KC1S1 ; WCY19KS1S0 ; WCY19KS1S1 ; WCY20IG1S1 ; WCY20IJ1N1 ; WCY20IJ1S1 ; WCY20IJ2S1 ; WCY20IJ3S1 ; WCY20IM1S1 ; WCY20IX1S0 ; WCY20IX2S0 ; WCY20IX3S0 ; WCY20IX4S0 ; WCY20IX5S0 ; WCY20IX6S0 ; WCY20IY1S1 ; WCY20IY2S1 ; WCY20IY3S1 ; WCY20IY4S1 ; WCY20IY5S1 ; WCY20KA1S1 ; WCY20KC1S1 ; WCY20KX1S0 ; WCY20KX2S0 ; WCY20KX3S0 ; WCY20KX4S0 ; WCY20KY1S1 ; WCY20KY2S1 ; WCY20KY3S1 ; WCY21E7IN5 ; WCY21I*BDa ; WCY21I*BDb ; WCY21I*S0 ; WCY21I*Tp1 ; WCY21IB1S4 ; WCY21IB2S4 ; WCY21IE1S4 ; WCY21IH1S4 ; WCY21IM1S4 ; WCY21IV1S4 ; WCY21IX1N1 ; WCY21IX1S0 ; WCY21IX2N1 ; WCY21IX2S0 ; WCY21IX3S0 ; WCY21IX4S0 ; WCY21IX5S0 ; WCY21IX6S0 ; WCY21IY1S1 ; WCY21IY2S1 ; WCY21IY3S1 ; WCY21IY4S1 ; WCY21K*SAa ; WCY21K*SAb ; WCY21KB1S4 ; WCY21KB2S4 ; WCY21KB3S4 ; WCY21KC1S4 ; WCY21KS1S4 ; WCY21KS2S4 ; WCY21KT1S4 ; WCY21KX1S0 ; WCY21KX2S0 ; WCY21KX3S0 ; WCY21KX4S0 ; WCY21KX5S0 ; WCY21KX6S0 ; WCY21KY1S1 ; WCY21KY2S1 ; WCY22IA2aS6 ; WCY22IA2bS6 ; WCY22IB1N2 ; WCY22IB2N2 ; WCY22IB3aS6 ; WCY22IB3bS6 ; WCY22IB0S6 ; WCY22IE3aS6 ; WCY22IE3bS6 ; WCY22IX1N1 ; WCY22IX1N5 ; WCY22IX1S0 ; WCY22IX2N1 ; WCY22IX2N5 ; WCY22IX2S0 ; WCY22IX3N1 ; WCY22IX3S0 ; WCY22IX4S0 ; WCY22IX5S0 ; WCY22IY1S1 ; WCY22IY2S1 ; WCY22IY3S1 ; WCY22IY4S1 ; WCY22IY5S1 ; WCY22IZ3aS6 ; WCY22IZ3bS6 ; WCY22KX1S0 ; WCY22KX2S0 ; WCY22KX3S0 ; WCY22KX4S0 ; WCY22KX5S0 ; WCY22KX6S0 ; WCY22KX7S0 ; WCY22KY1S1 ; WCY22KY2S1 ; WCY22S03S6 ; WCY22X*BBDa ; WCY22X*BDa ; WCY22X*Fiz2a ; WCY22X*Fiz2b ; WCY22X*M1a ; WCY22X*M2a ; WCY22X*M2b ; WCY22X*Md1a ; WCY22X*Tiika ; WME19BC1S1 ; WME19BM1S1 ; WME19BO1S1 ; WME20BE1S1 ; WME20BM1S1 ; WME20BO1S1 ; WME21BM1S4 ; WME21BX1S1 ; WME21BX2S1 ; WME21BX3S1 ; WME21BX4S1 ; WME22B1N5 ; WME22BX1S1 ; WME22BX2S1 ; WME22BX3S1 ; WME22BX4S1

POKAZ Z OBJAŚNIENIEM

Przykład z cytatem

REGULAR EXPRESSION

v1 1 match (1.2ms)

" \<div\sclass\=\"quotes__single\s\">.*quotes__singleVerticalCenter.*<p>\s*(?<cite>\S[^<]*\S)\s*<

