# **Отчет по лабораторной работе №5**

**«Средства автоматизации тестирования программного обеспечения»**

Выполнил студент группы 19-КБ-ПР2 Хаммаш М.

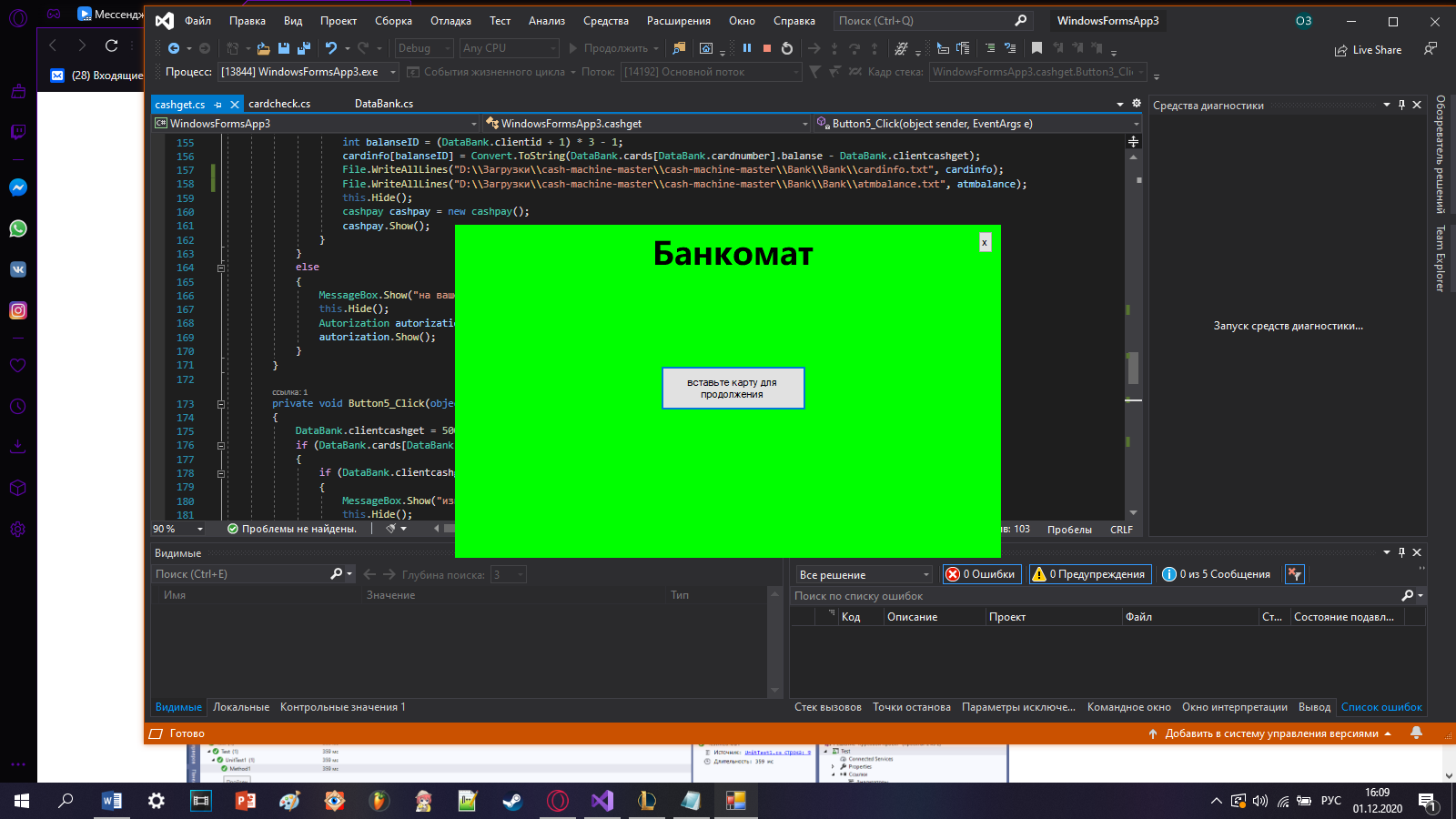
**Цель работы**: разработать тесты для приведенных классов, проводить тестирования программы, профилирование программы.

