**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Технологии и методы программирования»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**Выполнили:**

Ахраров Али, студент группы номер N3350

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Ищенко Алексей Петрович

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Введение**

В условиях стремительного развития информационных технологий контроль над использованием программного обеспечения становится всё более актуальным. Ограничение количества запусков и времени использования программ помогает предотвратить злоупотребления, обеспечивает соблюдение лицензионных соглашений и способствует монетизации программ через предложения о приобретении полной версии. В рамках данного проекта была разработана простейшая программа, выполняющая следующие функции:

1. Запрашивает у пользователя ФИО (Фамилия, Имя, Отчество) и сохраняет его в текстовый файл.
2. Проверяет наличие введённого ФИО в файле и уведомляет пользователя, если такое ФИО уже существует.
3. Ограничивает количество запусков и время использования программы.
4. По достижении лимита предлагает пользователю приобрести полную версию или деинсталлировать программу.
5. Сохраняет информацию о предыдущих запусках для предотвращения превышения суммарных лимитов при повторной установке.

**Цель и задачи проекта**

**Цель проекта** заключается в разработке программного обеспечения, которое контролирует использование программы посредством ограничения количества запусков и времени работы, а также обеспечивает механизмы монетизации и защиты от несанкционированного использования.

**Задачи проекта:**

1. Создание интерфейса для ввода ФИО пользователя.
2. Реализация механизма сохранения и проверки ФИО в текстовом файле.
3. Внедрение системы ограничения количества запусков и времени работы программы.
4. Разработка функционала предложения приобретения полной версии или деинсталляции программы при достижении лимитов.
5. Обеспечение сохранения информации о предыдущих запусках для корректного управления лимитами при повторной установке.
6. Защита конфигурационных файлов от несанкционированного доступа и модификации.

**Анализ требований**

Для успешной реализации поставленных задач программа должна обладать следующими характеристиками:

* **Пользовательский ввод:** Возможность ввода ФИО пользователя через консольный интерфейс.
* **Хранение данных:** Сохранение введённых ФИО в текстовом файле с проверкой на дублирование.
* **Ограничения использования:** Контроль количества запусков и времени работы программы с возможностью уведомления пользователя о достигнутых лимитах.
* **Монетизация:** Возможность предложения пользователю приобрести полную версию программы при достижении лимитов.
* **Самоустранение:** Функционал деинсталляции программы по желанию пользователя.
* **Сохранение данных о предыдущих запусках:** Обеспечение сохранности информации о предыдущих запусках для корректного управления лимитами при повторной установке.
* **Защита конфигурационных файлов:** Предотвращение несанкционированного доступа и изменений в файлах конфигурации.

**Архитектура программы**

Программа состоит из нескольких модулей, каждый из которых отвечает за выполнение определённых функций:

1. **Модуль ввода и проверки ФИО:**
   * Запрашивает у пользователя ФИО.
   * Проверяет наличие введённого ФИО в файле users.txt.
   * Добавляет новое ФИО в файл при отсутствии дубликатов.
2. **Модуль управления ограничениями:**
   * Отслеживает количество запусков программы и общее время её работы.
   * Сохраняет данные в файлы конфигурации config.txt и persistent.dat.
   * Проверяет, достигнуты ли лимиты использования.
3. **Модуль взаимодействия с пользователем:**
   * Информирует пользователя о текущих лимитах.
   * Предлагает варианты действий при достижении лимитов (приобрести полную версию или деинсталлировать программу).
4. **Модуль установки и деинсталляции:**
   * Создаёт необходимые файлы и директории при установке.
   * Удаляет файлы программы при деинсталляции, сохраняя при этом информацию о предыдущих запусках.
5. **Модуль защиты файлов:**
   * Обеспечивает защиту конфигурационных файлов от несанкционированного удаления и модификации.

**Выбор технологий и инструментов**

Для реализации программы был выбран язык программирования C++ благодаря его высокой производительности и возможности взаимодействия с системными API Windows. В качестве среды разработки использовался Microsoft Visual Studio, что обеспечило удобство разработки и отладки программы. Для управления файлами и директориями использовались стандартные библиотеки C++ и функции API Windows, такие как CreateDirectoryW, \_wfopen\_s и RemoveDirectoryW.

**Реализация**

**1. Модуль ввода и проверки ФИО**

Программа начинается с запроса у пользователя ФИО через консольный ввод. Введённая строка сохраняется в переменной типа std::wstring. Затем программа открывает файл users.txt для чтения и проверяет, существует ли уже введённое ФИО в списке. Если ФИО отсутствует, оно добавляется в конец файла. Для предотвращения переполнения файла, при добавлении новых записей, количество ФИО ограничивается пятью, удаляя самые старые записи при необходимости.

**2. Модуль управления ограничениями**

Для контроля количества запусков и времени работы программы используются два файла конфигурации:

* config.txt: Хранит количество запусков (runs).
* persistent.dat: Хранит общее количество запусков (totalRuns) и общее время использования (totalTime).

При запуске программы происходит чтение этих файлов. Если файлы отсутствуют, они создаются автоматически. После проверки текущих значений лимитов, программа решает, продолжать работу или предложить пользователю приобрести полную версию или деинсталлировать программу.

