**20.5 Практическая работа**

Задача

Цели домашнего задания

Приобрести практическое понимание работы с файлами.

Научиться:

* открывать файлы на запись по указанному пути и закрывать их;
* записывать разные типы данных в файлы;
* записывать массивы данных в файлы;
* пользоваться генератором случайных чисел.

Что входит в домашнее задание

1. Реализовать запись в ведомость учёта.
2. Реализовать рисование случайных картин.
3. Реализовать симуляцию игры «Рыбалка».
4. Реализовать симуляцию работы банкомата.

Общие условия

Разработка выполняется в среде [CLion](https://www.jetbrains.com/clion" \t "_blank). Текстовые документы можно редактировать в стандартном блокноте.

**Задание 1. Реализация записи в ведомость учёта**

**Что нужно сделать**

В одном из прошлых заданий мы уже создавали программу чтения и анализа ведомости выплат. Теперь требуется создать простую программу записи в эту ведомость.

Формат ведомости прежний. Сначала идёт имя и фамилия получателя денежных средств, далее располагается дата выдачи в формате ДД.ММ.ГГГГ. Завершает запись сумма выплаты в рублях. Данные разделяются между собой пробелами. В конце каждой записи должен быть расположен перевод строки.

При старте программы пользователь последовательно вводит данные для новой записи, которые записываются затем в файл в текстовом режиме. Программа должна добавлять новые записи в конец файла, не удаляя его текущее содержимое.

**Советы и рекомендации**

Введённые данные рекомендуется хотя бы минимально валидировать перед их записью.

Для валидации даты в указанном формате можно использовать функции std::stoi и std::subtr с корректно указанным диапазоном символов в дате. К примеру, вызов std::stoi(date.substr(3, 2)) вычленит из строковой даты целочисленный месяц для последующего анализа.

**Что оценивается**

Корректность работы программы. Программа должна дозаписывать данные в конец существующего списка.

**Как отправить задание на проверку**

Прислать ссылку на repl.it или файл .срр с решением.

**Задание 2. Реализация рисования случайных картин**

**Что нужно сделать**

С помощью генератора случайных чисел рисуется картинка из нулей и единиц. На вход от пользователя принимается размер картины: высота и ширина в пикселях. На выходе нужно создать файл pic.txt, содержимое которого будет наполнено случайными нулями и единицами в рамках размера картины.

Это ещё не всё. Если вы запустите программу несколько раз на одних и тех же исходных данных (ширина и высота картины), вы очень скоро обнаружите, что всё время выводится одна и та же картинка. Дело в том, что у генератора случайных чисел есть ещё и «зерно», от которого он отталкивается. Если это зерно не меняется, то и серия из rand() всегда будет возвращать одни и те же случайные числа.

Чтобы разнообразить вывод, вы можете использовать вызов std::srand(std::time(nullptr)); в самом начале программы. Эта операция задаёт начальное зерно случайных чисел в зависимости от текущего времени. Для её корректной работы не забудьте также включить заголовочный файл <ctime>.

**Что оценивается**

Корректность работы программы по созданию случайных изображений заданного размера.

**Как отправить задание на проверку**

Прислать ссылку на repl.it или файл .срр с решением.

**Задание 3. Реализация симуляции игры «Рыбалка»**

**Что нужно сделать**

Реализуйте простую симуляцию игры в рыбалку. В исходном текстовом файле реки (river.txt) задаётся список из видов рыб, которые можно в ней поймать. Рыбок может быть сколько угодно, разных видов. Виды при этом могут повторяться.

В начале программы пользователь указывает, какую именно рыбу он сейчас будет ловить — вид этой рыбы. После этого из первого файла друг за другом осуществляется чтение его содержимого, вид за видом. Если на отдельных шагах вид совпал с указанным пользователем, в выходной файл basket.txt (корзинка) записывается этот вид.

В конце программы показывается, сколько было поймано рыб за текущую ловлю. Программу можно запускать несколько раз, при этом уже пойманные рыбы должны сохраняться в файле-корзинке.

**Пример содержания исходного файла**

sunfish  
shad  
carp  
bass  
bullhead  
carp  
walleye  
catfish  
carp

**Что оценивается**

Корректность работы программы симуляции игры в рыбалку. Пойманные рыбы должны записываться в выходной файл.

**Как отправить задание на проверку**

Прислать ссылку на repl.it или файл .срр с решением.

**Задание 4. Реализация симуляции банкомата**

**Что нужно сделать**

Требуется реализовать упрощённую симуляцию работы банкомата. В банкомате могут храниться только бумажные купюры номиналом от 100 до 5000 рублей. Максимально в банкомате может храниться только 1000 купюр. Все они записываются в отдельный файл. У банкомата, как у устройства, две основных операции — снятие денег пользователем и наполнение деньгами инкассаторами банка.

Наполнение банкомата происходит, если при старте программы в консоль вводится символ плюс “+”.  Количество купюр рассчитывается так, чтобы банкомат был заполнен полностью. Все купюры при этом выбираются случайным образом.

Если на старте программы в консоль вводится минус (“-”), то симулируется снятие пользователем денег. Пользователь указывает сумму с точностью до 100 рублей. Мы будем считать, что каждый клиент обладает неограниченным балансом в системе и теоретически может снять любую сумму. На практике, если данная сумма не может быть снята из-за отсутствия подходящих денег в машине, показывается сообщение, что эта операция невозможна.

После выполнения любой из операций программа завершает выполнение. Состояние банкомата должно храниться в отдельном бинарном файле, автономно.

**Советы и рекомендации**

Вы можете хранить массив купюр целиком, помечая отсутствующие позиции нулями.

**Что оценивается**

Корректность работы программы симуляции банкомата. Инкассация и снятие денег должны работать корректно.

**Как отправить задание на проверку**

Прислать ссылку на repl.it или файл .срр с решением.

Оцените материалы практической работы