МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 2\_2**

по дисциплине: “Системное программирование”

на тему: ***“***Многофайловые проекты и шаблон MVC ***”***

Вариант 1

Выполнил**:** студент группы 10702121 Меркулова М. С.

Принял**:** Давыденко Н.В.

Минск 2023

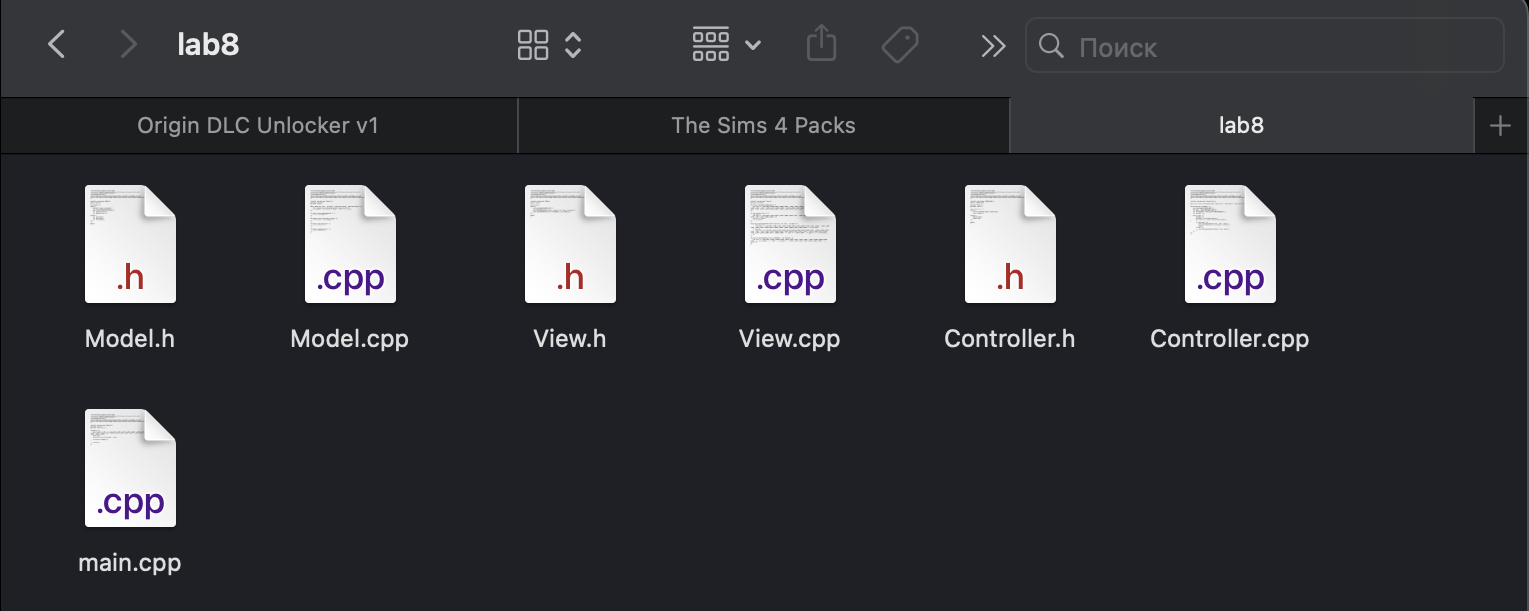
# Лабораторная работа №2\_1. Компиляция и отладка простейшего приложения в Linux

