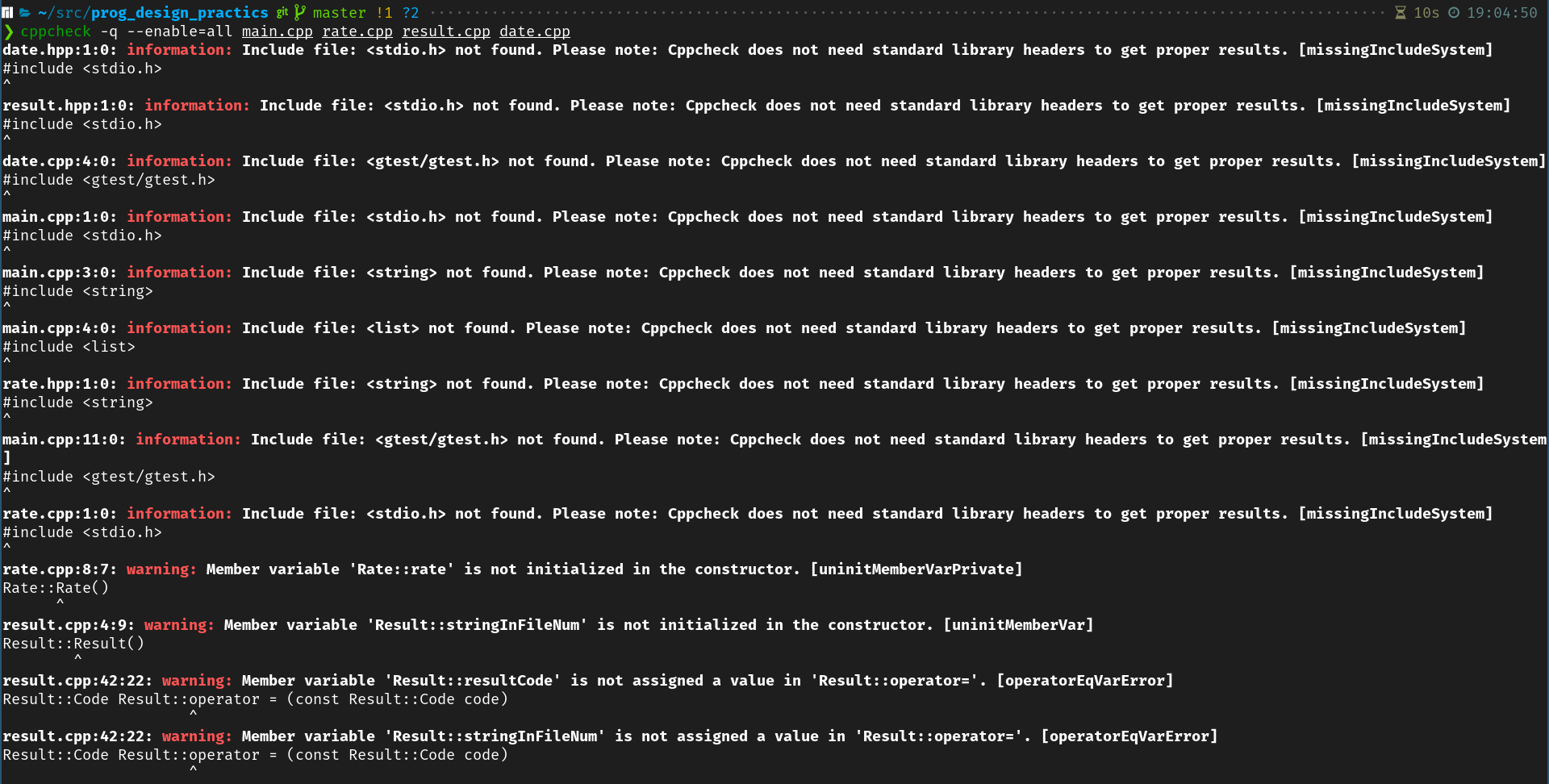
Статический анализ

В качестве статического анализатора использована программа cppcheck. Запуск без дополнительных параметров не выявил ошибок, поэтому анализ запущен повторно с включением всего, что похоже на ошибку. Результат работы программы представлен на рисунке 1.

