

DOMAIL SPECIFIC LANGUAGE FOR COMPUTING TAXES

Diana BELAIA^{1*}, Corneliu CATLABUGA¹, Ana BZOVII¹,
Igor MIHAILUIC¹, Constantin BAXANEAN¹

¹Department of Software Engineering and Automation, group FAF-213, Faculty of Computers, Informatics and Microelectronics, Technical University of Moldova, Chisinau, Republic of Moldova

*Author correspondent: Diana Belaia, diana.belaia@isa.utm.md

Scientific coordinator: Irina Cojuhari, conf. univ. dr., Gabriel Zaharia, univ. assist.

Abstract: The aim of the following article is to describe the notion of taxes, the rules of computations in different countries and one of possible ways to develop a Domain Specific Language (DSL) to make computing taxes easy for an average user. It contains some general information of taxes to make readers familiar with this field. Further, it explains how DSL works, the identified problem which is solved by DSL and it is presented the grammar.

Keywords: taxes, Domain Specific Language, grammar, problem.

Introduction

Computing taxes is the process of calculating how much tax an individual or organization owes to the government based on their income, assets, or other factors. The specific method of computing taxes varies depending on the country and tax system in question, but in general, it involves the following steps:

1. **Determine taxable income:** Taxable income is the amount of income that is subject to taxation. It is usually calculated by subtracting certain deductions and exemptions from gross income.
2. **Determine tax rate:** The tax rate is the percentage of taxable income that must be paid as tax. The tax rate can vary depending on the type of income, the taxpayer's income level, and other factors.
3. **Calculate tax liability:** To calculate tax liability, multiply taxable income by the applicable tax rate. Some tax systems also have additional taxes or fees that must be added to the tax liability.
4. **Apply credits and deductions:** Tax credits and deductions can reduce the amount of tax liability owed. Tax credits are typically dollar-for-dollar reductions in the amount of tax owed, while deductions reduce taxable income.
5. **File tax returns:** Taxpayers are required to file tax returns with the government to report their income, deductions, and tax liability. Tax returns are usually due on an annual basis, although some systems may require more frequent reporting.

It is important to note that computing taxes can be a complex process, and mistakes or omissions can result in penalties or legal problems. Many individuals and organizations choose to use tax software or hire professional tax preparers to help ensure accuracy and compliance with tax laws.

Taxes are mandatory contributions levied on individuals or corporations by a government entity (local, regional or national). From an accounting perspective, there are various types of taxes [1]:

1. **Income taxes:** A percentage of generated income that is relinquished to the state or federal government
2. **Corporate taxes:** A percentage of corporate profits taken as tax by the government to fund federal programs
3. **Estate taxes:** rate is applied to the fair market value of property in a person's estate at the time of death; the total estate must exceed thresholds set by state and federal governments

4. **Tariff:** Taxes on imported goods; imposed with the aim of strengthening domestic businesses
5. **Property taxes:** Based on the value of land and property assets
6. **Sales taxes:** Taxes levied on certain goods and services; varies by jurisdiction [2].

Taxes in different countries:

Computing taxes can vary from country to country since each nation has its own tax system and rules. Here is some general information about computing of taxes in different countries:

1. **United States:** income taxes are calculated based on a progressive tax system, meaning the tax rate is increasing as income is increasing. The Internal Revenue Service collects taxes on income, capital gains and dividends.
2. **United Kingdom:** taxes are collected by HM Revenue and Customs. The tax rate is based on a progressive tax system and the rates are different for different types of income.
3. **Canada:** the federal government and the provincial government both collect income taxes. The federal tax rate is based on the progressive rate system and each province has its own tax rates.
4. **Moldova:** the personal income tax rate in Moldova is progressive with rates ranging from 12% and 18% depending on the amount income earned.

The problem of computing taxes and DSL benefits

Nowadays computing of taxes can become a real problem for those who want to start their own business or for people who are not familiar with economics and law. Since there are a lot of tax rate systems and these systems vary from country to country it would be a saving solution to develop a Domain Specific Language with user friendly interface which would allow us to compute the amount of the tax for our income, to check the correctness of computations offered by special instances. Moreover, one of the advantages for those individuals who develop their own small business will be a possibility not to have an accountant or a special person who is usually responsible for this part of finance in the company.

Language overview:

The DSL for computing taxes is designed to provide a streamlined and user-friendly solution for tax preparation. The language uses several basic data structures to manipulate and store data, including strings for names, rates for tax rates between 0 and 1, percentages for tax rates between 0 and 100 with % at the end, and positive floats for income. Users can create and manipulate data by specifying the appropriate data structure and assigning values to them.

The DSL also includes basic control structures for specifying tax brackets and computing tax deductions. Users can declare tax brackets with multiple income ranges, each with an assigned tax rate or percentage. After the brackets are declared, users can specify the target income and any applicable deductions to compute the required tax deduction. Control flow is manipulated by specifying the appropriate brackets and income ranges.