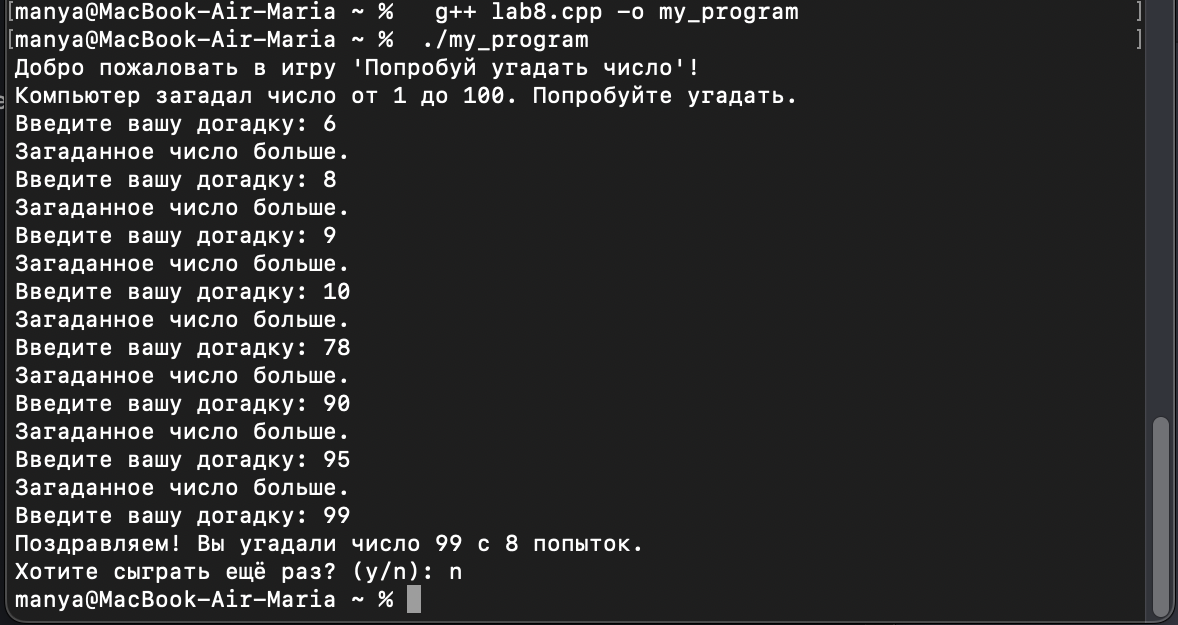
Цель работы: Научиться грамотно проектировать и разрабатывать многофайловые проекты на языке программирования С/C++ с использованием архитектурного шаблона проектирования и разработки гибких и масштабируемых приложений Model-View-Controller (MVC).

**Основное задание**

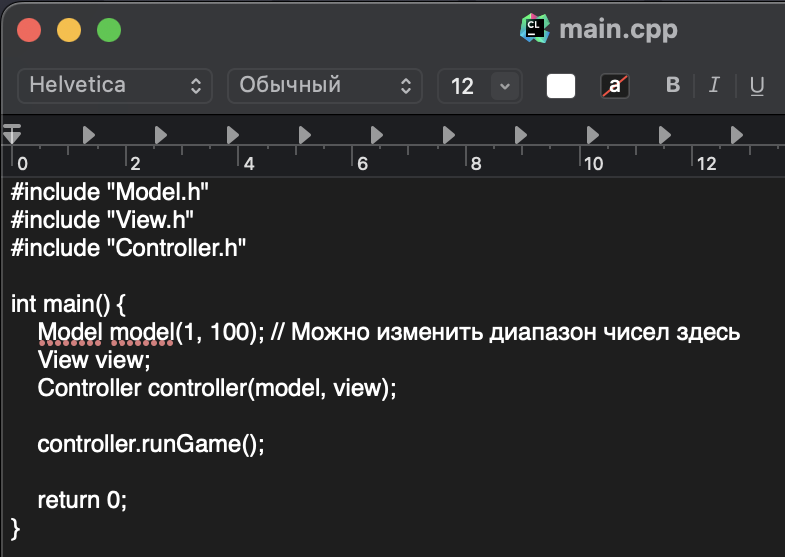
Разработать интерактивную программу «Try to Guess the Number» («Попробуй угадать число»), которая эмулирует классическую игру на отгадывание числа. Суть программы (игры) сводиться к следующему: компьютер генерирует слу- чайное число из диапазона, к примеру, от 1 до 100, а пользователь (далее иг- рок) пытается отгадать число за минимальное количество попыток. При каж- дой очередной попытке компьютер «подсказывает» игроку, как соизмеряется предложенный вариант игрока с действительным загаданным компьютером числом: загаданное число больше или меньше указанного (higher/lower). Как только игрок отгадывает число, компьютер должен «поздравить» его с выво- дом на экран угаданного числа и количество затраченных игроком попыток. Далее компьютер может «предложить» повторно сыграть игру или выйти из программы.

