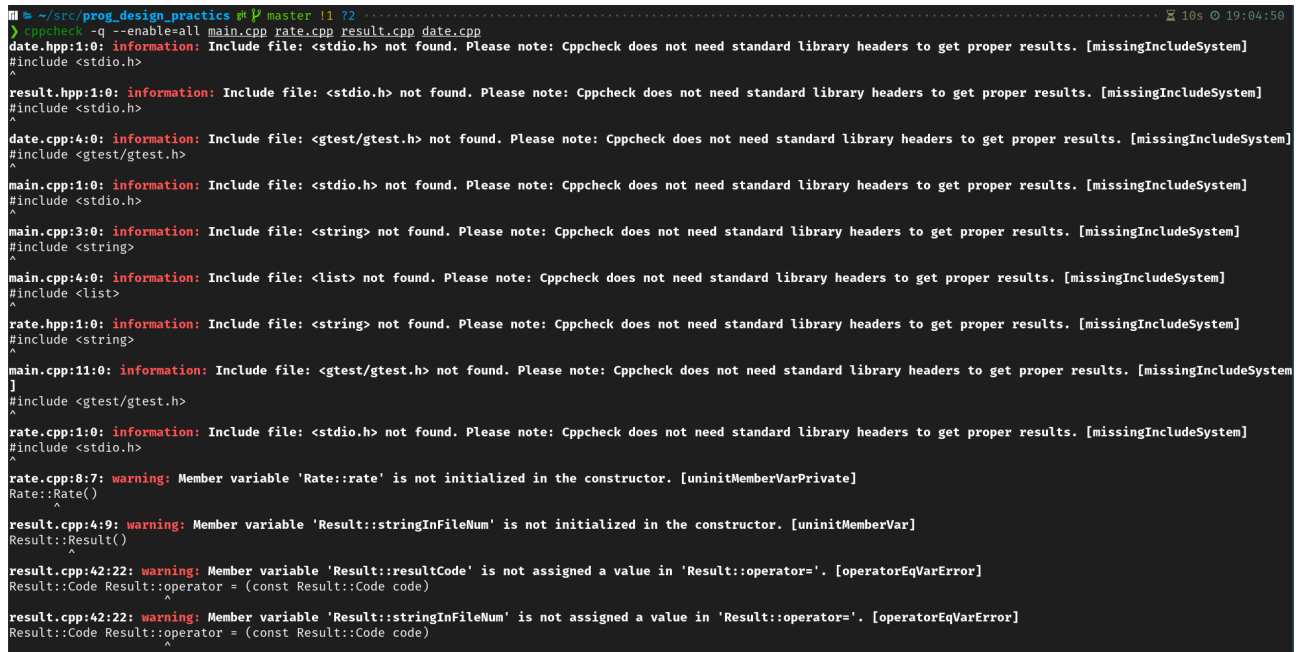


Статический анализ

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.



```
~/src/prog_design_practices # master !1 72
> cppcheck -q --enable=all main.cpp rate.cpp result.cpp date.cpp
date.hpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
result.hpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
date.cpp:4:0: information: Include file: <gtest/gtest.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <gtest/gtest.h>
^
main.cpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
main.cpp:3:0: information: Include file: <string> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <string>
^
main.cpp:4:0: information: Include file: <list> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <list>
^
rate.hpp:1:0: information: Include file: <string> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <string>
^
main.cpp:11:0: information: Include file: <gtest/gtest.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <gtest/gtest.h>
^
rate.cpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
rate.cpp:8:7: warning: Member variable 'Rate::rate' is not initialized in the constructor. [uninitMemberVarPrivate]
Rate::Rate()
^
result.cpp:4:9: warning: Member variable 'Result::stringInFileNum' is not initialized in the constructor. [uninitMemberVar]
Result::Result()
^
result.cpp:42:22: warning: Member variable 'Result::resultCode' is not assigned a value in 'Result::operator='. [operatorEqVarError]
Result::Code Result::operator = (const Result::Code code)
^
result.cpp:42:22: warning: Member variable 'Result::stringInFileNum' is not assigned a value in 'Result::operator='. [operatorEqVarError]
Result::Code Result::operator = (const Result::Code code)
^
```

Рисунок 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.

```
5
W 4 Rate::Rate() /* W: Member variable 'Rate::rate' is not i
3   : date(0, 0, 0) // have not gotten to this field yet
2 {
1 }
```

Рисунок 2 – Первая ошибка

```
5
4 Rate::Rate()
3   : rate(0), date(0, 0, 0) // have not gotten to this field yet
2 {
1 }
```

Рисунок 3 – После исправления первой ошибки

```
3
W 1 Result::Result() /* W: Member variable 'Re
2   : resultCode(Success)
3 {
4 }
```

