

Содержание

Цель работы.....	2
Задание на практическую работу	2
Выполнение практической работы.....	2
Вывод.....	6
Список информационных источников:	6

Цель работы

Ознакомится с методологией разработки через тестирование (TDD), с ее помощью с нуля спроектировать и разработать программу по выбранной теме.

Задание на практическую работу

1. Реализовать приложение с нуля согласно заданию с использованием методологии разработки через тестирование TDD (Test Driven Development) или разработки, управляемой поведением, BDD (Behavior Driven Development) по выбору студента.

2. Каждый промежуточный шаг разработки (рабочий код и тесты) коммитить в систему контроля версий.

3. Сложный графический интерфейс необязателен. Вся функциональность должна быть реализована в классах, допустимо использование консольного интерфейса, где явно не нужен графический интерфейс.

4. При сдаче преподавателю показывать функционал приложения и код, так, чтобы были видны diff (различия по строкам) между коммитами, начиная с первого, и можно было отследить последовательность разработки (в средстве git или на GitHub).

Выполнение практической работы

Выбранная тема: 1. Игра «Поле чудес» (угадать загаданное слово).

Программа разработана на языке python с использованием открытой библиотеки unittest для Unit-тестирования. Разработка производилась на ОС Windows, тестирование программы также осуществлялось на данной ОС.

Работу программы можно описать следующим образом: существует какое-то загаданное слово, которое пользователь должен отгадать. Для этого пользователь может ввести итоговое слово сразу, либо угадывать по буквам. Таким образом, например, если дано слово dog, то пользователь, например, угадывает первую и последнюю букву и на экран выводится “d_g”. Если пользователь догадался, он может ввести слово dog и выиграть. Также пользователь в любой момент может завершить игру.

В рамках программы был разработан удобный пользовательский интерфейс в текстовом виде в консоли. Разработка программы велась с

использованием системы контроля версий git, использовался локальный git-репозиторий, затем исходные коды уже полностью готовой программы была загружена в удаленный git-репозиторий.

Разработка программы велась по следующему алгоритму: сначала пишется тест для некоторой функции, производящей конкретное действие или действия, затем тест запускался (очевидно неудачно, так как еще не был написан код логики), после этого писался код логики самой функции, далее тест снова запускался и, если была обнаружена ошибка, код функции исправлялся, иначе разработка переходила к разработке следующей функциональности.

Структурно программа состоит из нескольких модулей: базовый класс игры, который обрабатывает основную часть игры и ввод отдельных букв. Также есть класс-наследник, который объявляет слово, которое нужно отгадать. Суть программы состоит в следующем: получить на вход буквы или полное слово, обработать введенные данные и вернуть пользователю результат его попытки. В случае угадывания слова, пользователь побеждает, в случае ввода верной буквы, они открываются и пользователю проще догадаться.

Выполнение работы представлено на рисунках 1-5. Код программы также доступен в репозитории на платформе GitHub.com <https://github.com/Erupb/tivpo3>.