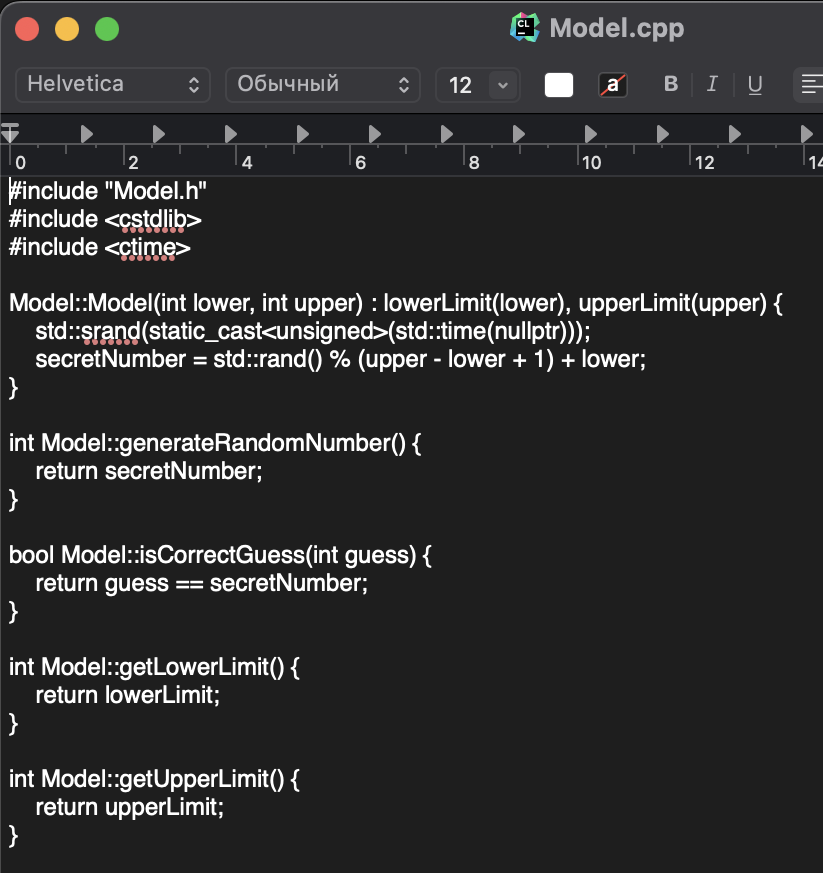
Для универсальности предложенной программы можно добавить возмож- ность выбора диапазона генерирования компьютером случайных чисел, а также задания ограничения на количество попыток. В случае, если игрок не укладывается в заданное количество попыток (т.е. проигрывает), программа должна выводить суровую надпись «Game Over».