TEST STRING

```
<div class="quotes__single">
...<div class="row">
...
...<div class="d-flex flex-column justify-content-center col-lg-2">
...
...<div class="authorAllBooks__singleImg d-none d-lg-block">
...
...<div class="quotes__char"></div>
...</div>
...
...<div class="col pl-2 pl-sm-0">
...<div class="quotes__singleVerticalCenter quotes__content">
...<div class="quotes__singleText">
...
...<div class="partial-collapse collapse"
id="comment56677" aria-expanded="false" style="height: 120px;">
...<div class="collapse-content">
...<p>
...Stąd wiedział, że deklarację o
bezpieczeństwie portali można było umieścić w tej samej przegródce, co
twierdzenia: mój piesek nie gryzie, mój synek to dobry chłopiec, bigos
jest świeży, pieniądze dam najdalej pojutrze, noc spędziłam u
przyjaciółki, na sercu leży mi wyłącznie dobro ojczyzny oraz odpowiesz
tylko na kilka pytań i zaraz cię zwolnimy.
...</p>
...</div>
...</div>
...<div class="d-flex justify-content-between">
...<button type="button" aria-label="Więcej"
class="btn btn-link read-more collapsed" data-toggle="collapse" data-
target="#comment56677" aria-expanded="false">
...<img class="ml-2"
...</div>
...</div>
```

EXPLANATION

" \<div\sclass\=\"quotes__single\s\">.*quotes__singleVerticalCenter.*<p>\s*(?<cite>\S[^<]*\S)\s*<

- \< matches the character < with index 60₁₀ (3C₁₆ or 74₈) literally (case sensitive)
- div matches the characters div literally (case sensitive)
- \s matches any whitespace character (equivalent to [\r\n\t\f\v\])
- class matches the characters class literally (case sensitive)
- = matches the character = with index 61₁₀ (3D₁₆ or 75₈) literally (case sensitive)
- " matches the character " with index 34₁₀ (22₁₆ or 42₈) literally (case sensitive)
- quotes__single matches the characters quotes__single literally (case sensitive)
- \s matches any whitespace character (equivalent to [\r\n\t\f\v\])
- " matches the character " with index 34₁₀ (22₁₆ or 42₈) literally (case sensitive)

MATCH INFORMATION

Match 1	0-1096	<div class="quotes__single">...<div class="row">...<div class="d-flex flex-column justify-content-center col-lg-2">...
Group cite	719-1067	Stąd wiedział, że deklarację o bezpieczeństwie portali można było umieścić w tej samej przegródce, c...

WPROWADZENIE W SYTUACJE TAKTYCZNA

Firma "S" utrzymująca system do zarządzania danymi badawczymi, publikacjami i pracownikami naukowymi. Pozyskała nowego klienta, który dotychczas korzystał z innego rozwiązania i zaszła potrzeba migracji danych z poprzedniego systemu. W systemie firmy "S" w encji opisującej publikacje istnieją osobne pola oznaczające nr artykułu, nr wydania, strony, wydawca itp. itd. W starym systemie wszystkie te pola znajdują się w dwóch polach tekstowych, aby wyeksportować dane z tych pól, należy skorzystać z wyrażeń regularnych.

ZADANIE

Z zastosowaniem [dobrych praktyk programistycznych](#) (lub [to](#)) zaimplementuj program korzystający z wyrażeń regularnych, który dostarczy interfejs umożliwiający rozdzielenie danych szczegółowych z dwóch dostarczonych pól tekstowych. Potwierdzeniem działania programu mają być testy jednostkowe przygotowane w oparciu o przykładowe dane.

[Plik csv](#) podzielony jest na następujące kolumny:

1. publisher - dane wej.
2. details - dane wej.
3. no - dane wyj.
4. vol - dane wyj.
5. article no - dane wyj.
6. pages in range - dane wyj.
7. pages as size - dane wyj.
8. publisher name - dane wyj.
9. publisher location - dane wyj.
10. publisher year - dane wyj.

Separator: "\$"

ZADANIE

Z zastosowaniem **dobrych praktyk programistycznych** (lub **to**) zaimplementuj program korzystający z wyrażeń regularnych, który dostarczy interfejs umożliwiający rozdzielenie danych szczegółowych z dwóch dostarczonych pól tekstowych. Potwierdzeniem działania programu mają być testy jednostkowe przygotowane w oparciu o przykładowe dane.