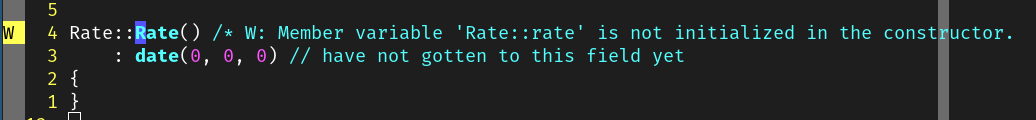


1. Результат работы cppcheck

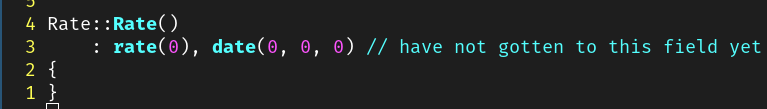
Большинство ошибок произошли из-за того, что cppcheck не смогла найти файлы стандартной библиотеки C++. Это можно обойти, указав путь к каталогам, их содержащим, но тогда ошибок выводится многократно больше, причём все новые ошибки относятся к коду внутри библиотек.

Последние 4 пункта относятся непосредственно к анализируемой программе и должны быть рассмотрены подробнее:

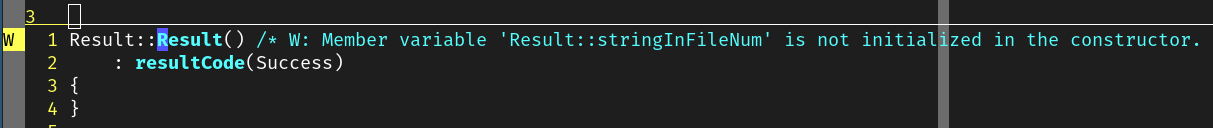
1. Говорит о том, что в конструкторе по умолчанию класса Rate не присваивается значение переменной rate. Ошибка не влияет на работу программы, но ради увеличения стабильности программы при доработке её следует исправить.
2. Аналогично для переменной stringInFileNum класса Result.
3. Говорит, что в перегруженном операторе присваивания в классе Result не присваивается значение переменной resultCode. Анализатор, по-видимому, не очень хорошо справляется с «True C code style», поскольку значение этой переменной присваивается, причём это единственное, что происходит в коде данной перегрузки. Тем не менее, её стоит привести к более понятной анализатору и человеку форме.
4. Говорит, что в рассмотренной ранее перегрузке не присваивается также значение переменной stringInFileNum. Это не является ошибкой, по замыслу данной перегрузки значение этой переменной должно оставаться прежним.