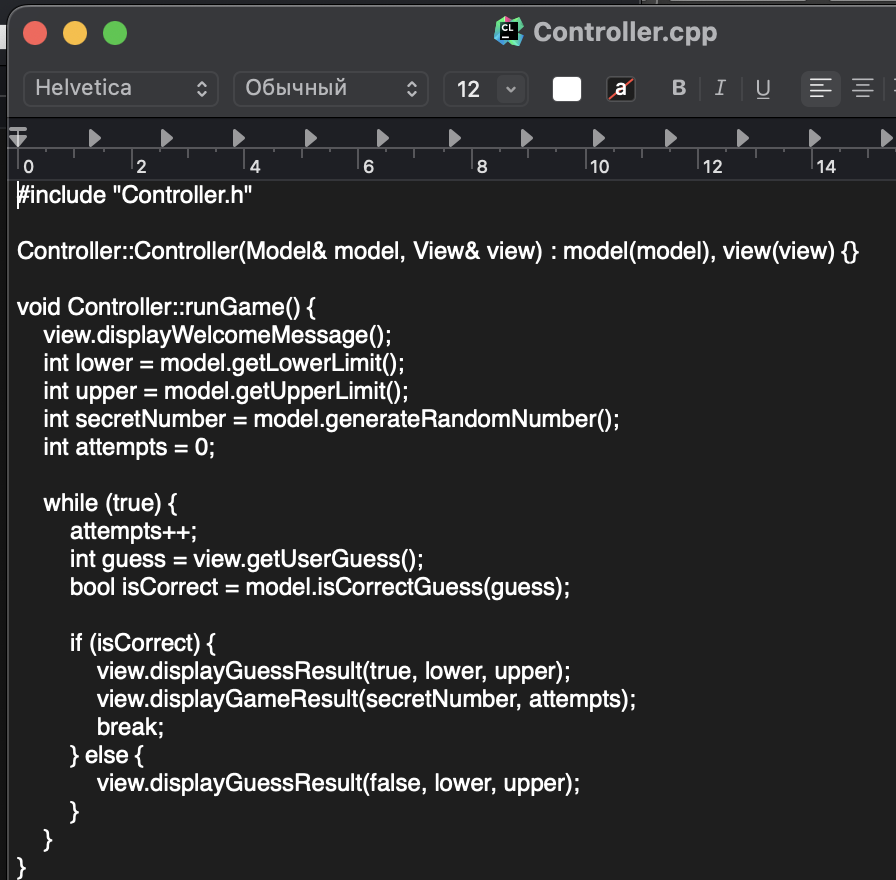
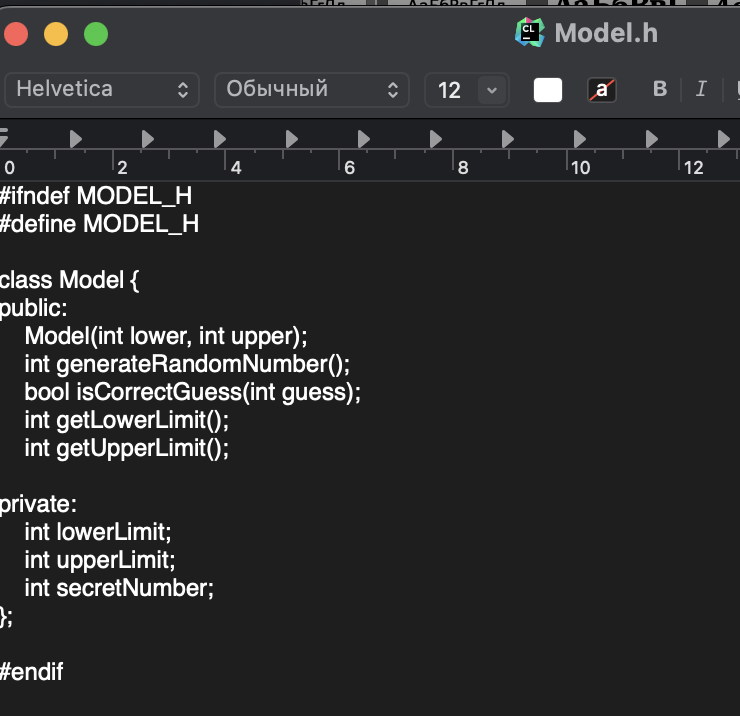