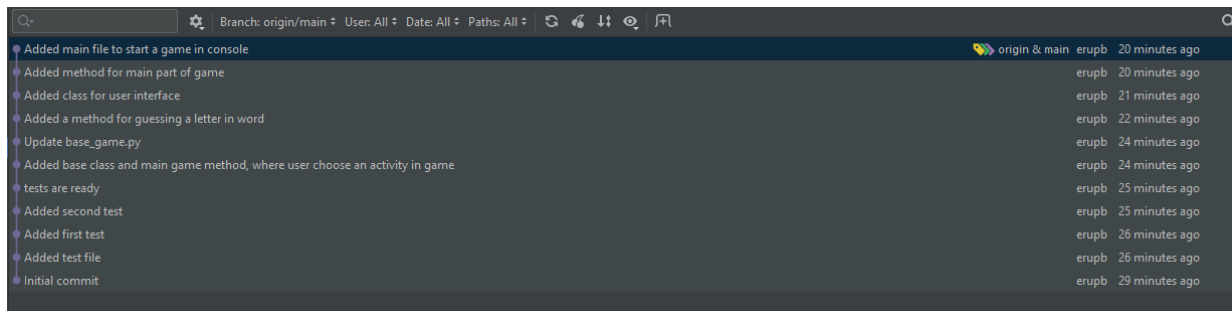


Рисунок 1 – Просмотр коммитов в локальный репозиторий проекта

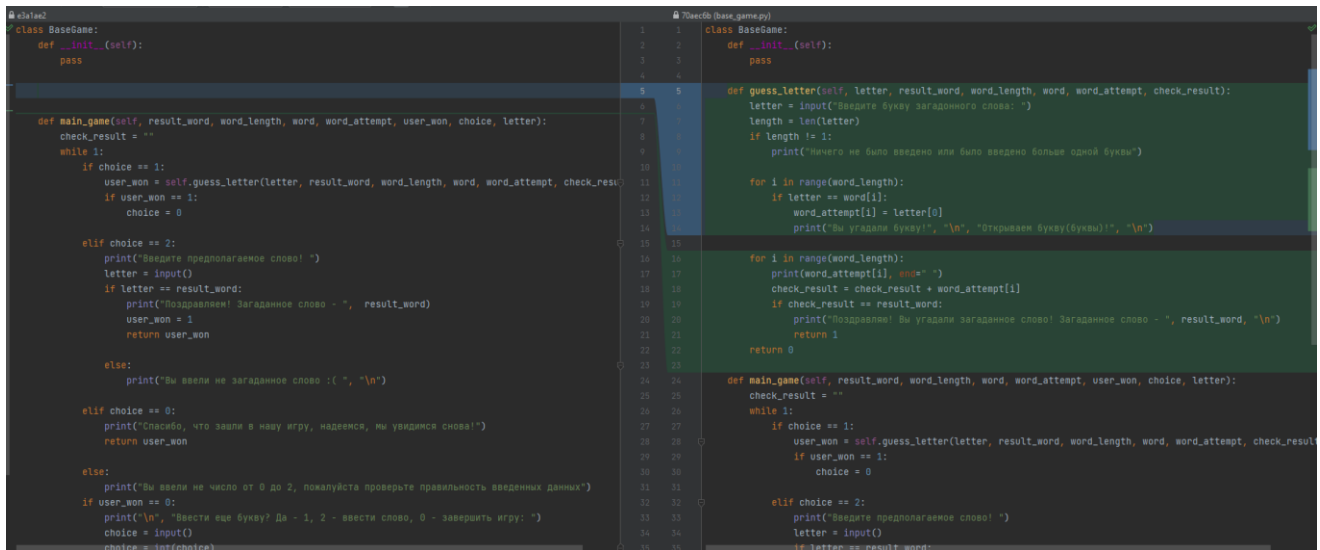


Рисунок 2 – Просмотр изменений в файлах в сравнении 2х коммитов

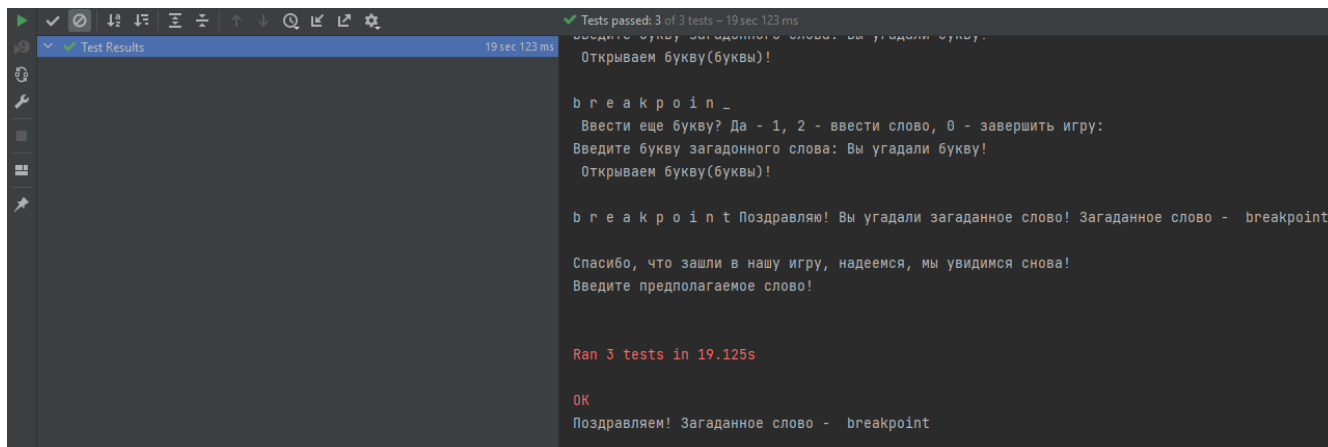


Рисунок 4 – Запуск и успешное прохождение тестов

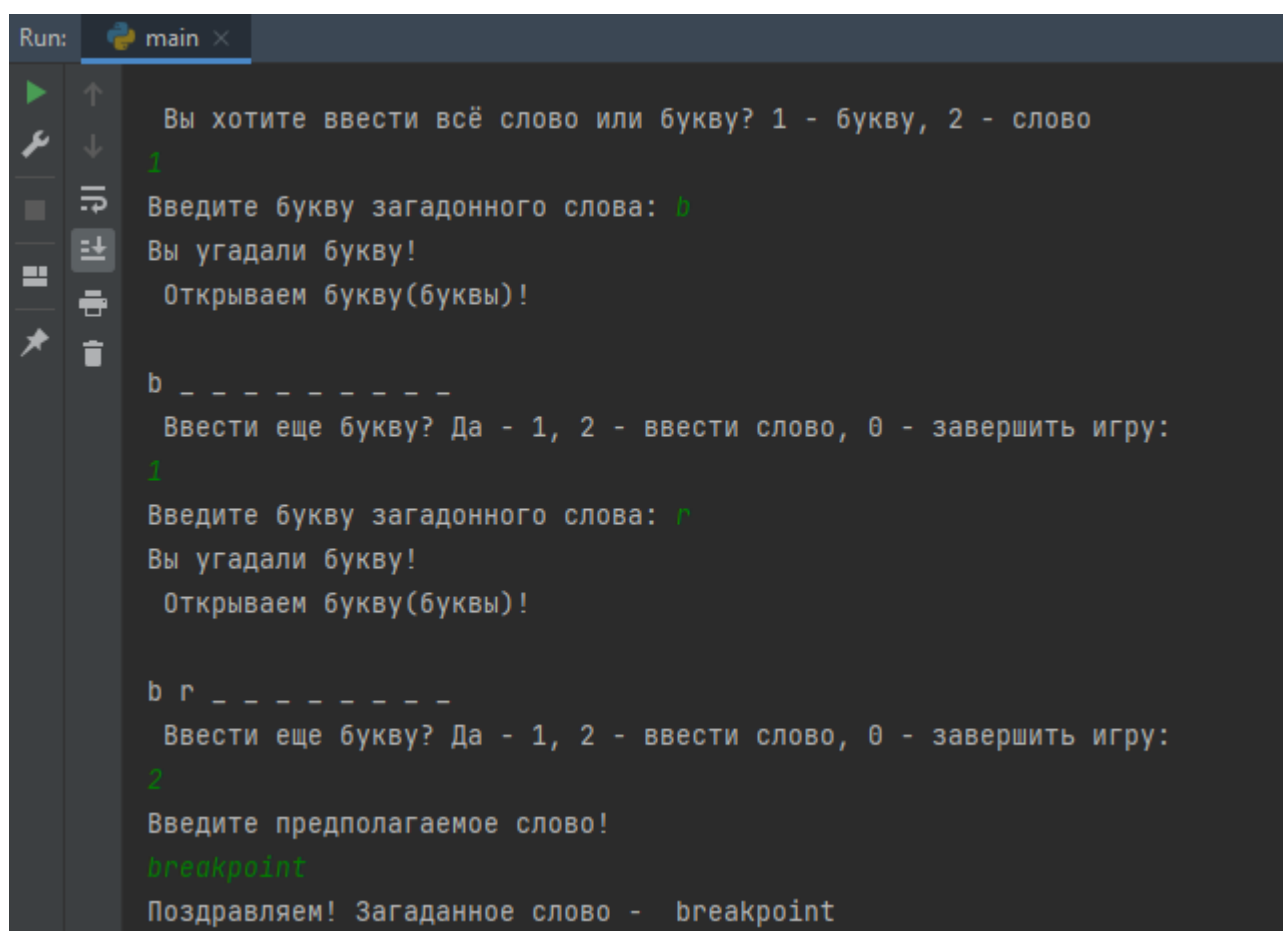


Рисунок 5 – Ручное тестирование работоспособности готовой программы

Вывод

В ходе выполнения данной работы были укреплены навыки проектирования, разработки и тестирования программ. Были укреплены навыки работы с библиотекой Unit-тестирования unittest, а также написания классов на языке Python. В ходе тестирования успешно выполнились все тесты, ручное тестирование выявило полную работоспособность программы в соответствии с поставленной задачей.

Список информационных источников:

1. Документация по библиотеке unittest, электронный ресурс URL: <https://docs.python.org/3/library/unittest.html> (дата последнего обращения: 10.10.2022) – текст электронный.
2. Документация по поиску изменений в git репозитории, электронный ресурс URL: https://www.jetbrains.com/help/pycharm/viewing-changes-information.html#local_changes (дата последнего обращения: 10.10.2022) – текст электронный.