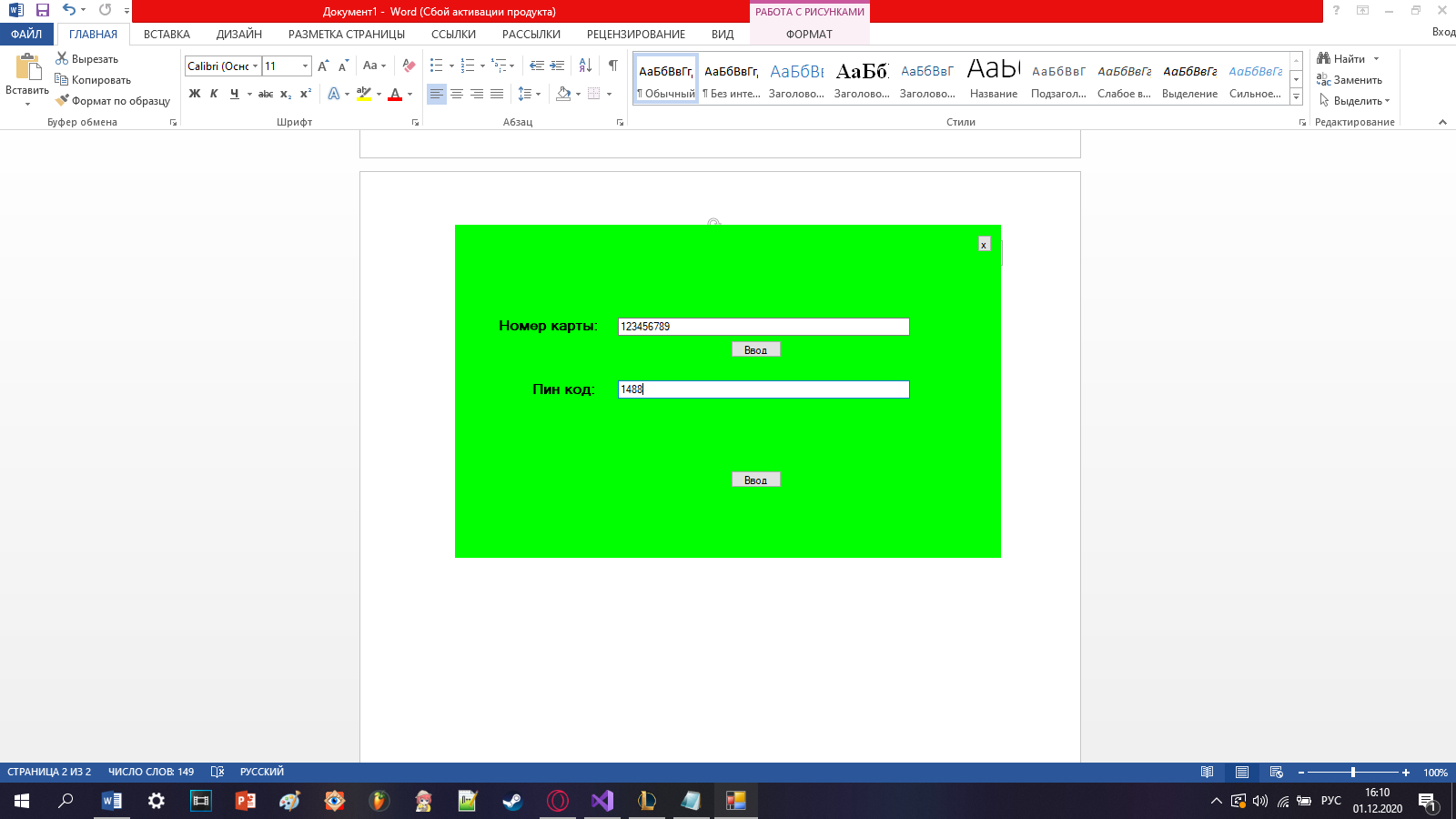
**Введение**: в процессе выполнения лабораторной работы мы желаем научиться выполнять тестирования и профилирование программы.