Plik csv podzielony jest na następujące kolumny:

1. **publisher** - dane wej.
2. **details** - dane wej.
3. **no** - dane wyj.
4. **vol** - dane wyj.
5. **article no** - dane wyj.
6. **pages in range** - dane wyj.
7. **pages as size** - dane wyj.
8. **publisher name** - dane wyj.
9. **publisher location** - dane wyj.
10. **publisher year** - dane wyj.

Separator: "\$"

ZADANIE

Z zastosowaniem **dobrych praktyk programistycznych** (lub **to**) zaimplementuj program korzystający z wyrażeń regularnych, który dostarczy interfejs umożliwiający rozdzielenie danych szczegółowych z dwóch dostarczonych pól tekstowych. Potwierdzeniem działania programu mają być testy jednostkowe przygotowane w oparciu o przykładowe dane.

Plik **csv** podzielony jest na następujące kolumny:

1. **publisher** - dane wej.
2. **details** - dane wej.
3. **no** - dane wyj.
4. **vol** - dane wyj.
5. **article no** - dane wyj.
6. **pages in range** - dane wyj.
7. **pages as size** - dane wyj.
8. **publisher name** - dane wyj.
9. **publisher location** - dane wyj.
10. **publisher year** - dane wyj.

Separator: "\$"

ZADANIE

Z zastosowaniem **dobrych praktyk programistycznych** (lub **to**) zaimplementuj program korzystający z wyrażeń regularnych, który dostarczy interfejs umożliwiający rozdzielenie danych szczegółowych z dwóch dostarczonych pól tekstowych. Potwierdzeniem działania programu mają być testy jednostkowe przygotowane w oparciu o przykładowe dane.

Plik **csv** podzielony jest na następujące kolumny:

1. **publisher** - dane wej.
2. **details** - dane wej.
3. **no** - dane wyj.
4. **vol** - dane wyj.
5. **article no** - dane wyj.
6. **pages in range** - dane wyj.
7. **pages as size** - dane wyj.
8. **publisher name** - dane wyj.
9. **publisher location** - dane wyj.
10. **publisher year** - dane wyj.

Separator: "\$"

ZASADY OCENIANIA

Ocenić będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

ZASADY OCENIANIA

Ocenić będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

Program należy wykonać w jednym z następujących języków: Java, Kotlin, Scala, Python, C#, C++

ZASADY OCENIANIA

Ocenie podlegać będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

Program należy wykonać w jednym z następujących języków: Java, Kotlin, Scala, Python, C#, C++

Do końca trwania laboratoriów:

1. przesłać za pomocą [formularza](#) link do utworzonego **publicznego** repozytorium na github.com
2. przesłać za pomocą [formularza](#) opracowane wyrażenia regularne. Formularz dopuszcza dowolną liczbę wypełnień. Czas przesłania i liczba odpowiedzi posłużą do oceny samodzielności wykonania.

Termin oddania pracy: 10.11.2022r. godz: 24:00

ZASADY OCENIANIA

Ocenić będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

Program należy wykonać w jednym z następujących języków: Java, Kotlin, Scala, Python, C#, C++

Do końca trwania laboratoriów:

1. przesłać za pomocą [formularza](#) link do utworzonego **publicznego** repozytorium na github.com
2. przesłać za pomocą [formularza](#) opracowane wyrażenia regularne. Formularz dopuszcza dowolną liczbę wypełnień. Czas przesłania i liczba odpowiedzi posłużą do oceny samodzielności wykonania.

Termin oddania pracy: 10.11.2022r. godz: 24:00

ZADANIE 2: KALKULATOR ANTRL

CO TO JEST I DO CZEGO SŁUŻY?

ANTLR (ang. ANother Tool for Language Recognition) to narzędzie służące do tworzenia kompilatorów oraz translatorów z opisu gramatyki zawierającego akcje w języku Java, C++, C# lub Python.

