1. **Спам-бот Rambler**

Реализовать по методологии TDD (Test-Driven Development) консольное приложение, предоставляющее возможность отправлять письма на указанный почтовый ящик сервиса rambler.ru. Обеспечить подгрузку содержания и темы писем из файла spam\_content.txt. В файле должно содержаться не менее 50 записей по образцу:

*ТЕМА\_ПИСЬМА – СОДЕРЖАНИЕ* (кириллица и латиница, формат UTF-8, всё в одной строке).

Обеспечить возможность покрытия 100% кода юнит-тестами. Предусмотреть все возможные исключительные ситуации (слишком большое время отклика, обрыв соединения, некорректный формат письма и т.д.). Предоставить возможность протоколирования (в свободной форме) каждого события в программе (отправка письма, получение ответа от сервера и т.д.) в файле log.txt. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

Что погуглить: протокол SMTP

1. **(\*) Интерпретатор командной строки**

Реализовать по методологии TDD (Test-Driven Development) консольное приложение – пользовательскую оболочку командной строки. Предусмотреть следующие команды для работы с файловой системой:

* shw – просмотр каталога
* cdir <path> - переход в каталог, указанный в <path>
* mdir <name> - создание нового каталога с именем, указанном в <name>. В случае отсутствия параметра <name> присвоить new
* crfile <name> - создание нового каталога с именем, указанном в <name>, прибавить постфикс .txt. В случае отсутствия параметра <name> присвоить new.txt
* shfile <name> - открытие файла в блокноте
* del <name> - безвозвратное удаление файла или каталога. Предусмотреть дополнительную проверку подкаталогов в каталоге и предупреждение пользователя о удалении файла и возможностью отказаться от действия
* help <command\_name> - строка помощи для каждой <command\_name>.

Рекомендации к выполнению: общая архитектура ЯДРО-ИНТЕРПРЕТАТОР, каждую комманду в отдельный класс, реализующие интерфейс IExecutable.

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

Что погуглить: стандарт POSIX

1. **Виджет “Прогноз погоды”**

Реализовать по методологии TDD (Test-Driven Development) графическое приложение, предоставляющее возможность отслеживать температуру воздуха в С, F, влажность воздуха и атмосферное давление и ясность/облачность/дождь/снег в Минске в реальном времени. Предусмотреть функции работы с API выбранного Вами онлайн-сервиса прогноза погоды и вывода данных на виджет.

Рекомендации к выполнению: общая архитектура СУБЪЕКТ-НАБЛЮДАТЕЛЬ.

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

Что погуглить: паттерн Наблюдатель

1. **(\*) Авиадиспетчер**

Реализовать по методологии TDD (Test-Driven Development) консольное приложение – эмулятор авиадиспетчера. Самолёт (С) движется по оси X с выбранной скоростью в секунду, даны два авиадиспетчера (А и Б) на концах отрезка, каждый из них контролирует свою часть.

А\_\_\_\_\_\_\_\_\_\_\_\_С->\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Б

Предусмотреть следующие функции: периодический опрос самолёта, периодическая выдача диспетчерами информации о расстоянии до границы обзора (|) или до аэропорта, автоматическая подписка или отписка пилота от обновлений.

Рекомендации к выполнению: общая архитектура СУБЪЕКТ-НАБЛЮДАТЕЛЬ, отдельные потоки на время и на обновления.

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

Что погуглить: паттерн Наблюдатель

1. **Шифратор AES**

Реализовать по методологии TDD (Test-Driven Development) консольное приложение, предоставляющее пользователю шифровать текстовые сообщения по алгоритму шифрования AES (своя реализация). Тестовые данные для каждого этапа шифрования (тестовые вектора) можно найти в стандарте FIPS-197. Предусмотреть следующие функции: генерация ключа, подгрузка исходного сообщения из файла source.txt, вывод результата в result.txt.