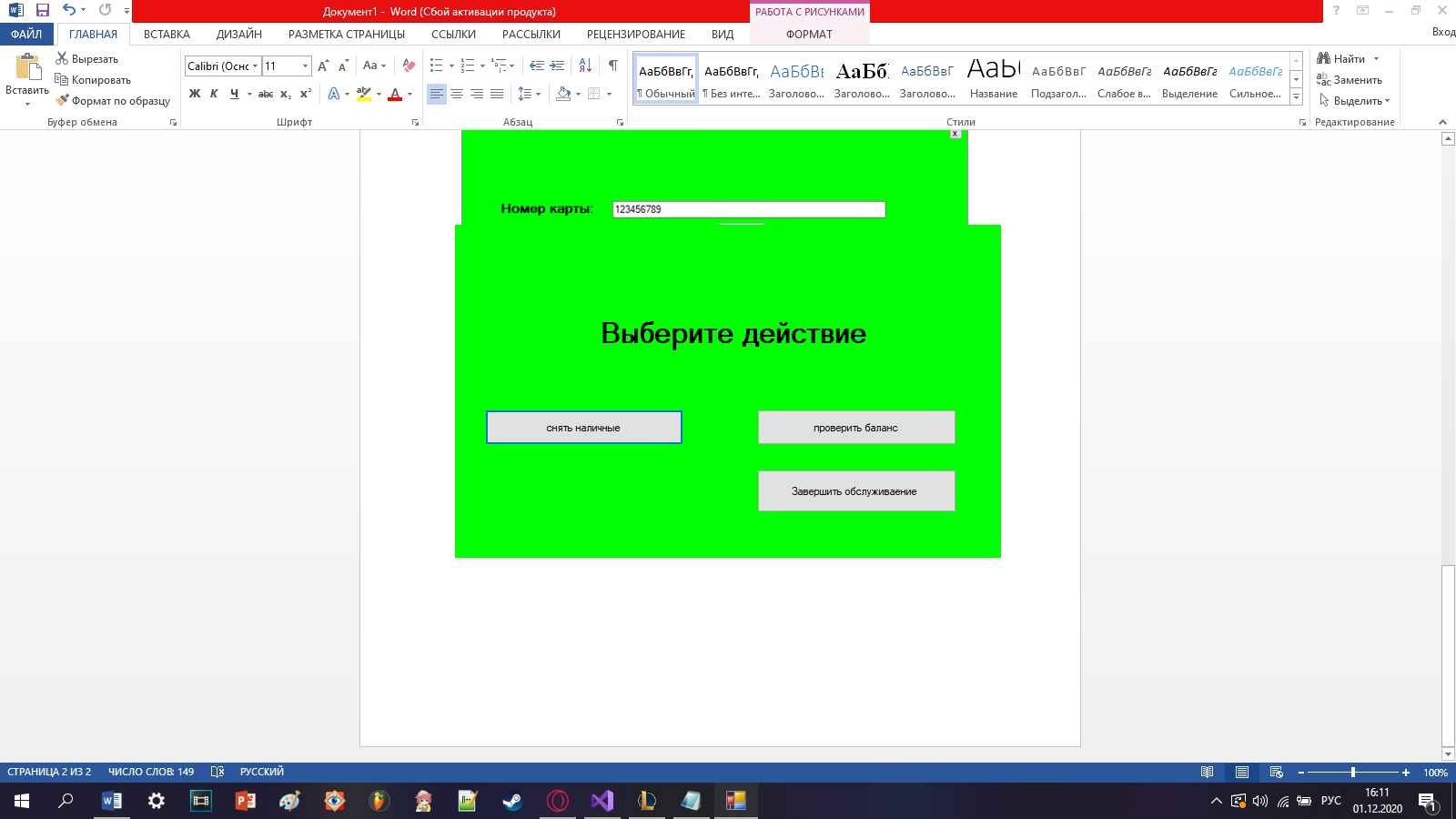
**Порядок выполнения лабораторной работы**:

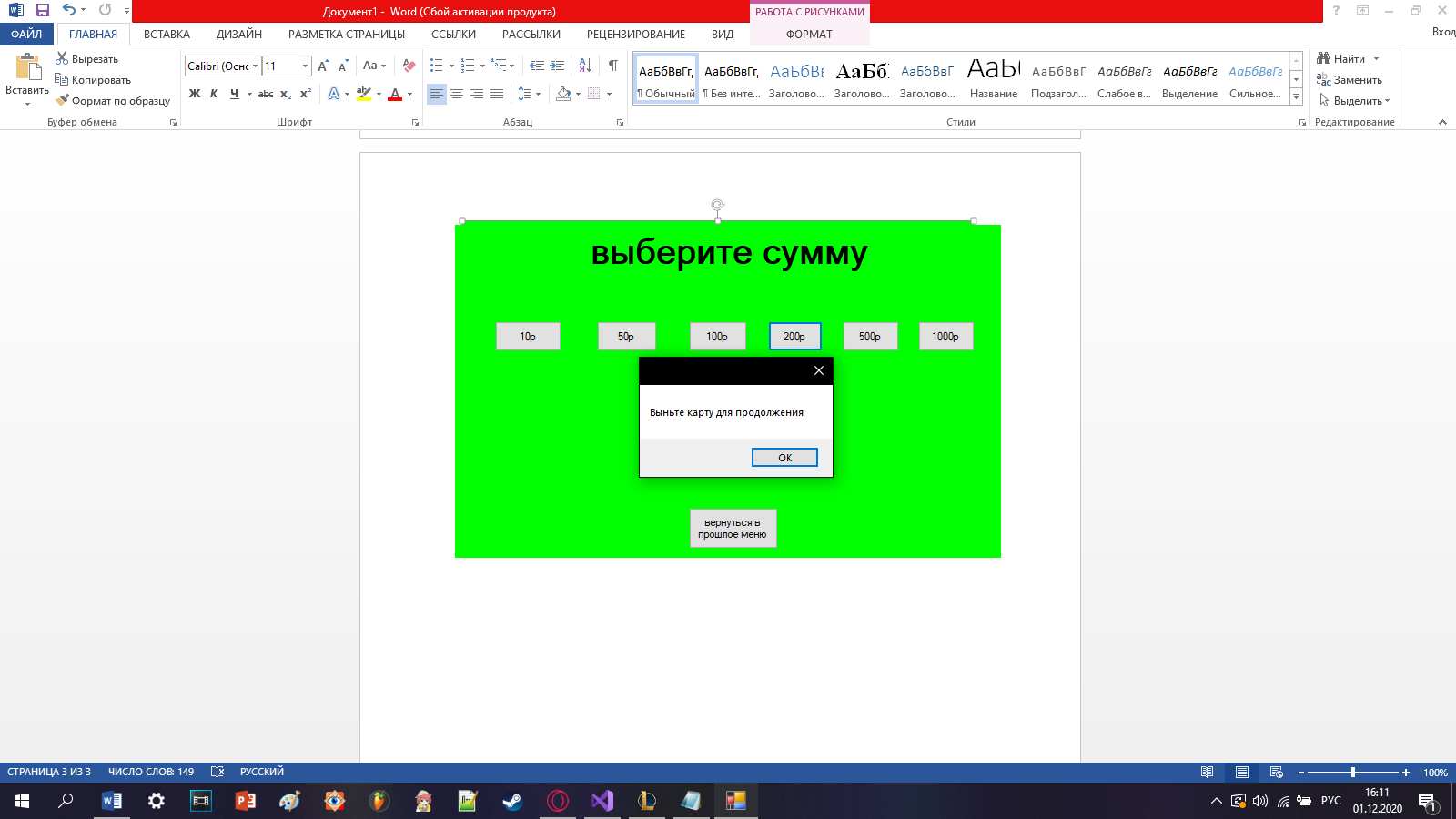
1. Создайте исходный код для классов на приведенных выше диаграммах.
2. Напишите автоматические тесты для проверки работоспособности программы в области вставки, обновления, удаления, поиска записей в базе данных.
3. Выполните тесты.
4. Внесите искусственно ошибку в программу.
5. Выполните повторно тесты. Сформируйте отчет о найденных ошибках. Исправьте ошибки. Выполните повторно тесты.
6. Выполните профилирование программы. Проконтролируйте использование оперативной памяти, процессора.
7. Определите узкие места программы, сформулируйте рекомендации по увеличению производительности программы за счет оптимизации узких мест.