**3. Модуль взаимодействия с пользователем**

Если пользователь достиг лимита по количеству запусков или времени использования, программа выводит соответствующее сообщение и предлагает два варианта действий:

1. **Приобрести полную версию:**
   * Сбрасываются лимиты использования.
   * Пользователь информируется о успешном приобретении полной версии.
2. **Деинсталлировать программу:**
   * Удаляются файлы users.txt и config.txt.
   * Удаляется созданная директория программы.
   * Информация о предыдущих запусках сохраняется в persistent.dat для предотвращения повторного использования лимитов при повторной установке.

**4. Модуль установки и деинсталляции**

При первом запуске программы выполняется проверка наличия файла config.txt. Если файл отсутствует, программа считается устанавливаемой впервые и создаёт необходимые директории и файлы (users.txt, config.txt, persistent.dat). При последующих запусках программа уведомляет пользователя о предыдущей установке и корректно обрабатывает лимиты использования.

**5. Модуль защиты файлов**

Для защиты конфигурационных файлов от удаления и модификации используются функции API Windows, такие как SetFileSecurity и SetNamedSecurityInfo. Эти функции устанавливают права доступа к файлам, ограничивая возможность их удаления или изменения только для администратора или определённых пользователей.

**Обеспечение безопасности**

Безопасность программы достигается следующими способами:

* **Ограничение прав доступа:** Конфигурационные файлы защищены от несанкционированного удаления и изменения с помощью системных функций безопасности.
* **Сохранность данных о предыдущих запусках:** Информация о предыдущих запусках сохраняется в persistent.dat, что препятствует обходу ограничений при повторной установке программы.
* **Проверка целостности файлов:** Программа проверяет наличие и корректность файлов конфигурации при каждом запуске, что предотвращает их подделку или повреждение.
* **Изоляция данных:** Использование отдельных файлов для хранения различных данных (ФИО, конфигурации, информация о запусках) обеспечивает дополнительный уровень защиты.

**Тестирование**

Для обеспечения корректной работы программы было проведено комплексное тестирование, включающее следующие этапы:

1. **Функциональные тесты:**
   * Проверка ввода и сохранения ФИО.
   * Проверка обнаружения дублирующихся ФИО.
   * Проверка ограничения количества запусков и времени работы.
2. **Тесты безопасности:**
   * Проверка защиты конфигурационных файлов от удаления и модификации.
   * Проверка сохранности информации о предыдущих запусках при деинсталляции и повторной установке.
3. **Тесты устойчивости:**
   * Проверка работы программы при отсутствии или повреждении файлов конфигурации.
   * Проверка корректности обработки ошибок при невозможности доступа к файлам.
4. **Нагрузочные тесты:**
   * Оценка производительности программы при быстром последовательном запуске до достижения лимитов.
5. **Тесты совместимости:**
   * Проверка работы программы на различных версиях операционной системы Windows.
   * Оценка возможности адаптации программы для других операционных систем при необходимости.

**Результаты**

Разработанная программа успешно выполняет все поставленные задачи:

* **Ввод и сохранение ФИО:** Пользователь может вводить своё ФИО, которое сохраняется в файл users.txt. При повторном вводе того же ФИО программа уведомляет пользователя о наличии записи.
* **Ограничение использования:** Программа корректно отслеживает количество запусков и время работы. При достижении лимитов пользователь получает соответствующее уведомление и варианты действий.
* **Монетизация и деинсталляция:** Пользователь может выбрать приобретение полной версии программы, что сбрасывает лимиты, или деинсталлировать программу, удаляя её файлы с сохранением информации о предыдущих запусках.
* **Защита данных:** Конфигурационные файлы защищены от несанкционированного доступа и изменений, что обеспечивает надежность системы ограничений.
* **Сохранность информации при повторной установке:** При повторной установке программа учитывает предыдущие запуски и не позволяет превышать суммарные лимиты использования.

**Заключение**

В рамках данного проекта была разработана простейшая, но функциональная программа, обеспечивающая контроль над использованием программного обеспечения посредством ограничения количества запусков и времени работы. Реализованные механизмы позволяют эффективно управлять использованием программы, предотвращая злоупотребления и обеспечивая возможности монетизации через предложения о приобретении полной версии.

**Основные достижения проекта:**

* Успешная реализация функционала ввода и сохранения ФИО пользователя.
* Надёжная система ограничения использования программы по количеству запусков и времени работы.
* Эффективные механизмы защиты конфигурационных файлов и сохранности информации о предыдущих запусках.
* Интуитивно понятный интерфейс взаимодействия с пользователем при достижении лимитов.

**Перспективы развития:**

* Расширение функционала программы для поддержки дополнительных типов ограничений.
* Разработка графического интерфейса для улучшения взаимодействия с пользователем.
* Интеграция системы онлайн-активации и лицензирования для повышения уровня защиты и удобства приобретения полной версии.
* Адаптация программы для работы на других операционных системах, таких как Linux или macOS.

**Приложения**

**Пример файла users.txt**

Иванов Иван Иванович

Петров Пётр Петрович

Сидоров Сидор Сидорович

*Файл содержит список ФИО пользователей, каждый записан с новой строки.*

**Пример файла config.txt**

3

*Файл хранит количество запусков программы.*

**Пример файла persistent.dat**

4

120

*Первое число — общее количество запусков программы. Второе число — общее время использования программы в секундах.*