Министерство науки и среднего профессионального образования
Российской Федерации ПОУ «Волго-Вятский колледж
информатики, финансов, права и управления»

Отчёт

на тему: **«Внедрение и тестирование
программного обеспечения»**

Выполнил:
студент ИСП-41
Поздеев Владислав Евгеньевич

Киров
2024

Содержание

Введение

Целью практики было обучение тестированию программного обеспечения. Во время практики я научился писать юнит-тесты по принципу белой коробки на языке программирования Rust. Тестирование это важный этап создания надёжного программного обеспечения, ведь в течении разработки сложной взаимосвязанной системы, результат изменения одной части приложения может затронуть остальные. Этап тестирования позволяет предостеречь появление таких неожиданных багов.

В ходе разработки был создан парсер для обработки особого формата конфигурации VPN. Конфигурация имеет иерархическую структуру включающую себя:

1. Комментарии в любой части документа
2. Блоки декларации, которые включают в себя:
    2.1. Константы, и другие декларации

Такая структура имеет также название "Рекурсивная композиция"[1]

Из-за того что сложно написать корректный парсер, который парсит всю структуру целиком было принято решение раздробить парсер на отдельные части, которые удобно разрабатывать и тестировать. В качестве методологии разработки была выбрана TDD (test-driven development)[2].

Сам парсер написан при помощи особой техники: леворекурсивные монадические парсер-комбинаторы, используя библиотеку nom
Парсер разделён на две части: ядро (написанно на языке программирования Rust) и удобный API на языке программирования Python

1 Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес; пер. с англ. - СПб.: Питер, 2022. - Текст : непосредственный. Глава «Рекурсивная композиция»

2 Википедия : свободная энциклопедия. - URL: https://ru.wikipedia.org/wiki/Разработка_через_тестирование (дата обращения: 12.12.2024). - Текст : электронный.

Описание тестов

Тесты нацелены на отдельные компоненты парсера, такие как парсинг комментариев, парсинг блоков деклараций, и констант. Отдельные компоненты парсера тестируются при помощи юнит-тестов: с помощью встроенной библиотеки тестирования Rust. Итоговый парсер тестируется вручную при помощи тестовых конфигураций.

Дата тестирования: 11.12.2024

Инициалы тестировщика: _____

Шаг	Элемент тестирования	Хорошо/ плохо
1.	Запуск юнит-тестов. В корне ядра проекта необходимо ввести комманду `cargo test`	Хорошо
2.	Вывод результата тестирования пасринга констант	Хорошо
3.	Вывод результата тестирования парсинга комментариев	Хорошо
4.	Вывод результата тестирования парсинга объявления деклараций	Хорошо
5.	Тестирование парсера целиком, при помощи API для языка программирования Python	Хорошо
6.	Развёртывание виртуального окружения: В корне проекта необходимо ввести следующую комманду: `python3.8 -m venv .venv (имя бинарного файла может отличаться от операционной системы)`	Хорошо
7.	Установка пакетов: `pandas, pandas-stubs, openpyxl`	Хорошо
8.	Установка ядра: см. readme.md	Хорошо
9.	Запуск файла тестирования time python ./src	Хорошо

Время тестирования юнит-тестов: 0.00 секунд

Время тестирования парсинга конфигурации: 0.484 секунды

Вывод лога консоли тестирования можно получить в «Приложение 1» и в «Приложение 2»

Заключение

В ходе практики были успешно достигнуты поставленные цели по изучению тестирования программного обеспечения. В процессе разработки парсера для VPN конфигураций был получен практический опыт написания юнит-тестов по методологии "белого ящика" на языке программирования Rust. Применение подхода TDD (разработка через тестирование) позволило эффективно разделить сложную задачу парсинга на маленькие подзадачи и обеспечить надежность каждого компонента системы.

Были успешно реализованы и протестированы ключевые компоненты парсера:

- Парсинг комментариев с различными вариантами форматирования
- Обработка констант с учетом типа, ключа и значения
- Парсинг определений с корректной обработкой пробельных символов

Все разработанные тесты успешно выполняются, что подтверждает корректность работы созданных компонентов. Полученные навыки тестирования и практический опыт применения TDD будут полезны в дальнейшей разработке надежного программного обеспечения.

Приложение 1. Вывод запуска юнит-тестов

```
wennerryle@fedora:~/Рабочий стол/projects/vpn_config_parser_core$ cargo test
    Finished `test` profile [unoptimized + debuginfo] target(s) in 0.02s
     Running unittests src/lib.rs (target/debug/deps/vpn_config_parser-
c21bb072b2f8bc76)

running 3 tests
test lexems::declare::constant::test_constant_parsing ... ok
test lexems::hash_comment::test_hash_comment_parsing ... ok
test lexems::declare::definition::test_parsing_declare_definition ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out;
finished in 0.00s

wennerryle@fedora:~/Рабочий стол/projects/vpn_config_parser_core$
```

Приложение 2. Часть вывода парсинга конфигурации

```
(.venv) wennerryle@fedora:~/Рабочий стол/projects/vpn_config_parser$ time python ./src/
                 name                                key    type                      value
0   PRIVET_FROM_PARSER!                AuthNtLmSecureHash    byte          +rnKF9SywQ==
1   PRIVET_FROM_PARSER!                      AuthPassword    byte          HhQa9FXamE=
2   PRIVET_FROM_PARSER!                          AuthType    uint                       1
3   PRIVET_FROM_PARSER!                       CreatedTime  uint64           1688358278279
4   PRIVET_FROM_PARSER!                        ExpireTime  uint64           1689681600000
..                  ...                               ...     ...                     ...
67       adevyatiyarova  Traffic.SendTraffic.BroadcastBytes  uint64             1642299
68       adevyatiyarova  Traffic.SendTraffic.BroadcastCount  uint64                5665
69       adevyatiyarova    Traffic.SendTraffic.UnicastBytes  uint64            18822400
70       adevyatiyarova    Traffic.SendTraffic.UnicastCount  uint64              134842
71       adevyatiyarova                       UpdatedTime  uint64           1682650115763

[72 rows x 4 columns]

real    0m0,484s
user    0m0,846s
(.venv) wennerryle@fedora:~/Рабочий стол/projects/vpn_config_parser$
```