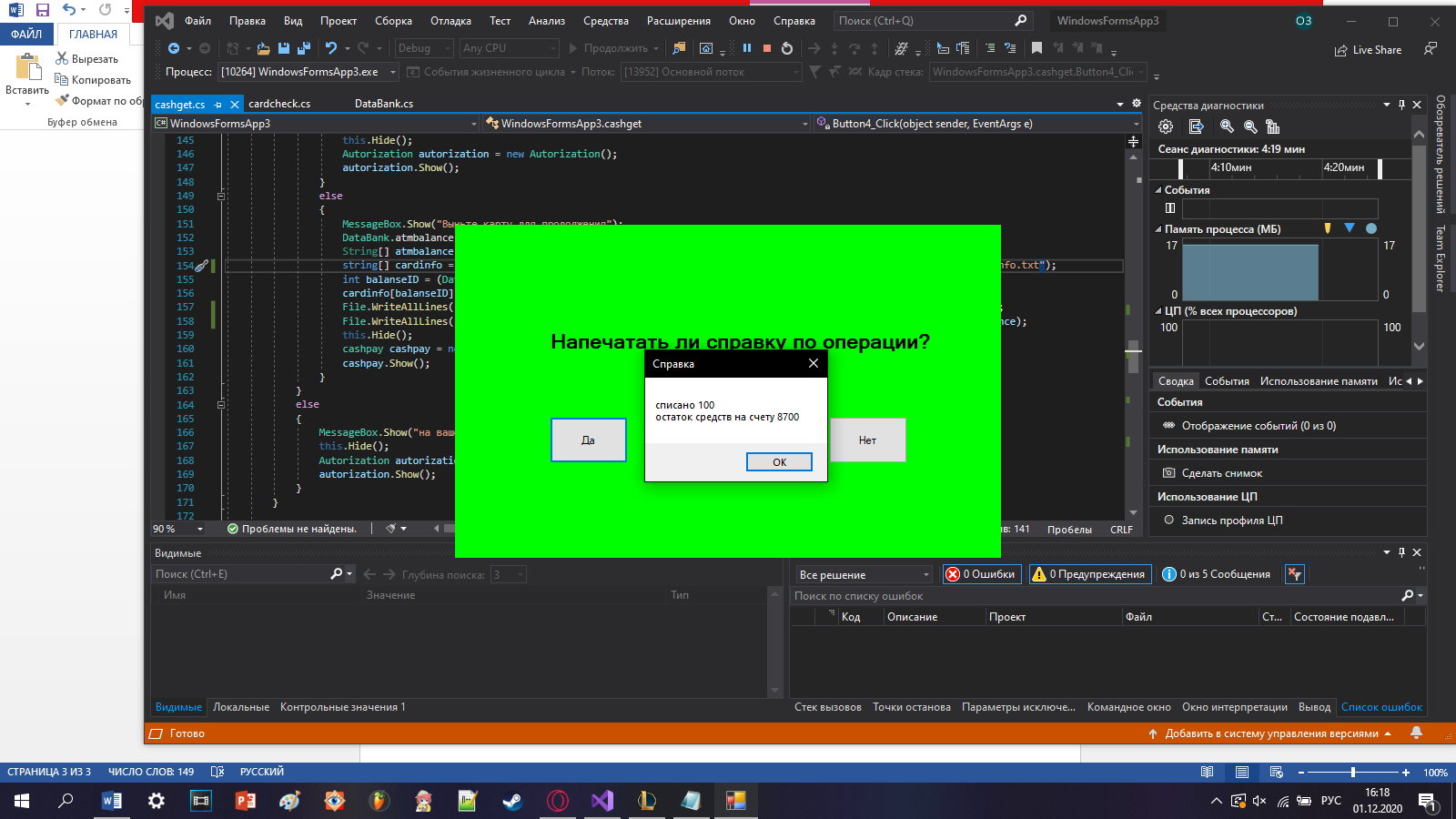
Тестирование по «Черному ящику» и тестирование по «Белому ящику»  отличаются тем что по «Белому ящику»  каждый шаг проверяется поочередно а по «Черному ящику» полностью.



