**Перечень задач**

*PyJsik*

История изменений документа

|  |  |  |
| --- | --- | --- |
| Дата | Автор | Внесённые изменения |
| 12.01.2022 | М.Ю.Чуйко | Исходная версия требований |
| 30.01.2022 | М.Ю.Чуйко | Дополнены требования, обнаруженные при тестировании |
|  |  |  |
|  |  |  |

Нумерация происходит следующим образом. Каждый архитектурный модуль имеет своё кодовое название

IO

LA

SYNA

CG

У каждой задачи в рамках модуля будет своё наименование. Проект сравнительно небольшой, потому каждая задача будет относиться к разработке целого класса

IO-001

Разработать систему ввода-вывода. Система должна соответствовать всем требованиям REQ\_IO.

LA-001

Разработка класса data класса Token. Класс представляет из себя модель токенизированной лексемы. Он хранит тип лексемы и её позицию в исходном коде. Работа данного класса должна соответствовать всем требованиям REQ\_LA

LA-002

Разработать data класс Tokens, который представляет из себя хранилище типов токенов, регулярных выражений каждого типа и итоговый список токенов исходного текста. Работа данного класса должна соответствовать всем требованиям REQ\_LA

LA-003

Разработать класс Tokenizer, который реализует алгоритм токенизации, выводит в консоль ошибку, если встречена лексема неизвестного типа и возвращает список токенов, в случае успешной обработки исходного кода. Работа данного класса должна соответствовать всем требованиям REQ\_LA.

LA-004

Разработать класс LexicalAnalizer, который принимает на вход текст исходного кода и возвращает массив токенов. Данный класс должен выводить в консоль результат своей работы и предлагать сохранять его в текстовом файле. Работа данного класса должна соответствовать всем требованиям REQ\_LA.

SYNA-001

Разработать класс-представление SyntaxTree. Класс реализует дерево синтаксического разбора в соответствии с правилами языка Python. Работа данного класса должна соответствовать всем требованиям REQ\_SINTA.

SYNA-002

Реализовать класс SyntaxAnalizer, реализующий обработку токенов из списка токенов и выводящий в консоль ошибку, если была обнаружена синтаксическая ошибка. Работа данного класса должна соответствовать всем требованиям REQ\_SINTA.

SYNA-003

Разработать класс SyntaxParser, реализующий приём списка токенов и возвращающий объект класса SyntaxTree. Данный класс должен выводить в консоль результат своей работы и предлагать сохранять его в текстовом файле. Работа данного класса должна соответствовать всем требованиям REQ\_SINTA.

CG-001

Разработать класс-грамматику Gramma, содержащий правила грамматики для языка JavaScript. Правила должны быть реализованы в виде функций, принимающих на вход узел дерева синтаксического разбора и возвращающие текст на языке JavaScript. Работа данного класса должна соответствовать всем требованиям REQ\_CG.

CG-002

Разработать класс CodeTranlator, который принимает на вход узел дерева синтаксического разбора, передаёт грамматике и формирует конечный текст. Работа данного класса должна соответствовать всем требованиям REQ\_CG.

CG-003

Разработать класс Codegen, который принимает на вход дерево синтаксического разбора и возвращает конечный текст программы на языке JavaScript. Данный класс должен выводить в консоль результат своей работы и сохранять его в текстовом файле. Работа данного класса должна соответствовать всем требованиям REQ\_CG.

ERR-001

Разработать и реализовать класс обработки ошибок, возникающий при работе всех вышеперечисленных классов. Дополнять функционал класса по мере разработки и тестирования продукта.

Данный перечень задач реализован в среде командной разработке Trello

Дополнение задач:

CG-001

ISSUE ID 01

Грамматику необходимо дополнить правилами описания условных операторов

SYNA-002

ISSUE ID 02

Настроить модуль на работу с обновлённой грамматикой.

CG-002

ISSUE ID 03

Добавить функционал для работы с условными операторами. Реализовать обработку узла синтаксического дерева, обозначающего условный оператор.

CG-003

ISSUE ID 04

Добавить функционал для работы с условным оператором.