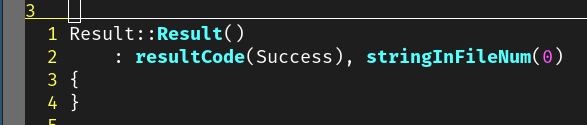
1. Первая ошибка



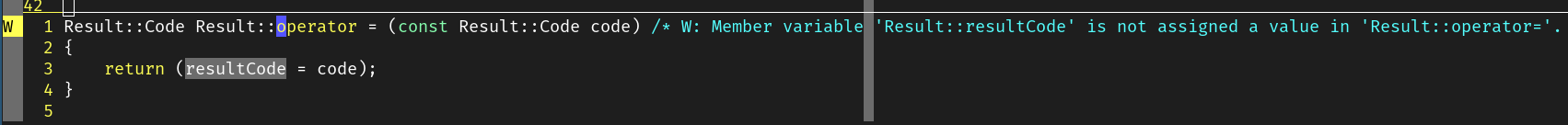
1. После исправления первой ошибки



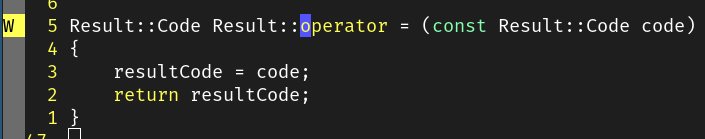
1. Вторая ошибка



1. После исправления второй ошибки

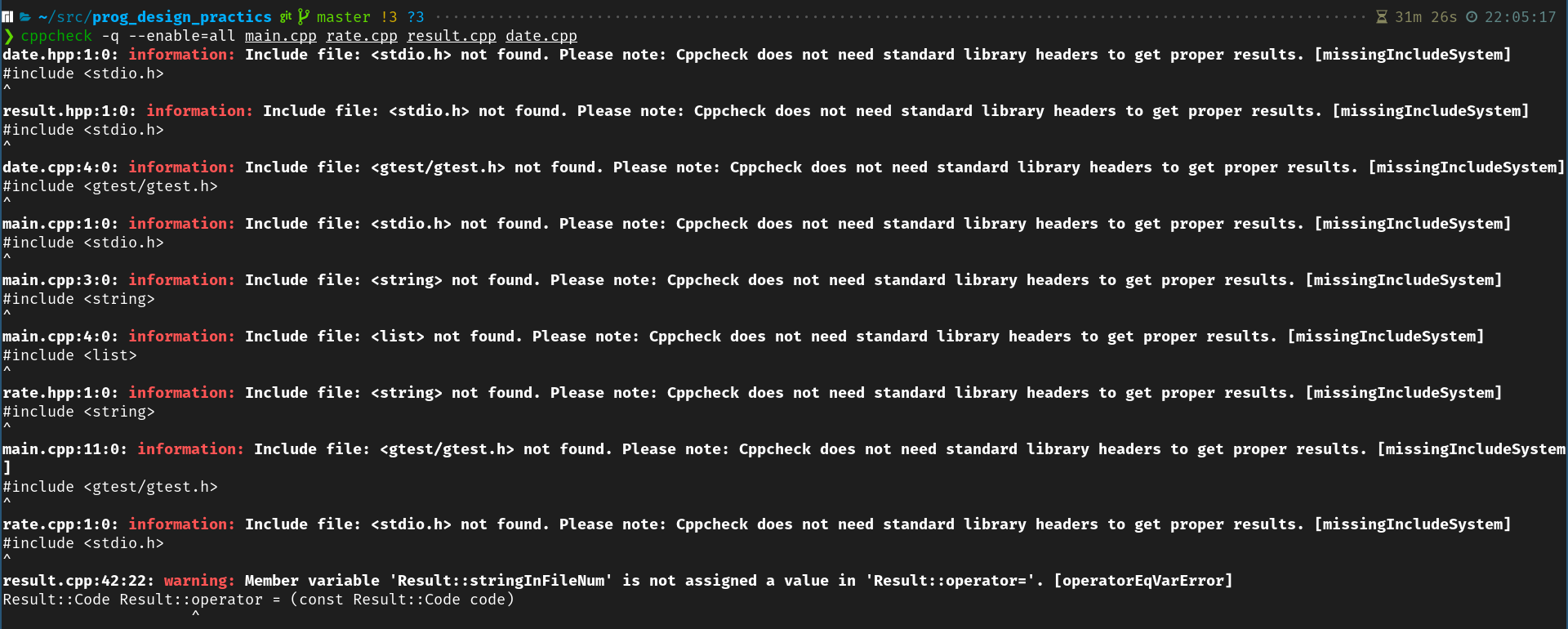


1. Третья ошибка



1. После исправления третьей ошибки

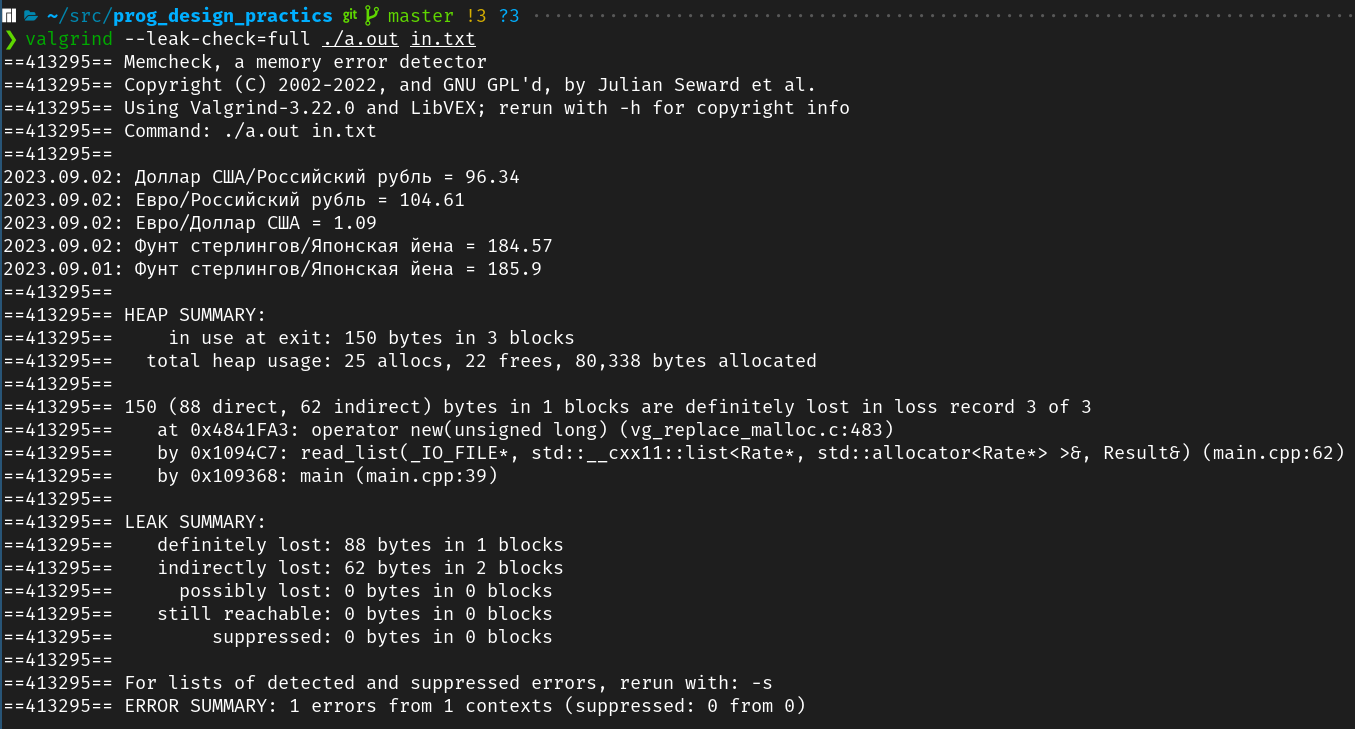
После внесения исправлений cppcheck даёт результат, представленный на рисунке 8. Как видно, новых ошибок не внесено, а оставшиеся, как показано ранее, на самом деле не являются ошибками.