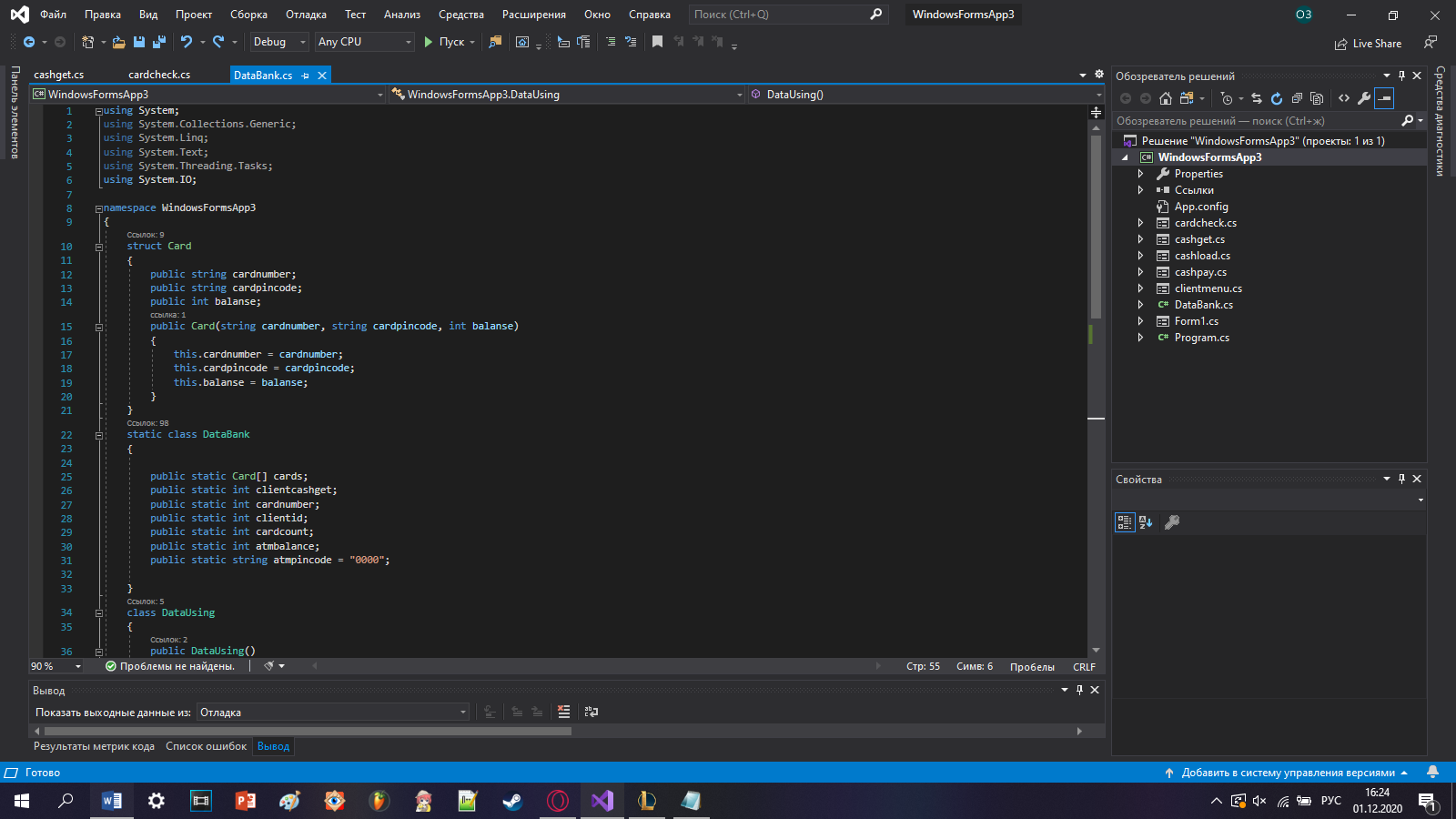




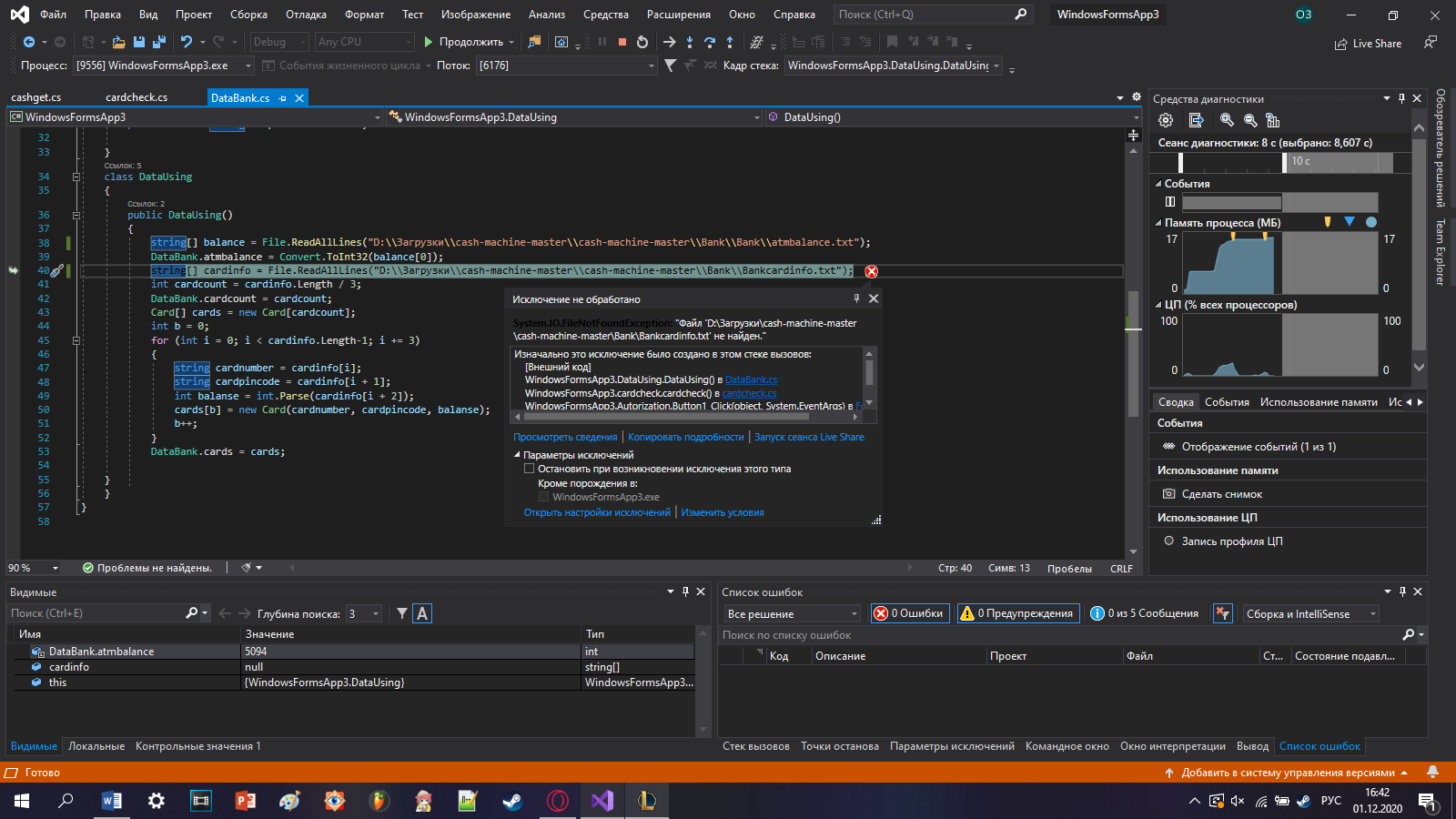




Основной метод, в котором