Программный продукт состоит из следующих подсистем:

1. Пользовательский интерфейс.
2. Модуль лексического анализа.
3. Модуль синтаксического анализа.
4. Модуль семантического анализа.
5. Модуль генерации кода.

### Функциональные требования

1. Продукт должен транслировать код с языка программирования Pascal на язык программирования Python.
2. Продукт должен принимать файл с кодом на языке программирования Pascal.
3. Продукт должен формировать файл с кодом на языке программирования Python.
4. Продукт должен отображать входной код на языке программирования Pascal.
5. Продукт должен отображать итоговый код на языке программирования Python.

### Нефункциональные требования

1. Продукт должен быть написан на языке Python.
2. Продукт должен иметь пользовательский интерфейс.
3. Контроль версий продукта должен с помощью GitHub.
4. Для каждой подсистемы должны быть подготовлены unit-тесты.
5. Взаимодействие между системами должно быть протестировано интеграционными тестами.

## Требования к подсистеме «Пользовательский интерфейс»

1. В программном продукте должны быть следующие элементы интерфейса:

* форма добавления файла с исходным кодом на Pascal;
* форма подтверждения трансляции кода;
* текстовое поле с входным кодом на языке программирования Pascal;
* текстовое поле с выходным кодом на языке программирования Python;
* форма сохранения файла с кодом на Python.

1. Программный продукт должен позволять пользователю вводить входной код на языке программирования Pascal через соответствующую форму.
2. Программный продукт должен принимать файлы с расширением «.pas» при добавлении файла по нажатии кнопки.
3. Программный продукт должен отображать код на языке программирования Pascal из входного файла через текстовое поле.
4. Если во входном файле пользователя есть некорректные данные, то по нажатии кнопки подтверждения трансляции кода пользователь должен получать сообщение с описанием ошибки.
5. Если во входном коде пользователя, введённым через текстовое поле, есть некорректные данные, то по нажатии кнопки подтверждения трансляции кода пользователь должен получать сообщение с описанием ошибки.
6. Сообщение об ошибке (REQ\_UI\_005 и REQ\_UI\_006) должно содержать в себе информацию о типе ошибки (лексическая, синтаксическая, неверный ввод) и номер строки, где она возникла.
7. Программный продукт должен позволять пользователю сохранять выходной файл с расширением .py при сохранении файла по нажатии кнопки.
8. Программный продукт должен выводить пользователю выходной код на языке программирования Python через текстовое поле.

## Требования к подсистеме «Модуль лексического анализа»

1. Модуль должен получать на вход набор символов входного кода на языке программирования Pascal, разделять его на токены (см. REQ\_LE\_002).
2. Каждый токен должен содержать код лексемы из выбранного подмножества языка Pascal, значение лексемы, номер строки и столбца, в которых начинается лексема.
3. В случае нахождения лексемы, не попадающей в список выделенных (см. REQ\_LE\_002), модуль возвращает сообщение с описанием ошибки.

## Требования к подсистеме «Модуль синтаксического анализа»

1. На вход модулю подаётся последовательность токенов (см. REQ\_LE\_001), на основе которой строится дерево разбора.
2. В случае нахождения синтаксической или лексической ошибки, модуль возвращает сообщение с описанием ошибки.

## Требования к подсистеме «Модуль семантического анализа»

1. Модуль проверяет на корректность входное дерево разбора (см. REQ\_SY\_001).
2. В случае нахождения семантической ошибки, модуль возвращает сообщение с описанием ошибки.

## Требования к подсистеме «Модуль генерации кода»

1. На вход поступает дерево разбора (см. REQ\_SY\_001), на его основе происходит генерация кода на языке Python (см. REQ\_CG\_002) и выводится в пользовательский интерфейс.