

## Проект 1: Виселица

### Описание

Разработайте консольную версию игры "Виселица", в которой игрок пытается угадать загаданное слово, вводя буквы по одной за раз. Слово выбирается по уровню сложности, случайно из предварительно заданного списка слов и категории. Количество попыток ограничено, и за каждую неверную догадку визуализируется часть виселицы и фигурки висельника.

### Функциональные требования

- Программа должна выбрать случайную категорию слов, если этого не сделал пользователь.
- Программа должна выбрать уровень сложности, если этого не сделал пользователь.
- Программа должна выбирать случайное слово из заранее определенного списка слов.
- Игрок вводит одну букву за раз, чтобы угадать слово.
- Экран должен обновляться после каждого ввода, показывая уже угаданные буквы и прочерки на местах неуказанных букв.
- Визуализация виселицы должна добавляться постепенно с каждой неправильной попыткой.
- Игра завершается, когда слово угадано полностью или когда висельник полностью нарисован.
- Количество попыток ограничено и должно быть указано в начале игры.

### Нефункциональные требования

- Код должен быть написан ясно и структурировано в соответствии с требованиями, указанными в разделе "Требования к ДЗ" информационного блока.
- Программа должна иметь текстовый интерфейс.

### Описание входных и выходных данных

#### Ввод

- Ввод буквы осуществляется через стандартную консоль ввода.
- Ввод не должен быть чувствительным к регистру.

#### Вывод

- На каждом этапе игры должно отображаться текущее состояние угадываемого слова и виселицы.
- При успешном угадывании буквы нужно обновить состояние слова на экране.
- При неправильной букве нужно обновить изображение виселицы и указать кол-во оставшихся попыток.

### Инструкции по реализации

- Создайте словарь, содержащий список слов для загадывания.
- Реализуйте систему для случайного выбора слова из словаря в начале каждой игры.
- Средства для получения ввода пользователя и вывода состояния игры.
- Логiku определения правильности угаданной буквы.
- Логiku визуализации виселицы при каждой ошибке.

- Учтите факт сложности игры: не забудьте сделать настраиваемым количество допустимых ошибок, и чтобы эту настройку можно было легко изменять или сделать настройку зависимой от сложности слова.
- При реализации вам могут пригодиться следующие примитивы:
- Словарь: получение случайного слова.
- Игровая сессия: хранения ответа, текущего состояния ответа пользователя, максимального числа попыток и числа произведенных попыток.
- Результат угадывания (ADT): текущее состояние, количество сделанных попыток, максимальное количество попыток и сообщение для пользователя.

### Тестирование

- Напишите тесты для проверки правильности выбора слова из списка.
- Проверьте корректность отображения состояния игры после каждого ввода пользователя.
- Убедитесь, что введенные буквы корректно обрабатываются вне зависимости от их регистра.
- Игра не запускается, если загадываемое слово имеет некорректную длину.
- После превышения заданного количества попыток игра всегда возвращает поражение.
- Состояние игры корректно изменяется при угадывании/не угадывании.
- Проверка, что при отгадывании ввод строки длиной больше чем 1 (опечатка) приводит к повторному вводу, без изменения состояния.

### Ограничения и советы

- Ограничение: В игре должен быть использован стандартный текстовый ввод/вывод.
- Ограничение: Не использовать внешние библиотеки для управления состоянием игры и ввода/вывода.
- Совет: Не пытайтесь делать всё в одной функции/классе, подумайте как вы разделите код и будете его тестировать.
- Совет: Разделите логику игры и интерфейс на отдельные модули.
- Совет: Поделите логику на части и начните разработку с какой-то из них, например, выбора случайного слова или машины состояния игры.
- Совет: Тестируйте каждую часть приложения по мере ее реализации.
- Совет: Попробуйте представить, что будет если вызвать 2 или более экземпляра игры, подумайте над понятием игровой сессии

### Дополнительные материалы

- [https://en.wikipedia.org/wiki/Hangman\\_\(game\)](https://en.wikipedia.org/wiki/Hangman_(game))
- <https://habr.com/ru/companies/lanit/articles/660895/>
- <https://github.com/topics/terminal-game>