происходило тестирование блока кода

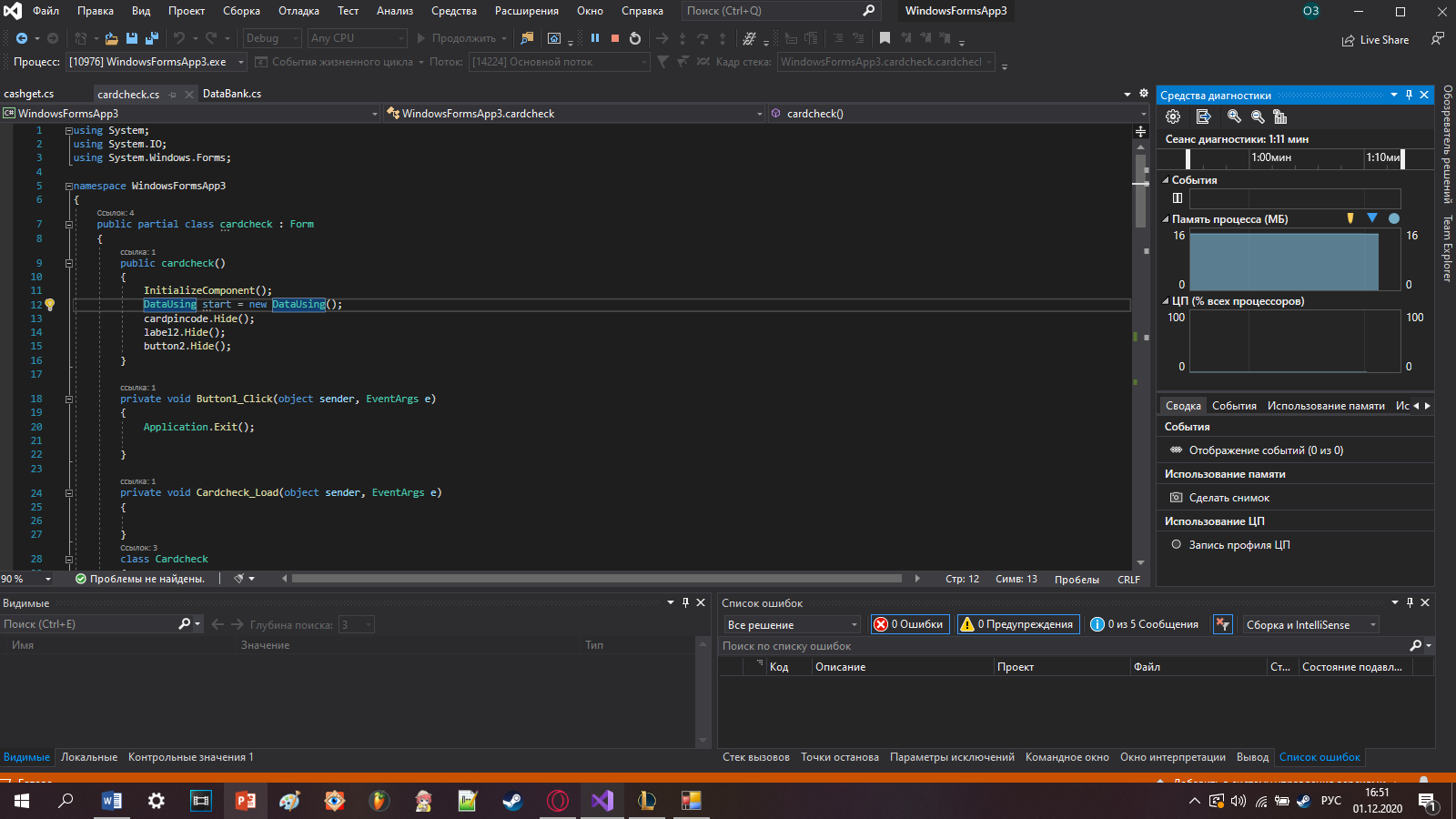


Искуственное внесение ошибки в программу.

