

Введение



Приветствую. Если Вы читаете этот документ – видимо, Вас интересует мир разработки, и Вы желаете дополнить общедоступное приложение проверки системы для приложений AMAZING

Это похвальное желание, которое, однако, должно быть поддержано документацией. Этот файл – подобие документации, призванное помочь начинающим разработчикам разобраться в особенностях проекта и стандартных шаблонов

Термин	Объяснение
Приложение	Исполняемый файл в широком смысле или проект с кодом программы в отношении проекта AmazingSimpleHelper
Репозиторий	Публичный репозиторий с кодом на GitHub, доступный по ссылке https://github.com/Drygok/AmazingSimpleHelper
Тест	Блок кода, призванный проверить соответствие ресурсов компьютера и/или системы на возможность запуска соответствующих приложений или собрать информацию для отладки
Точка входа	Первая функция, вызываемая в приложении [для AmazingSimpleHelper это статический метод Main]
Файл с точкой входа	Файл с кодом программы, содержащий функцию-точку входа [для AmazingSimpleHelper это файл Program]
Автор	Автор документа и изначального приложения (Snegirev Maxim vk.com/drygok)

Таблица 1. Терминология

Приступая к работе



Инструкция по добавлению теста `AmazingSimpleHelper`

Для начала работы необходимо клонировать репозиторий, после чего редактировать его в любом удобном редакторе (для эффективной работы рекомендуется использовать IDE Microsoft Visual Studio)

Порядок действий

Добавление теста происходит следующим образом

1. Создаем класс с тестом в директории `Tests`
2. Добавляем вызов конструктора в массив `DevelopmentTests` в файле с точкой входа
3. Дополняем репозиторий своими правками (Pull requests)

ВАЖНО!

Пожалуйста, учтите, что тесты из `DevelopmentTests` могут содержать непроверенный код, поэтому исполнять их следует только из оригинального репозитория

Рекомендации для тестов

Особенности добавления тестов

- Заголовок файла теста должен отражать его суть и заканчиваться словом `Test`
- Класс должен наследоваться от класса `AbstractTest`
- Класс должен переопределять поле `Name` как заголовок теста (напр., "Инициализация приложения")
- Класс должен переопределять метод `Test()` как код этого теста и возвращать результат теста

Возвращать строку с результатом теста следует только при успешном прохождении теста, при ошибке рекомендуется возвращать исключение с установленным сообщением об ошибке, содержащим краткую информацию о сути проблемы

Пример кода

Примеры реализации тестов можно увидеть в классах `StartHelperTest` и `ErrorTest`, повторяя их поведение в своем коде

При появлении необходимости передавать дополнительные параметры в код своих тестов допустимо создание нестандартного конструктора (пример такого исполнения можно получить в классе `ErrorTest`)