Рисунок 4 – Вторая ошибка

```
3
1 Result::Result()
2   : resultCode(Success), stringInFileNum(0)
3 {
4 }
```

Рисунок 5 – После исправления второй ошибки

```

42
W 1 Result::Code Result::operator = (const Result::Code code) /* W:
   2 {
   3     return (resultCode = code);
   4 }
   5

```

Рисунок 6 – Третья ошибка

```

6
W 5 Result::Code Result::operator = (const Result::Code code)
   4 {
   3     resultCode = code;
   2     return resultCode;
   1 }

```

Рисунок 7 – 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.

```

~/src/prog_design_practices # master 13 ?3
> cppcheck -q --enable=all main.cpp rate.cpp result.cpp date.cpp
date.hpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
result.hpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
date.cpp:4:0: information: Include file: <gtest/gtest.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <gtest/gtest.h>
^
main.cpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
main.cpp:3:0: information: Include file: <string> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <string>
^
main.cpp:4:0: information: Include file: <list> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <list>
^
rate.hpp:1:0: information: Include file: <string> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <string>
^
main.cpp:11:0: information: Include file: <gtest/gtest.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <gtest/gtest.h>
^
result.cpp:1:0: information: Include file: <stdio.h> not found. Please note: Cppcheck does not need standard library headers to get proper results. [missingIncludeSystem]
#include <stdio.h>
^
result.cpp:42:22: warning: Member variable 'Result::stringInFileNum' is not assigned a value in 'Result::operator='. [operatorEqVarError]
Result::Code Result::operator = (const Result::Code code)
                        ^

```

Рисунок 8 – 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.

```

~/src/prog_design_practics master 13 ?3
> valgrind --leak-check=full ./a.out in.txt
==413295== Memcheck, a memory error detector
==413295== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==413295== Using Valgrind-3.22.0 and LibVEX; rerun with -h for copyright info
==413295== Command: ./a.out in.txt
==413295==
2023.09.02: Доллар США/Российский рубль = 96.34
2023.09.02: Евро/Российский рубль = 104.61
2023.09.02: Евро/Доллар США = 1.09
2023.09.02: Фунт стерлингов/Японская йена = 184.57
2023.09.01: Фунт стерлингов/Японская йена = 185.9
==413295==
==413295== HEAP SUMMARY:
==413295==   in use at exit: 150 bytes in 3 blocks
==413295== total heap usage: 25 allocs, 22 frees, 80,338 bytes allocated
==413295==
==413295== 150 (88 direct, 62 indirect) bytes in 1 blocks are definitely lost in loss record 3 of 3
==413295==   at 0x4841FA3: operator new(unsigned long) (vg_replace_malloc.c:483)
==413295==   by 0x1094C7: read_list(_IO_FILE*, std::__cxx11::list<Rate*, std::allocator<Rate*> >&, Result&) (main.cpp:62)
==413295==   by 0x109368: main (main.cpp:39)
==413295==
==413295== LEAK SUMMARY:
==413295==   definitely lost: 88 bytes in 1 blocks
==413295==   indirectly lost: 62 bytes in 2 blocks
==413295==   possibly lost: 0 bytes in 0 blocks
==413295==   still reachable: 0 bytes in 0 blocks
==413295==   suppressed: 0 bytes in 0 blocks
==413295==
==413295== For lists of detected and suppressed errors, rerun with: -s
==413295== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)

```

Рисунок 9 – 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().

```

9
8 void free_list(std::list<Rate*>& lst)
7 {
6     auto it = lst.begin();
5     while(it != lst.end()) {
4         delete *it;
3         it = lst.erase(it);
2     }
1 }
84

```

Рисунок 10 – Добавленная процедура

После указанных исправлений утечек не обнаружено:

```
~/src/prog_design_practics git master !3 ?3
> valgrind --leak-check=full ./a.out in.txt
==413534== Memcheck, a memory error detector
==413534== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==413534== Using Valgrind-3.22.0 and LibVEX; rerun with -h for copyright info
==413534== Command: ./a.out in.txt
==413534==
2023.09.02: Доллар США/Российский рубль = 96.34
2023.09.02: Евро/Российский рубль = 104.61
2023.09.02: Евро/Доллар США = 1.09
2023.09.02: Фунт стерлингов/Японская йена = 184.57
2023.09.01: Фунт стерлингов/Японская йена = 185.9
==413534==
==413534== HEAP SUMMARY:
==413534==       in use at exit: 0 bytes in 0 blocks
==413534==   total heap usage: 25 allocs, 25 frees, 80,338 bytes allocated
==413534==
==413534== All heap blocks were freed -- no leaks are possible
==413534==
==413534== For lists of detected and suppressed errors, rerun with: -s
==413534== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Рисунок 11 – Результат работы valgrind memcheck после внесения исправлений