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Пинг-понг**

Реализовать по методологии TDD (Test-Driven Development) два консольных приложения (клиент и сервер). Клиент высылает серверу сообщение “ping”. Сервер, получив сообщение от клиента, отвечает “pong”. Предоставить возможность протоколирования (в свободной форме) каждого события в программе (отправка запроса, получение ответа от сервера и т.д.) в файле log.txt. Обеспечить возможность сериализации и десериализации сообщений.

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

Что погуглить: сокеты

1. **Текстовый браузер**

Реализовать по методологии TDD (Test-Driven Development) консольное приложение, предоставляющее пользователю возможность просмотра веб-страниц из консоли. Предусмотреть следующие функции работы с веб-страницами:

* Отравка GET-запроса на введённый URL адрес
* Возможность ввода адреса пользователем
* Вывод на экран всего содержимого под тегом <body>

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

Что погуглить: запросы http

1. **Файловый менеджер**

Реализовать по методологии TDD (Test-Driven Development) графическое приложение, предоставляющее пользователю возможность просмотра файлов. Реализовать двухоконный режим просмотра с возможностью изменения пути к каталогу. Предусмотреть следующие функции: переход по двойному щелчку мыши в каталог, изменение фокуса на файле или каталоге по нажатию клавиш вверх-вниз, переход в каталог или открытие файла по нажатию enter, возможность переноса и копированию файла или каталога по нажатию F6 и F5 соответственно.

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Скринер рабочего экрана**

Реализовать по методологии TDD (Test-Driven Development) службу, работающую в фоновом режиме и позволяющее делать скриншоты рабочего стола с настраиваемым периодом времени (задаётся как аргумент командной строки). Обеспечить хранение графических файлов в папке. Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Клавиатурный трекер**

Реализовать по методологии TDD (Test-Driven Development) службу, работающую в фоновом режиме и позволяющее делать “снимки” нажатий на клавиши клавиатуры с настраиваемым периодом времени (задаётся как аргумент командной строки). Обеспечить хранение снимков в виде текстовых файлов в папке. Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Файерволл**

Реализовать по методологии TDD (Test-Driven Development) консольное приложение, предоставляющее пользователю вносить IP адреса или имена узлов в сети. Впоследствии на эти узлы с рабочей машины доступ будет заблокирован. Адреса вносятся пользователем в качестве аргументов командной строки (подсказка С:\\Windows\System32\Drivers\Etc\hosts.txt).

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Хот-кей менеджер**

Реализовать по методологии TDD (Test-Driven Development) службу, позволяющая просмотреть “горячие” клавиши ОС. Предусмотреть графический интерфейс для просмотра. Реализовать назначение новых хот-кеев пользователем. Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Упаковщик архивов для отправки через vk.com**

Реализовать приложение, позволяющее упаковывать файлы в архив (сжатие не является обязательным требованием) с собственным расширением .vk для отправки файлов через файлообменный сервис vk.com. Предусмотреть контроль упаковки каждого файла, каталога и подкаталога (лимит архива 199 Мб). Обеспечить работу с функциями VK API для отправки файла, не заходя в браузерный клиент.

Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Змейка**

Реализовать игру “Змейка”. Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

1. **Контроллер заряда батареи**

Реализовать по методологии TDD (Test-Driven Development) службу, работающую в фоновом режиме и позволяющее делать периодические замеры заряда батареи ноутбука. Служба доступна из трей-панели ОС, при нажатии даётся заряд батареи в процентах, примерное время до разрядки и степень нагрузки на систему (по ОЗУ и ЦП). Обеспечить возможность покрытия 100% кода юнит-тестами. Обработать все возможные исключительные ситуации. По результатам всех тестов сформировать отчёт в файле report.txt следующего формата:

*(+) CHECK TIMEOUT – PASS*

*(-) CHECK FOR RESPONSE 200 – FAIL*

*…*

Рекомендации к выполнению: общая архитектура СУБЪЕКТ-НАБЛЮДАТЕЛЬ.