Domyślnie ANTLR generuje lekser i parser w Javie, a plik z gramatyką ma rozszerzenie .g

[Dokumentacja](#)

WZOROWY POKAZ

Gramatyki antlr4

```
wget https://raw.githubusercontent.com/antlr/grammars-v4/master/cpp/CPP14Lexer.g4
wget https://raw.githubusercontent.com/antlr/grammars-v4/master/cpp/CPP14Parser.g4
wget https://www.antlr.org/download/antlr-4.11.1-complete.jar
nano example.cpp
java -jar ./antlr-4.11.1-complete.jar CPP14Lexer.g4 CPP14Parser.g4
java -cp ./antlr-4.11.1-complete.jar org.antlr.v4.gui.TestRig CPP14 translationUnit -tree -gui example
```

```
#include <iostream>
int main()
{
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```

WZOROWY POKAZ

```
1 grammar Calculator;  
2  
3 expr: op=('*' | '/') expr expr # MulDiv  
4     | op=('+' | '-') expr expr # AddSub  
5     | INT                        # Int  
6     ;  
7  
8 INT: [0-9]+ ;  
9 DOT: '.' ;  
10 TIMES: '*' ;  
11 DIV: '/' ;  
12 PLUS: '+' ;  
13 MINUS: '-' ;  
14 WS : [ \t\r\n]+ -> skip ;
```

WZOROWY POKAZ

```
1 grammar Calculator;  
2  
3 expr: op=('*' | '/') expr expr # MulDiv  
4     | op=('+' | '-') expr expr # AddSub  
5     | INT                        # Int  
6     ;  
7  
8 INT: [0-9]+ ;  
9 DOT: '.' ;  
10 TIMES: '*' ;  
11 DIV: '/' ;  
12 PLUS: '+' ;  
13 MINUS: '-' ;  
14 WS : [ \t\r\n]+ -> skip ;
```


WZOROWY POKAZ

```
1 grammar Calculator;  
2  
3 expr: op=('*' | '/') expr expr # MulDiv  
4     | op=('+' | '-') expr expr # AddSub  
5     | INT                        # Int  
6     ;  
7  
8 INT: [0-9]+ ;  
9 DOT: '.' ;  
10 TIMES: '*' ;  
11 DIV: '/' ;  
12 PLUS: '+' ;  
13 MINUS: '-' ;  
14 WS : [ \t\r\n]+ -> skip ;
```

WZOROWY POKAZ