Programs in the DSL require input such as the income and tax bracket ranges, as well as any applicable deductions. The output of the program is the required tax deduction based on the income, bracket, and deductions specified by the user.

The language also includes error handling mechanisms to prevent common errors. Programs can go wrong if the range declarations do not start from 0 or end at the maximum value, or if the ranges overlap. The language communicates errors to the user through error messages and warnings, helping users to identify and fix errors in their code.

Overall, the DSL for computing taxes is designed to simplify tax preparation and provide a more efficient and reliable solution for individuals and businesses. By providing a user-friendly interface and comprehensive error handling, the language aims to make tax preparation easier and more accessible for everyone.

Syntax

Syntax of programming language is considered to be notations and rules to control the structure. Nowadays all scripts and programs which are supposed to compute the taxes are written in compiled programming languages since it is faster. Our goal is to develop a programming language the syntax pretty similar to Python syntax since it is easy to understand and use even for people who do not have deep knowledge of mathematics and do not want to spend time on learning the usage of DSL, its specific structure.

Grammar of DSL

To better understand the grammar of a particular programming language it is needed to refer to some general notions. The notations necessary for grammar:

V_n – the finite set of non-terminal symbols.

V_t – the finite set of terminal symbols which, also, defines the alphabet of the language

S – the starting symbol

P – the finite set of productions (allowed combinations to generate the strings)

DSL grammar:

$V_n = \{$

<program>, <newtaxbracket>, <newrange>, <text>, <character>, <real>, <integer>, <rate>, <percentage>, <letter>, <symbol>, <digit>, <newline>, <newtaxcompute>, <deductions>, <charity>, <standart>

$\}$

$V_t = \{$

tax_bracket, range, tax_compute, amount, deductions, charity_deduction, standart_deduction, a..z, A..Z, 0..9, max, ., -, <, >, =, %, ", \, /, @, #, !, ?, +, *, \$, NEWLINE, IDENTATION

$\}$

$S = \{ \langle \text{program} \rangle \}$

$P = \{$

<program > -> <newtaxbracket> NEWLINE <newline> <newtaxcompute>
 <newtaxbracket> -> tax_bracket "<text>" NEWLINE <newline> <newrange> | tax_bracket "<text>" NEWLINE <newline> <newrange> <newbracket>
 <newline> -> NEWLINE | NEWLINE <newline> | ϵ
 <text> -> <character> | <character><text> | ϵ
 <character> -> <letter> | <symbol> | <digit>
 <real> -> <integer>.<integer> | <integer> | max
 <integer> -> <digit><integer> | <digit>
 <rate> -> 0.<integer> | 1 | 0
 <percentage> -> <digit><digit>.<integer> % | 100 % | 0 %
 <letter> -> a | A | b | B | ... | z | Z
 <symbol> -> . | ! | ? | , | : | ; | - | < | > | @ | # | % | \ | / | + | * | \$
 <digit> -> 0 | 1 | 2 | ... | 9
 <newrange> -> IDENTATION range <real>..<real> -> <percentage> NEWLINE
 <newline> | IDENTATION range <real>..<real> -> <rate> NEWLINE <newline> |
 IDENTATION range <real>..<real> -> <percentage> NEWLINE <newline> <newrange>
 | IDENTATION range <real>..<real> -> <rate> NEWLINE <newline> <newrange>
 <newtaxcompute> -> tax_compute "<text>" NEWLINE <newline> IDENTATION
 bracket = "<text>" NEWLINE <newline> IDENTATION income = <real> <newline> |
 tax_compute "<text>" NEWLINE <newline> IDENTATION bracket = "<text>"
 NEWLINE <newline> IDENTATION income = <real> NEWLINE <newline> | ϵ

```
<deductions> -> INDENTATION deductions NEWLINE <newline> <charity> <standart> |  
INDENTATION deductions NEWLINE <newline> <charity> | INDENTATION deductions  
NEWLINE <newline> <standart> | ε  
<charity> -> INDENTATION INDENTATION charity_deduction = <float> NEWLINE  
<newline>  
<standart> -> INDENTATION INDENTATION standart_deduction = <float>  
NEWLINE <newline>  
}
```