1. Результат работы cppcheck после внесения исправлений

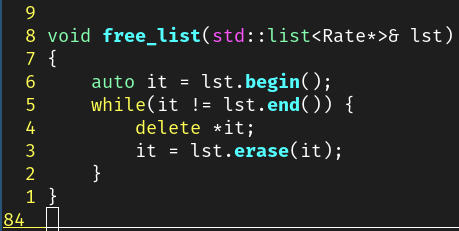
Динамический анализ

В качестве динамического анализатора задействована программа valgrind с инструментом memcheck (по умолчанию) с подробным описанием каждой найденной утечки.



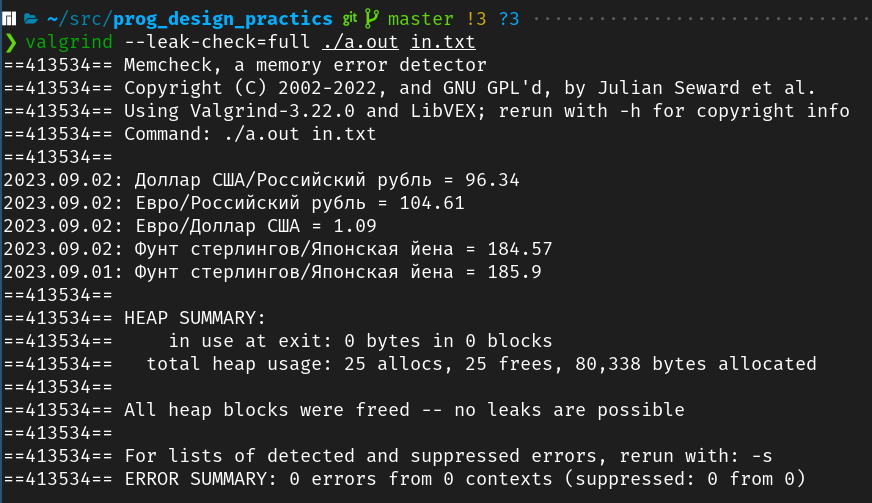
1. Результат работы valgrind memcheck

Программой обнаружена утечка памяти из-за того, что в функции main() не производится освобождение выделенной в куче памяти для объектов класса Rate, хранящихся в контейнере std::list. В данной программе это не является существенной ошибкой, так как указанная память используется до самого завершения работы программы, и освобождена может быть лишь непосредственно перед завершением, что не имеет смысла, так как операционная система самостоятельно освободит всю выделенную программе память. Тем не менее, ради упрощения дальнейшей доработки кода следует добавить процедуру освобождения памяти и её вызов в конце функции main().



1. Добавленная процедура

После указанных исправлений утечек не обнаружено:



1. Результат работы valgrind memcheck после внесения исправлений