```
1 grammar Calculator;  
2  
3 expr: op=('*' | '/') expr expr # MulDiv  
4     | op=('+' | '-') expr expr # AddSub  
5     | INT                        # Int  
6     ;  
7  
8 INT: [0-9]+ ;  
9 DOT: '.' ;  
10 TIMES: '*' ;  
11 DIV: '/' ;  
12 PLUS: '+' ;  
13 MINUS: '-' ;  
14 WS : [ \t\r\n]+ -> skip ;
```

WZOROWY POKAZ

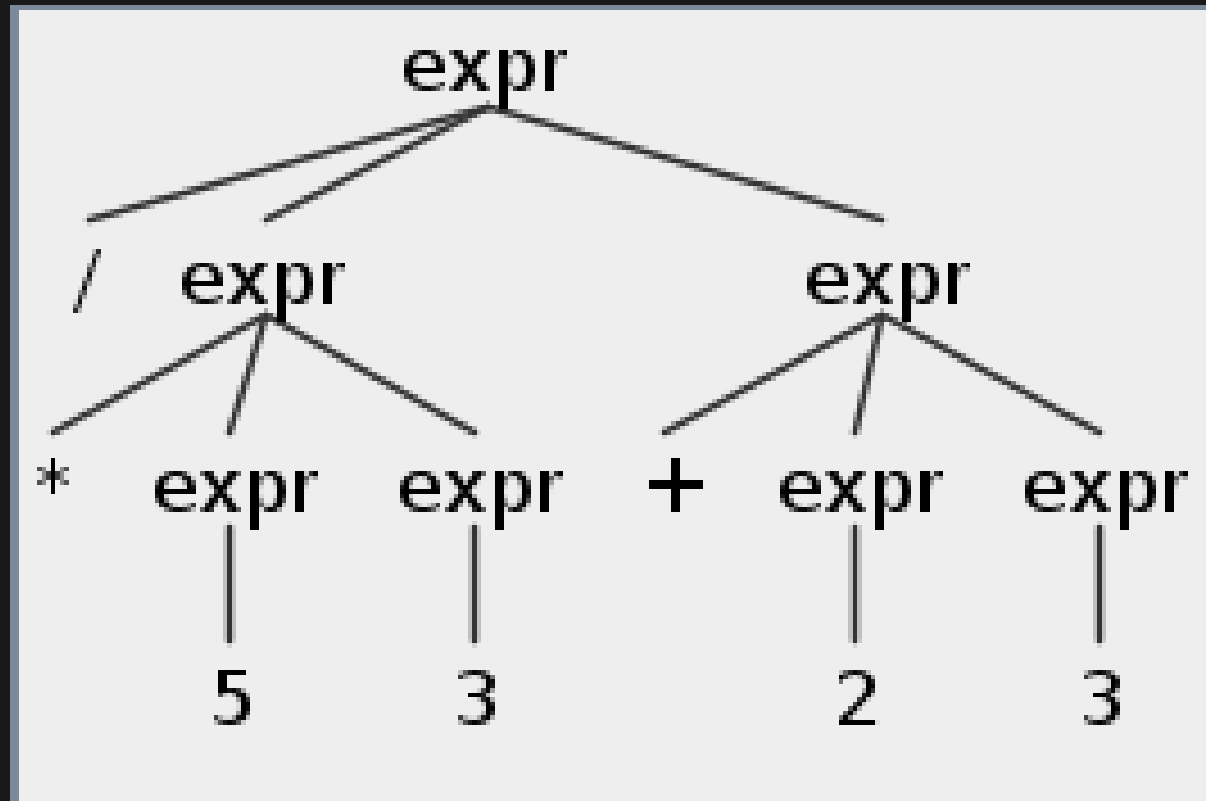
```
1 grammar Calculator;
2
3 expr: op=('*' | '/') expr expr # MulDiv
4     | op=('+' | '-') expr expr # AddSub
5     | INT # Int
6     ;
7
8 INT: [0-9]+ ;
9 DOT: '.' ;
10 TIMES: '*' ;
11 DIV: '/' ;
12 PLUS: '+' ;
13 MINUS: '-' ;
14 WS : [ \t\r\n]+ -> skip ;
```

WZOROWY POKAZ

