Класс **HangmanGame** реализует консольную версию игры "Виселица". В нем используется объектно-ориентированный подход (ООП) для организации кода.

1. **Структура класса:**
   * **Приватные члены:**
     + **words**: вектор для хранения слов.
     + **secretWord**: загаданное слово.
     + **guessedWord**: угаданное слово с пропущенными буквами.
     + **attempts**: количество попыток.
     + **maxAttempts**: максимальное количество попыток (задается как константа).
     + **startTime**: время начала игры.
   * **Публичные методы:**
     + **loadWordsFromFile()**: загрузка слов из файла.
     + **decrypt()**: расшифровка слова с помощью шифра Цезаря.
     + **selectSecretWord()**: выбор случайного загаданного слова.
     + **initializeGuessedWord()**: инициализация угаданного слова с пустыми местами для букв.
     + **displayGuessedWord()**: отображение текущего состояния угаданного слова.
     + **drawHangman()**: отображение виселицы в зависимости от количества попыток.
     + **guessLetter()**: угадывание буквы и обновление угаданного слова.
     + **isWordGuessed()**: проверка, угадано ли слово полностью.
     + **displayStatistics()**: отображение статистики игры.
     + **play()**: основной метод для запуска игры.
2. **Реализация методов:**
   * **loadWordsFromFile()** загружает слова из файла, расшифровывает их и добавляет в вектор **words**.
   * **decrypt()** выполняет расшифровку слова с помощью шифра Цезаря с фиксированным сдвигом.
   * **selectSecretWord()** выбирает случайное загаданное слово из вектора **words**.
   * **initializeGuessedWord()** создает угаданное слово с пропущенными буквами.
   * **displayGuessedWord()** отображает текущее состояние угаданного слова.
   * **drawHangman()** рисует виселицу в зависимости от количества неправильных попыток.
   * **guessLetter()** проверяет, содержится ли угаданная буква в загаданном слове, и обновляет угаданное слово.
   * **isWordGuessed()** проверяет, угадано ли слово полностью.
   * **displayStatistics()** выводит статистику игры.
   * **play()** осуществляет основной игровой процесс.
3. **Загрузка слов:**
   * Слова загружаются из файла **words.txt**, где они хранятся в зашифрованном виде с использованием шифра Цезаря с сдвигом 7.
4. **Безопасность и обработка ошибок:**
   * Программа предусматривает обработку ошибок при загрузке слов из файла и закрытии файла.
   * Предотвращены нежелательные попытки доступа к переменным через приватные члены класса.
5. **Кодирование и структурирование:**
   * Использование объектно-ориентированного подхода способствует четкости и структурированности кода.
   * Каждый метод выполняет четко определенную функцию, что облегчает чтение и поддержку кода.

Таким образом, представленный код обеспечивает полноценную реализацию игры "Виселица" с использованием ООП и шифра Цезаря для безопасного хранения слов.