Нумерация:

Номер задачи пишется в квадратных скобках и состоит из следующих частей:

1. сокращение IS;
2. название подсистемы. Возможные значения: FRONT – задачи, относящиеся к интерфейсу программы, IO – задачи, относящиеся к подсистеме ввода и вывода, TRANSLATE – задачи, относящиеся к подсистеме трансляции;
3. порядковый номер. Для каждой подсистемы используется отдельный порядковый числовой номер для требования.

Между частями номера требования пишется символ «\_». После номера задачи пишутся заголовок задачи, развёрнутое описание и ссылки на требования, по которым составлена задача.

Задачи по разработке графического интерфейса пользователя:

Разработать пользовательский интерфейс

Разработать пользовательский интерфейс, готовый к подключению к нему компонентов транслятора. В интерфейсе должны присутствовать все предполагаемые элементы управления. Также требуется продумать цветовую схему.

Ссылки: REQ\_FRONT\_1 – 16.

Разработать класс элемента управления «Кнопка»

Написать класс пользовательского элемента управления, выполняющего функцию кнопки.

Ссылки: REQ\_FRONT\_2.

Разработать класс элемента управления «Поле ввода»

Написать класс пользовательского элемента управления, выполняющего функцию поля для ввода текста. Поле должно иметь возможность блокировать изменение текста и показывать плейсхолдер, должно обладать рамкой, меняющей цвет при получении/потере фокуса ввода.

Ссылки: REQ\_FRONT\_2, REQ\_FRONT\_3, REQ\_FRONT\_5.

Диалоги для чтения из файла и сохранения в файл

Добавить два диалога для работы с файловой системой: один для чтения кода на языке C# из файлов формата TXT и CS, второй для сохранения кода на языке Java в файлы формата TXT и JAVA.

Ссылки: REQ\_FRONT\_6, REQ\_FRONT\_7, REQ\_IO\_1, REQ\_IO\_3.

Подтверждение выхода из программы

Добавить окно с подтверждением выхода, появляющееся при наличии переведённого кода.

Ссылки: REQ\_FRONT\_16.

Задачи по разработке подсистемы ввода/вывода:

Проверка входного файла на пустоту

Добавить проверку файла, выбранного с помощью диалога, на пустоту. Если файл пустой, выводить соответствующее предупреждение в виде окна с сообщением.

Ссылки: REQ\_IO\_2.

Задачи по разработке подсистемы трансляции:

Разработать лексический анализатор

Разработать лексический анализатор, выполняющий разбиение входного кода на языке C# на лексемы. Выходными данными анализатора должен быть двухмерный массив структур вида

* текст лексемы;
* тип лексемы;
* номер строки в тексте;
* номер столбца в тексте.

Ссылки: REQ\_TRANSLATE\_1.

1. //TODO