```
1 grammar Calculator;  
2  
3 expr: op=('*' | '/') expr expr # MulDiv  
4     | op=('+' | '-') expr expr # AddSub  
5     | INT                        # Int  
6     ;  
7  
8 INT: [0-9]+ ;  
9 DOT: '.' ;  
10 TIMES: '*' ;  
11 DIV: '/' ;  
12 PLUS: '+' ;  
13 MINUS: '-' ;  
14 WS : [ \t\r\n]+ -> skip ;
```

WZOROWY POKAZ

/ * 5 3 + 2 3



ZADANIE

1. Utworzyć publiczne repozytorium na github.com. Zmieścić link w [formularzu](#)
2. Do końca zajęć laboratoryjnych za pomocą publicznie dostępnych gramatyk (Java, Python lub C#), wygenerować za pomocą ANTRL4 drzewo składniowe dla własnego przykładowego kodu składającego się z przynajmniej jednej pętli. Cały proces należy udokumentować i zamieścić w repozytorium w postaci pliku zapisanego w języku Markdown.
3. Opracować gramatykę języka - kalkulatora. Zaimplementować język w wykorzystując ANTRL4.

ZASADY OCENIANIA

Ocenie podlegać będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zakres funkcjonalności kalkulatora;
4. Zastosowanie dobrych praktyk programistycznych;
5. Wykorzystanie systemu kontroli wersji.

Termin oddania pracy: 15.11.2022r. godz: 24:00

ZASADY OCENIANIA

Ocenie podlegać będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zakres funkcjonalności kalkulatora;
4. Zastosowanie dobrych praktyk programistycznych;
5. Wykorzystanie systemu kontroli wersji.

Termin oddania pracy: 15.11.2022r. godz: 24:00

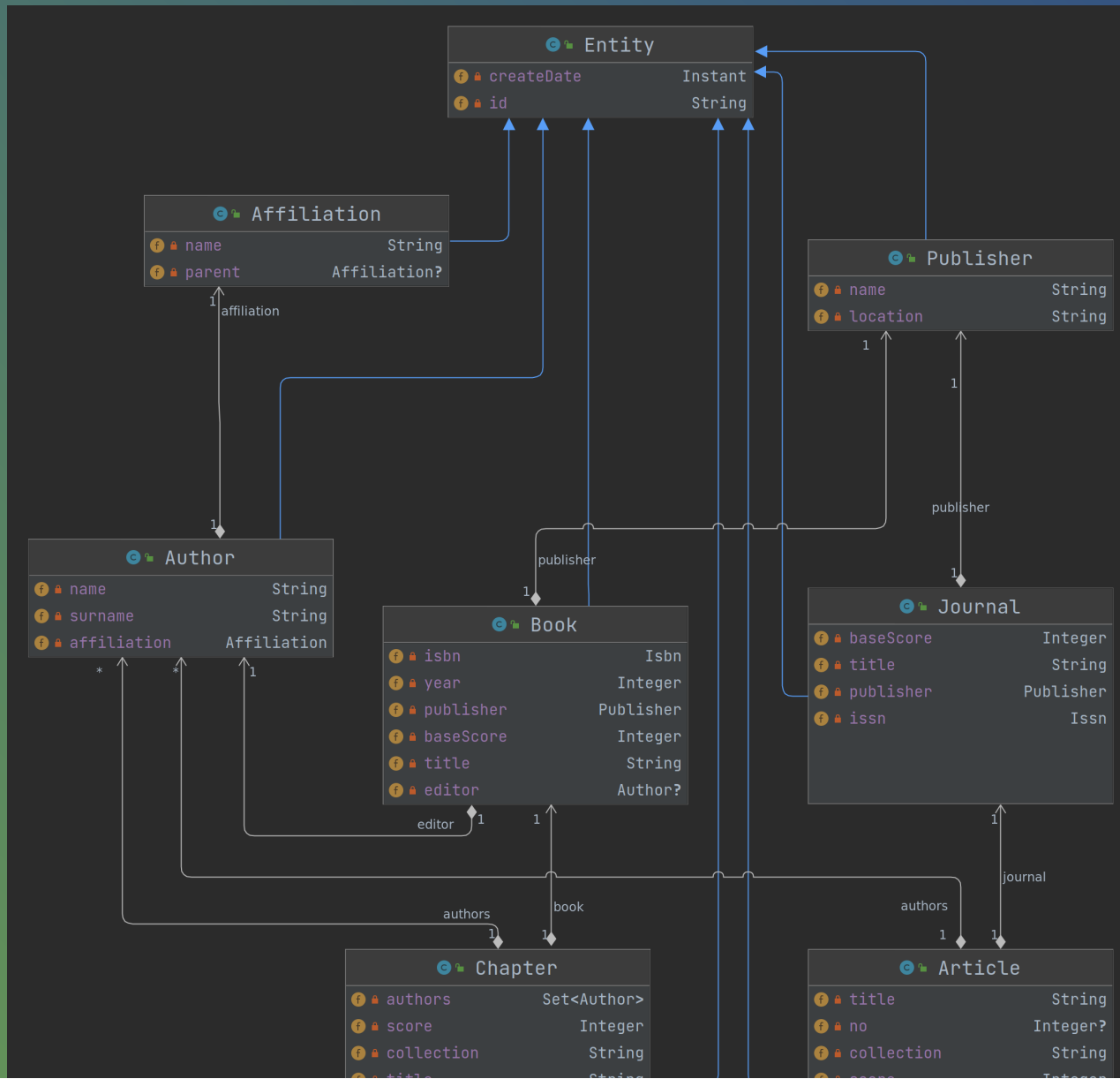
ZADANIE 3: SCRIPT ENGINE

A KOMU TO POTRZEBNE? A DLACZEGO?

Script Engine to interpretator języka programowania komputerowego, którego funkcją jest interpretacja tekstu programów pochodzącego od użytkowników, tłumaczenie takiego tekstu na kod maszynowy wykonywalny przez komputery oraz realizacja szeregu funkcji.

W praktyce jest on wykorzystywany jako interfejs do wpływania w zaawansowany (nieprzewidziany w trakcie implementacji) sposób w stan systemu informatycznego. Przykładowo jest to korekcja danych znajdujących się w systemie lub wywoływanie zadań znajdujących się w harmonogramie (ang. scheduling) realizujących np. integracje z systemami zewnętrznymi.

POKAZ Z OBJAŚNIENIEM



🔒 🔑 title	String
🔒 🔑 book	Book

🔒 🔑 score	Integer
🔒 🔑 articleNo	Integer?
🔒 🔑 authors	Set<Author>
🔒 🔑 vol	Integer
🔒 🔑 journal	Journal

REPREZENTACJA ENCJI W BAZIE NOSQL

