Задача №1. Coding

Дано: описание грамматики в виде текстового файла

Найти: закодированное представление грамматики в виде последовательности целых чисел

Метод решения:

Ищем в описании грамматики терминалы, нетерминалы и семантики (например, с помощью регулярных выражений)

Присваиваем каждому найденному элементу свой номер (код)

Затем читаем описание грамматики посимвольно, и если данная последовательность символов – терминал, нетерминал или семантика (из найденных), то выводим его код

Фрагмент программы на языке Kotlin (процесс кодирования):

for (command in grammar) {  
 if ("Eofgram" in command)  
 break  
  
 // кодируем семантики  
 val semantics = command  
 .*split*(" ")  
 .*filter* **{ it**.*matches*(Regex("\\$[a-z]+[0-9]\*")) **}** for (semantic in semantics) {  
 if (semanticNumber >= 1000) {  
 break  
 }  
 if (semantic !in codes.keys) {  
 codes[semantic] = semanticNumber++  
 }  
 }  
  
 // кодируем нетерминалы, обозначенные заглавными буквами  
  
 val nonterminals = command  
 .*split*(" ")  
 .*filter* **{ it**.*matches*(Regex("[A-Z]+")) **}** for (nonterminal in nonterminals) {  
 if (nonterminalNumber > 50) {  
 break  
 }  
 if (nonterminal !in codes.keys) {  
 codes[nonterminal] = nonterminalNumber++  
 }  
 }  
  
 // кодируем терминалы  
 val rightPartCommand = command.*split*(" : ")[1]  
 val terminals = rightPartCommand  
 .*split*(" ")  
 .*filter* **{  
 it**.*matches*(Regex("'.+'")) ||  
 **it**.*matches*(Regex("[a-z]+"))  
 **}** for (terminal in terminals) {  
 if (terminalNumber > 100) {  
 break  
 }  
 if (terminal !in codes.keys) {  
 codes[terminal] = terminalNumber++  
 }  
 }  
}