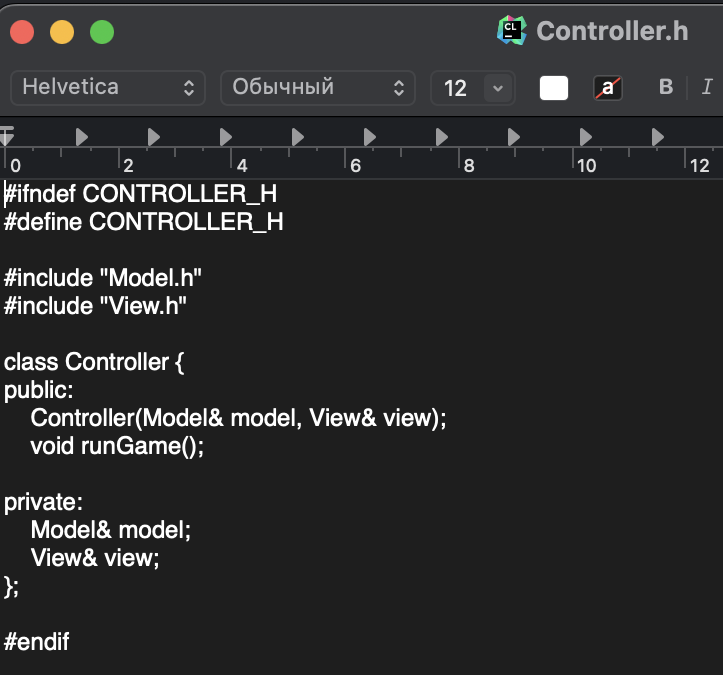


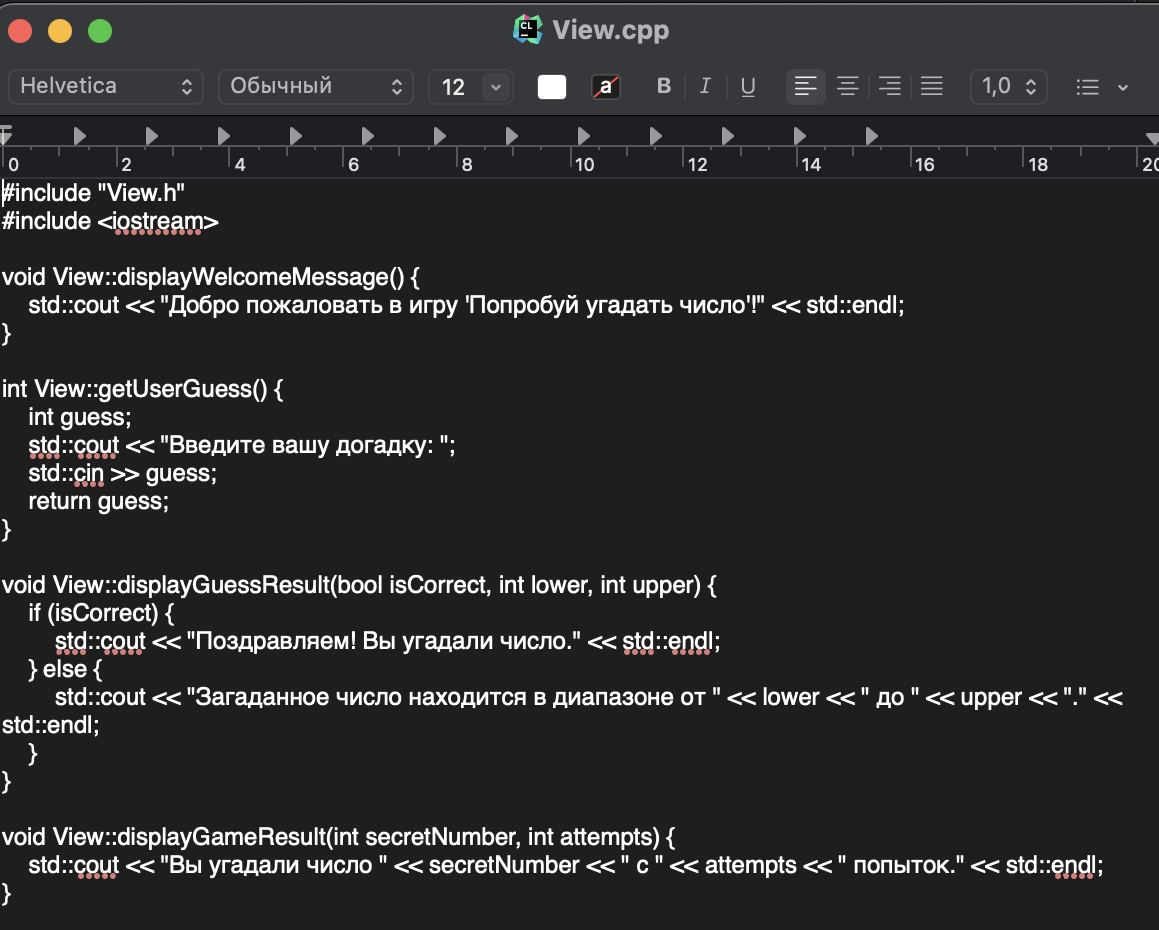
**Код программы**

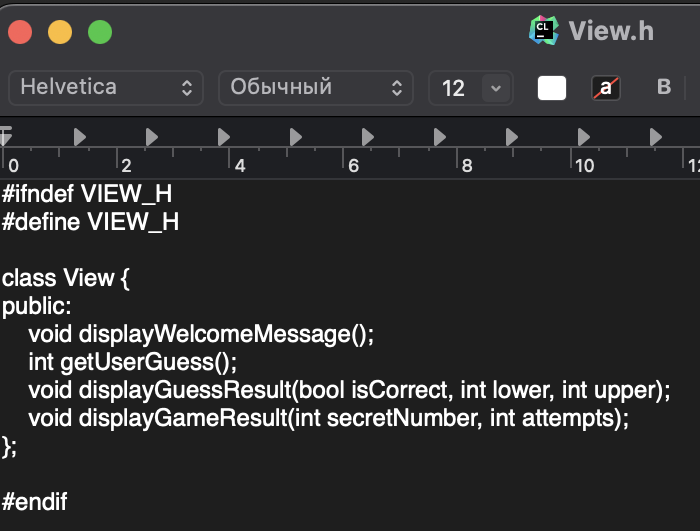












**Контрольный вопрос**

**Какие существенные преимущества имеют многофайловые проекты перед однофайловыми? Что такое сборка?**

Многофайловые проекты имеют несколько существенных преимуществ перед однофайловыми:

1. **Модульность и организация:** В многофайловых проектах вы можете организовать код логически, разделяя его на отдельные файлы, каждый из которых отвечает за определенные функции или компоненты. Это облегчает поддержку и управление проектом.
2. **Повторное использование кода:** Вы можете создавать библиотеки и модули, которые могут быть повторно использованы в других проектах, что сокращает дублирование кода.
3. **Командная работа:** В больших проектах разработка может выполняться несколькими разработчиками. Многофайловая организация позволяет разработчикам независимо работать над своими частями проекта.
4. **Улучшенная читаемость:** Многофайловая организация обычно обеспечивает лучшую читаемость и понимание кода, так как код разделен на более мелкие компоненты.
5. **Более быстрая компиляция:** В случае изменения только одного файла, не требуется перекомпилировать весь проект, а только измененный файл, что может значительно ускорить процесс компиляции.

Что такое сборка: Сборка (в контексте программирования) - это процесс создания исполняемой программы или библиотеки из ее исходных файлов. В многофайловых проектах сборка включает в себя компиляцию каждого исходного файла в объектные файлы, а затем связывание этих объектных файлов в исполняемую программу или библиотеку. Сборка обычно выполняется с использованием инструментов для сборки, таких как компиляторы и линковщики.

Сборка может включать в себя следующие этапы:

1. **Компиляция:** Исходные файлы переводятся в объектные файлы. Каждый исходный файл компилируется независимо от других.
2. **Линковка:** Объектные файлы связываются вместе для создания исполняемой программы или библиотеки. Линковка разрешает ссылки между объектными файлами и статическими библиотеками.
3. **Оптимизация:** Некоторые инструменты сборки могут выполнять оптимизации кода для улучшения производительности и уменьшения размера итоговой программы.

Сборка - это важная часть процесса разработки программного обеспечения, и она позволяет создавать готовые к использованию программы из исходного кода.

Начало формы