```
1 //Journal
2 {
3   "id": "1",
4   "baseScore": 100.0,
5   "title": "Biuletyn WAT",
6   "publisherId": "2",
7   "issn": "1234-5865"
8 }
9 // Publisher
10 {
11   "id": "2",
12   "name": "Wydawnictwo WAT",
13   "location": "Warszawa",
14 }
15
```

REPREZENTACJA ENCJI W BAZIE NOSQL

```
2  {
3    "id": "1",
4    "baseScore": 100.0,
5    "title": "Biuletyn WAT",
6    "publisherId": "2",
7    "issn": "1234-5865"
8  }
9  // Publisher
10 {
11   "id": "2",
12   "name": "Wydawnictwo WAT",
13   "location": "Warszawa",
14 }
15
16
```

REPREZENTACJA ENCJI W BAZIE NOSQL

```
15  
16  
17 //Journal with Publisher  
18 {  
19   "id": "1",  
20   "baseScore": 100.0,  
21   "title": "Biuletyn WAT",  
22   "publisher": {  
23     "id": "2",  
24     "name": "Wydawnictwo WAT",  
25     "location": "Warszawa",  
26   },  
27   "issn": "1234-5865"  
28 }
```

REPREZENTACJA ENCJI W BAZIE NOSQL

```
15  
16  
17 //Journal with Publisher  
18 {  
19   "id": "1",  
20   "baseScore": 100.0,  
21   "title": "Biuletyn WAT",  
22   "publisher": {  
23     "id": "2",  
24     "name": "Wydawnictwo WAT",  
25     "location": "Warszawa",  
26   },  
27   "issn": "1234-5865"  
28 }
```


REPREZENTACJA ENCJI W BAZIE NOSQL

