

Лабораторная работа #3

Многофайловые проекты и шаблон MVC

*[Model-View-Controller (MVC), Single Responsibility Principle (SRP),
g++/gcc, *.c, *.cc, *.h, *.o]*



ЛАБОРАТОРНАЯ РАБОТА # 3

Многофайловые проекты и шаблон MVC

Цель работы

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

Требования

- 1) Разработать многофайловый консольный проект на C/C++ согласно варианту задания с использованием шаблона (паттерна) проектирования **MVC**.
- 2) Размерность матрицы в индивидуальном задании задаётся пользователем на стадии выполнения программы. Для повышения производительности программы использовать динамическое выделение памяти.
- 3) ЗАПРЕЩАЕТСЯ в программе использовать под любым предлогом ГЛОБАЛЬНЫЕ переменные!
- 4) Каждое задание оформить в виде отдельной бизнес-функции.
- 5) Все функции должны быть сгруппированы по соответствующим отдельным файлам.
- 6) Все функции должны быть самодостаточные, т.е. при их разработке необходимо придерживаться принципа **Single Responsibility Principle**.
- 7) При выполнении задания **запрещается** использовать интегрированные средства разработки (*Integrated Development Environment, IDE*). Рекомендуется задействовать любой текстовый редактор (к примеру, **gedit**) и набор компиляторов GNU Compiler Collection (GCC), в частности, компиляторы языков программирования C/C++ **gcc/g++**, а для отладки – **gdb**-отладчик.
- 8) При разработке программ необходимо придерживаться соглашения по написанию кода на C/C++ (*Code-Convention*).

Основное задание

Разработать интерактивную программу «*Try to Guess the Number*» («Попробуй угадать число»), которая эмулирует классическую игру на отгадывание числа. Суть программы (игры) сводится к следующему: компьютер генерирует случайное число из диапазона, к примеру, от 1 до 100, а пользователь (далее игрок) пытается отгадать число за минимальное количество попыток. При каждой очередной попытке компьютер «подсказывает» игроку, как соотносится 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 выводом 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 возможность 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».

Индивидуальное задание

- 1) Поменять местами строку, содержащую элемент с наибольшим значением в матрице со строкой, содержащей элемент с наименьшим значением. Вывести на экран полученную матрицу. Для каждой строки с нулевым элементом на главной диагонали вывести ее номер и значение наибольшего из элементов этой строки.
- 2) Определить, является ли матрица симметричной (относительно главной диагонали). Вывести на экран соответствующее сообщение. Найти максимальный элемент среди стоящих на главной и побочной диагонали и поменять местами с элементом, стоящим на пересечении этих диагоналей.
- 3) Выведите номера столбцов, все элементы, которых четны. Для каждого столбца с отрицательным элементом на главной диагонали вывести его номер и сумму всех элементов этого столбца.
- 4) Среди строк заданной матрицы, содержащих только нечетные элементы, найти строку с максимальной суммой модулей элементов.

- 5) Среди столбцов заданной матрицы, содержащих только такие элементы, которые по модулю не больше n , найти столбец с минимальным произведением элементов.
- 6) Выведите номера столбцов, элементы каждой из которых образуют монотонную последовательность (монотонно убывающую или монотонно возрастающую).
- 7) Найти максимальный среди всех элементов тех строк заданной матрицы, которые упорядочены (либо по возрастанию, либо по убыванию).
- 8) Найти номер строки заданной матрицы, в которой находится самая длинная серия (последовательность одинаковых элементов).
- 9) Найти строку заданной матрицы, в которой длина максимальной серии (последовательности одинаковых элементов) минимальна.
- 10) Найти максимальное из чисел, встречающихся в заданной матрице более одного раза.
- 11) Найти минимальное из чисел, встречающихся в заданной матрице ровно один раз.
- 12) Найти максимальное из чисел, встречающихся в заданной матрице ровно два раза.
- 13) Для заданной целочисленной матрицы найти максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.
- 14) Для заданной целочисленной матрицы найти минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.
- 15) Подсчитать количество столбцов заданной матрицы, которые составлены из попарно различных чисел.
- 16) Подсчитать количество строк заданной матрицы, являющихся перестановкой чисел 1, 2, ..., 20.
- 17) Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Соседями элемента a_{ij} в матрице назовем элементы a_{kl} с $i-1 \leq k \leq i+1$, $j-1 \leq l \leq j+1$, $(k,l) \neq (i,j)$. Подсчитать количество локальных минимумов заданной матрицы.
- 18) Найти максимум среди всех локальных минимумов заданной матрицы. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Соседями элемента a_{ij} в матрице назовем элементы a_{kl} с $i-1 \leq k \leq i+1$, $j-1 \leq l \leq j+1$, $(k,l) \neq (i,j)$.

- 19) Найти минимум среди всех локальных максимумов заданной матрицы. Элемент матрицы называется локальным максимумом, если он строго больше всех имеющихся у него соседей. Соседями элемента a_{ij} в матрице назовем элементы a_{kl} с $i-1 \leq k \leq i+1, j-1 \leq l \leq j+1, (k,l) \neq (i,j)$.
- 20) Определить, становится ли симметричной (относительно главной диагонали) заданная матрица после замены на число 0 каждого локального минимума. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Соседями элемента a_{ij} в матрице назовем элементы a_{kl} с $i-1 \leq k \leq i+1, j-1 \leq l \leq j+1, (k,l) \neq (i,j)$.
- 21) Определить, становится ли симметричной (относительно главной диагонали) заданная матрица после замены на число 0 каждого локального максимума. Элемент матрицы называется локальным максимумом, если он строго больше всех имеющихся у него соседей. Соседями элемента a_{ij} в матрице назовем элементы a_{kl} с $i-1 \leq k \leq i+1, j-1 \leq l \leq j+1, (k,l) \neq (i,j)$.
- 22) Две строки матрицы назовем похожими, если совпадают множества чисел, встречающихся в этих строках. Найти количество строк в максимальном множестве попарно непохожих строк заданной матрицы.
- 23) Определить, является ли действительная матрица размера $m \times n$ ортонормированной, т.е. такой, в которой скалярное произведение каждой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно 1. Вывести на экран соответствующее сообщение.

Best of LUCK with it, and remember to HAVE FUN while you're learning :)

Victor Ivanchenko



Контрольные вопросы

1. Объясните основные термины (*объявление (declaration)*, *определение (definition)*, *вызов (call)*), которые связаны с описанием и использованием функций в языке C/C++, приведите пример. Зачем нужно определение функции?
2. Какие существенные преимущества имеют многофайловые проекты перед однофайловыми? Что такое сборка?
3. Почему в C/C++ раздельная компиляция? Опишите её преимущества и недостатки.
4. Опишите общий алгоритм процесса сборки многофайлового проекта и стадии его компиляции и выполнения.
5. Что всегда будет присутствовать (сопровождать) в любой востребованной программе?
6. Что такое рефакторинг кода?
7. Опишите концепцию проектирования приложений с использованием шаблона Model-View-Controller. Зачем он применяется? Какие преимущества даёт использование данного шаблона при проектировании и реализации программного проекта?
8. Объясните принцип "*Единственной ответственности*" (*Single Responsibility Principle, SRP*), который используется для проектирования и реализации независимого или слабосвязанного кода.
9. Что такое нисходящее и восходящее программирование?
10. В чём преимущество и для чего нужно использовать в C/C++ проектах файлы-заголовки? Какие проблемы они решают? С помощью какой программной конструкции языка можно предотвратить бесконечное (рекурсивное) включение заголовочных файлов друг в друга?