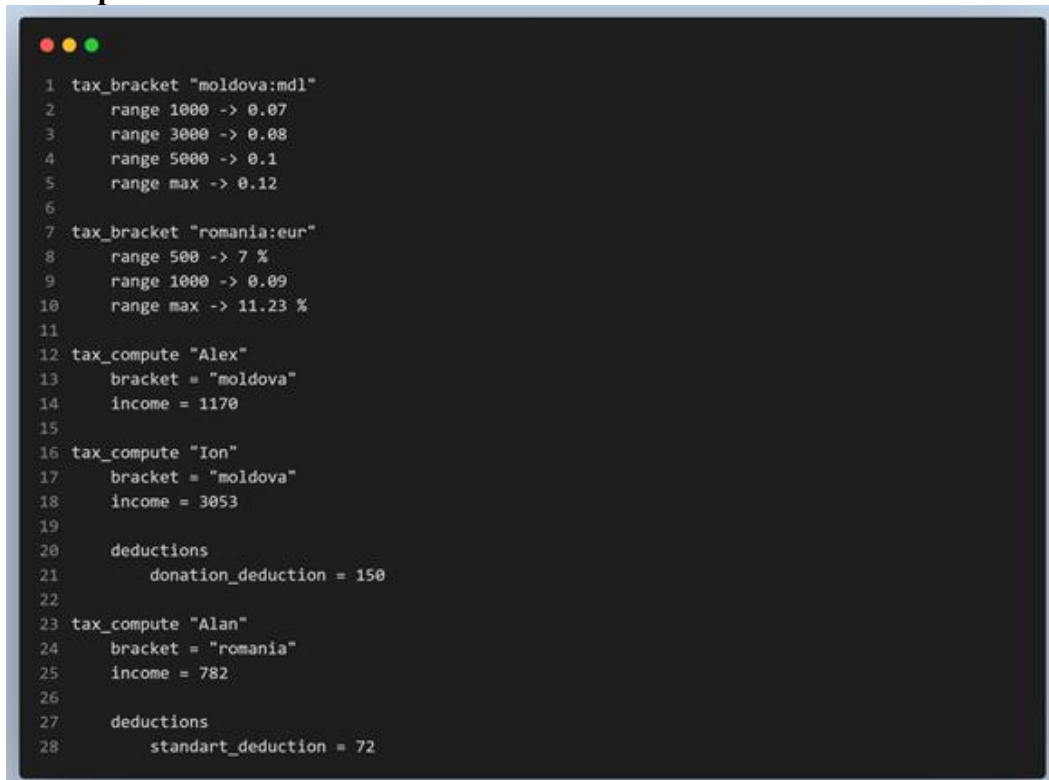
Syntax description:

The program file should start by declaring all the tax brackets that might be used later in the program. Each tax bracket should have a unique name incased in double quotation marks ["..."], used for identification. After name declaration the user should define one or multiple ranges, where the max-income has its specific tax rate, in increasing order, the last range having a max income and to each range being assigned a rate [x.y...] or percent [x.y... %] value.

Following the tax bracket(s) declaration(s) the user should declare the tax compute(s) target(s), each having assigned a name, brackets in which that tax compute should be classified and income. Optionally the user can declare the deductions where standard and/or donation deductions are specified.

The program should output the tax compute target along its required tax.

Code Example



```
1 tax_bracket "moldova:mdl"  
2   range 1000 -> 0.07  
3   range 3000 -> 0.08  
4   range 5000 -> 0.1  
5   range max -> 0.12  
6  
7 tax_bracket "romania:eur"  
8   range 500 -> 7 %  
9   range 1000 -> 0.09  
10  range max -> 11.23 %  
11  
12 tax_compute "Alex"  
13   bracket = "moldova"  
14   income = 1170  
15  
16 tax_compute "Ion"  
17   bracket = "moldova"  
18   income = 3053  
19  
20   deductions  
21     donation_deduction = 150  
22  
23 tax_compute "Alan"  
24   bracket = "romania"  
25   income = 782  
26  
27   deductions  
28     standart_deduction = 72
```

Figure 1. Code example

Parse tree:

Parsing tree represents the detailed way to generate the code using allowed productions. On Figure 1 the part of parsing tree for DSL for computing taxes is presented:



Figure 2. Parsing tree example

Conclusions

Domain Specific Language for computing of the taxes is a good solution to unite the income taxes from most part of the countries and to make tax computation easier for business. Despite on the narrow field of application, DSL for taxes can help to improve people's knowledge about the taxes, economics and law in general. Moreover, DSL can reduce the number of specialists who are needed to be hired to solve the computational problems and this is the greate advantage for bussiness since we have to pay less.

References:

1. Basis of tax computation. [online]. [accessed 24.02.2023]. Available: <https://www.accountingnotes.net/international-taxation/domestic-tax/basis-of-tax-computation/7469>
2. Preparing a Tax Computation. [online]. [accessed 25.02.2023]. Available: [https://www.iras.gov.sg/taxes/corporate-income-tax/form-c-s-form-c-s-\(lite\)-form-c-filing/preparing-a-tax-computation](https://www.iras.gov.sg/taxes/corporate-income-tax/form-c-s-form-c-s-(lite)-form-c-filing/preparing-a-tax-computation)
3. Domain Specific Language. [online]. [accessed 10.03.2023]. Available: <https://stackoverflow.com/questions/809574/what-is-a-domain-specific-language-anybody-using-it-and-in-what-way>