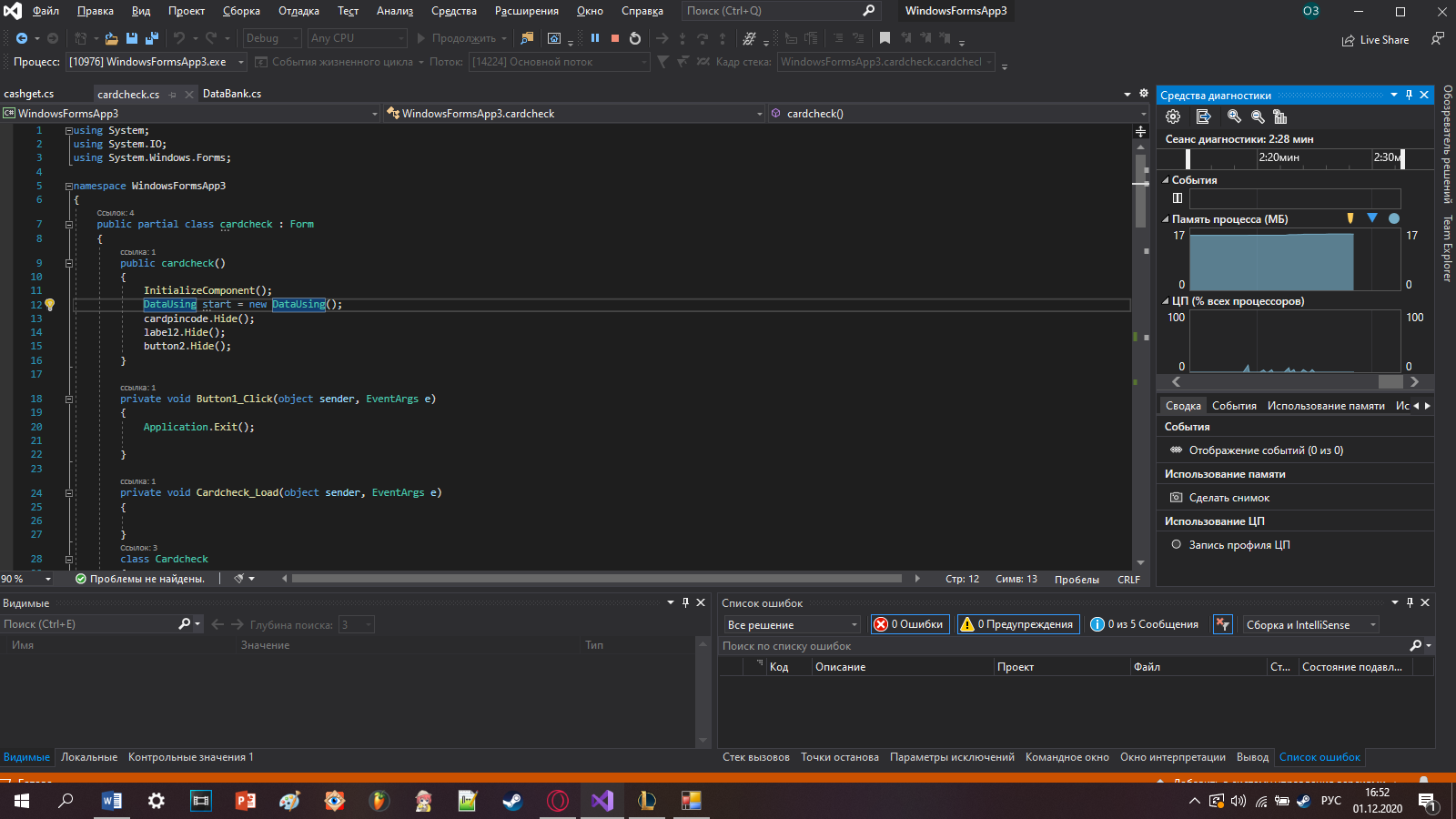


Ошибка была за счет нарушения пути получения информации нужной для работы терминала.

Использование памяти процессора.



Использование ЦП



**Заключение**:

в процессе выполнения лабораторной работы мы научились делать тестирование программы по принципу «Черного» и «Белого» ящика. В качестве входных данных в тестировании по принципу «Черного» ящика были кнопки, которые записывали в отдельную переменную информацию о том, какая кнопка была нажата и в качестве выходных данных была информация о том, какие кнопки были нажаты (когда уже заканчивался процесс стирки), которая выводилась на экран. Тестирование по принципу «Белого» ящика заключалось в том, что при нажатии на любую кнопку программы сразу выводилась на экран информация о том, какая кнопка была нажата. Тестирование отдельных блоков программы происходило благодаря встроенному тестировщику в VS Unit Test. Так, в качестве основного блока, который мы использовали для тестирования был метод подсчета времени стирки. Во время тестирования мы выявили несколько ошибок, которое было связано с подсчетом времени программы. Далее, мы научились пользоваться встроенным профилировщиком в VS. Он помог отследить загрузку процессора на каждом шаге выполнения программы, также он показал то, сколько было кэш-промахов и то, насколько сильно каждый метод нагружал процессор. Также благодаря ему мы отслеживали загрузку памяти во время простоя программы , во время нажатия кнопок и во время стирки (делая снимки состояния программы). Во время тестирования были устранены слабые места, единственное, что можно оптимизировать в программе – уменьшить количество условных переходов, ибо их в программе немало и некоторые из них можно оптимизировать.

**Рекомендуемая литература**

1. Тестирование по принципу черного ящика: <https://habr.com/ru/post/462837/>
2. Тестирование по принципу белого ящика: <https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/ruchnoe-testirovanie/testirovanie-belogo-iashchika>
3. Юнит-тестирование: <https://habr.com/ru/post/169381/>
4. Профилировщик производительности в VS: <https://docs.microsoft.com/ru-ru/visualstudio/profiling/profiling-feature-tour?view=vs-2019>