```
8 }
9 // Publisher
10 {
11   "id": "2",
12   "name": "Wydawnictwo WAT",
13   "location": "Warszawa",
14 }
15
16
17 //Journal with Publisher
18 {
19   "id": "1",
20   "baseScore": 100.0,
21   "title": "Biuletyn WAT",
22   "publisher": {
```



LIVE CODING

ZADANIE

Opracować aplikację, która:

- posiada przynajmniej 5 encji;
- posiada api restowe realizujące przynajmniej po jednej operacji: GET, POST, PUT, DELETE;
- posiada system zarządzania bazą danych, który ma być umieszczony w kontenerze i odpowiednio uruchomiony;
- jest napisane w jednym z języków: Java, Kotlin, Scala, Python lub C#;
- ma dostarczać dodatkowe endpointy restowe do wykonywania skryptów wpływających na stan aplikacji;
- ma w README.md zamieszczony przykładowy skrypt do realizacji za pomocą endpointu do wykonywania skryptów. Logika skryptu ma wykraczać poza podstawowe operacje zapisu i odczytu z bazy danych;
- ma posiadać mechanizm wykonujący zdefiniowane zadania zgodnie z harmonogramem zadań (ang. job scheduler) np. [CroneScheduler](#)*;
- ma dostarczać odpowiednie endpointy restowe do modyfikowania, edytowania i podglądu zadań z harmonogramu;
- ma w README.md zamieszczony przykładowy skrypt scheduler-owy, który w trakcie wykonywania skorzysta z zewnętrznego API. Dopuszczalne jest dodanie dodatkowych serwisów w aplikacji realizujących komunikację z zewnętrznym API. W skrypcie muszą znaleźć się instrukcje wpływające na wywołanie metody z zewnętrznego serwisu;*
- ma w README.md zapisane wszelkie dodatkowe operacje, wraz z poleceniami uruchomienia aplikacji.

* zadania wymagane przy ocenie na 5

ZADANIE

Opracować aplikację, która:

- posiada przynajmniej 5 encji;
- posiada api restowe realizujące przynajmniej po jednej operacji: GET, POST, PUT, DELETE;
- posiada system zarządzania bazą danych, który ma być umieszczony w kontenerze i odpowiednio uruchomiony;
- jest napisane w jednym z języków: Java, Kotlin, Scala, Python lub C#;
- ma dostarczać dodatkowe endpointy restowe do wykonywania skryptów wpływających na stan aplikacji;
- ma w README.md zamieszczony przykładowy skrypt do realizacji za pomocą endpointu do wykonywania skryptów. Logika skryptu ma wykraczać poza podstawowe operacje zapisu i odczytu z bazy danych;
- ma posiadać mechanizm wykonujący zdefiniowane zadania zgodnie z harmonogramem zadań (ang. job scheduler) np. [CroneScheduler](#)*;
- ma dostarczać odpowiednie endpointy restowe do modyfikowania, edytowania i podglądu zadań z harmonogramu;
- ma w README.md zamieszczony przykładowy skrypt scheduler-owy, który w trakcie wykonywania skorzysta z zewnętrznego API. Dopuszczalne jest dodanie dodatkowych serwisów w aplikacji realizujących komunikację z zewnętrznym API. W skrypcie muszą znaleźć się instrukcje wpływające na wywołanie metody z zewnętrznego serwisu;*
- ma w README.md zapisane wszelkie dodatkowe operacje, wraz z poleceniami uruchomienia aplikacji.

* zadania wymagane przy ocenie na 5

ZASADY OCENIANIA

Ocenić będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

Do końca trwania laboratoriów należy za pomocą [formularza](#) przesłać link do **publicznego** repozytorium github.com, w którym będzie opracowywane zadanie. **Termin oddania pracy: 15.11.2022r. godz: 24:00**

ZASADY OCENIANIA

Ocenie podlegać będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

Do końca trwania laboratoriów należy za pomocą [formularza](#) przesłać link do **publicznego** repozytorium github.com, w którym będzie opracowywane zadanie. **Termin oddania pracy: 15.11.2022r. godz: 24:00**

ZADANIE 4: CODE GENERATION AND MANIPULATION

WPROWADZAM W SYTUACJE TAKTYCZNĄ

Firma „A” utrzymuje duży system informatyczny do zarządzania danymi badawczymi, informacjami naukowymi i pracownikami uczelni. Posiada ona dużą liczbę klientów, w tym WAT, który oczkuje, że w encji opisującej autora znajdzie się pole określające jego stopień wojskowy. Zmiany takiej nie można wprowadzić dla całego systemu, ponieważ większość uczelni nie oczkuje takiego pola, więc zmiana musi być wyłącznie na systemie uruchomianym dla WAT.

MOŻLIWE ROZWIĄZANIA

MOŻLIWE ROZWIĄZANIA

- Osobna gałąź

MOŻLIWE ROZWIĄZANIA

- Osobna gałąź
- Feature toggle

MOŻLIWE ROZWIĄZANIA

- Osobna gałąź
- Feature toggle
- Refleksja - manipulacja kodem w czasie wykonywania programu



LIVE CODING

ZADANIE

Do aplikacji z poprzedniego zadania dodać możliwość rozszerzania encji o dodatkowe pola poprzez dodatkową konfigurację np. z pliku JSON, gradle.properties lub parametrów systemowych. Funkcjonalność ma dodatkowo rozszerzyć klasy DTO response i request, tak aby endpointy restowe obsługiwały nowe pola. W README.md należy opisać sposób, w jaki należy dostarczyć do systemu konfigurację.

ZASADY OCENIANIA

Ocenić będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

Prace należy kontynuować na repozytorium z poprzedniego zadania.

Termin oddania pracy: ???r. godz: 24:00

ZASADY OCENIANIA

Ocenić będą:

1. Poprawność wykonania;
2. Samodzielność;
3. Zastosowanie dobrych praktyk programistycznych;
4. Wykorzystanie systemu kontroli wersji.

Prace należy kontynuować na repozytorium z poprzedniego zadania.

Termin oddania pracy: ???r. godz: 24:00