Министерство образования и науки Российской Федерации

**Саратовский государственный технический университет**

**имени Гагарина Ю.А.**

**Инструментальные программные средства обфускации исходных текстов программ**

Методические указания

к выполнению практической работы

по курсу «Инструментальные средства информационных систем»

для студентов направления

«Информационные системы и технологии»

Выполнил Крюков Александр Б2-ИФСТ-32

Саратов 2022

# Инструментальные программные средства обфускации исходных текстов программ

**1 часть: Анализ существующих инструментальных средств обфускации исходного текста программ.**

1. ProGuard

* ProGuard — это jar-файл, который переставляет символы в нашем коде, оптимизирует его и, вроде как, повышает безопасность. ProGuard – инструмент командной строки, который уменьшает размер приложения за счет сокращения байт-кода и запутывания имён классов, полей, методов. ProGuard помогает сжать любое приложение Java или Kotlin, будь то на мобильном, встроенном или настольном компьютере. Делает приложения на 90% меньше и на 20% быстрее, обеспечивая при этом базовую защиту от статического обратного проектирования с помощью [обфускации кода.](https://www.guardsquare.com/what-is-code-obfuscation) Сообщество с открытым исходным кодом
  + На этапе сжатия обнаруживаются и удаляются неиспользуемые классы, поля, методы и атрибуты.
  + Шаг оптимизатора оптимизирует байт-код и удаляет неиспользуемые инструкции.
  + На этапе запутывания остальные классы, поля и методы переименовываются, используя короткие бессмысленные имена.
  + На последнем этапе предварительной проверки к классам добавляется информация о предварительной проверке, необходимая для Java Micro Edition, а также для Java 6 и более поздних версий.
* Бесплатно

1. Javascript-obfuscator (пакет)
   * JavaScript Obfuscator — это мощный и бесплатный обфускатор для JavaScript и Node.js, содержащий большое количество функций и возможностей для защиты исходного кода. Доступен как в виде локального инструмента с открытым исходным кодом, так и в виде онлайн сервиса. Посредством ряда преобразований, таких как переименование переменных/функций/аргументов, удаление строк и другие, ваш исходный код преобразуется во что-то нечитаемое, но работает точно так же, как и раньше.
   * Имеет большое разнообразие настроек для обфускации и минимизации. Есть возможность вставить js код или загрузить файл. Так же можно удобно скачать обфусцированный файл.
   * Бесплатно
   * Npm пакет

Проанализировав варианты, я выбрал Obfuscator.io, потому что поддерживает js файлы, очень мощное средство, которое позволяет производить обфускацию в веб-приложении, наличие большого количества настроек и бесплатное использование.

Требования к входным данным:

* Файл, написанный на Javascript

Требования к выходным данным:

* Абсолютно нечитаемый минимизированный код

Рисунок 1. Html разметка приложения.



Рисунок 2. Исходный js файл.

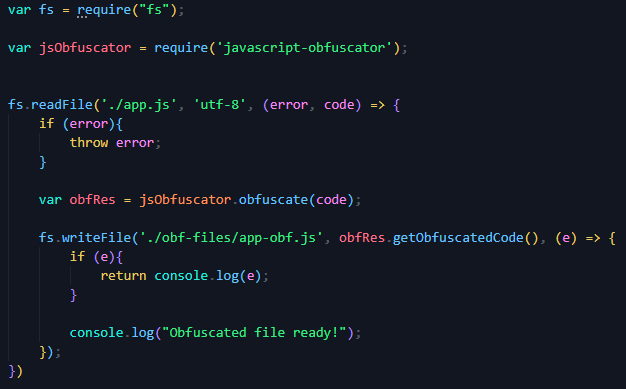
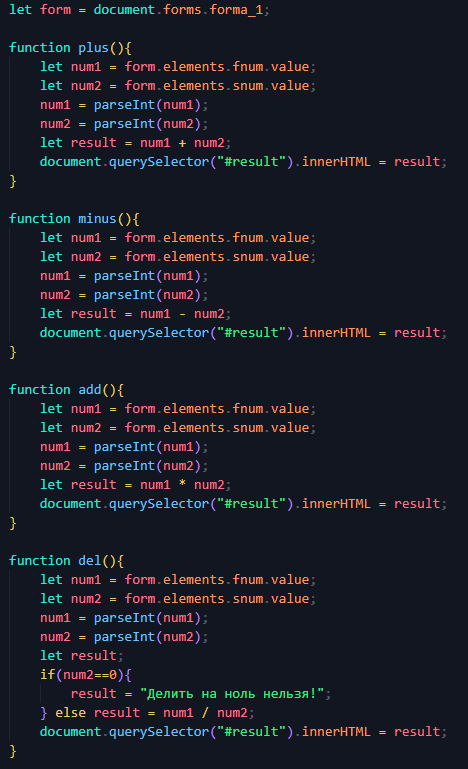


Рисунок 3. Код программы для обфускации.

Выходные параметры:

Рисунок 4. Создание обфусцированного файла.



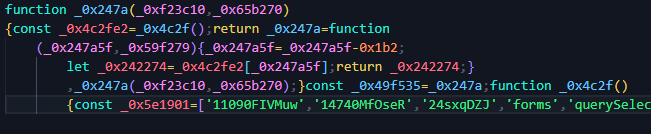


Рисунок 5. Полученный обфусцированный код.



Рисунок 5. Подключение обфусцированного кода.

Вывод: на выходе мы получаем обфусцированный и минимизированный код, в котором имена переменных функций изменены в 16-ую систему счисления, добавлены разного рода запутывающие (ненужные) функции, такие как parseInt, добавлены избыточные блоки try catch. Причем после обфускации код работает исправно, что значит, что он эквивалентен исходному.

**2 часть. Практическая реализация элементарного обфускационного инструментального средства.**

Требования к входным данным:

* Файл, написанный на Javascript

Требования к выходным данным:

* Усложненный код

Рисунок 6. Html разметка приложения.



Рисунок 7. Исходный js файл.

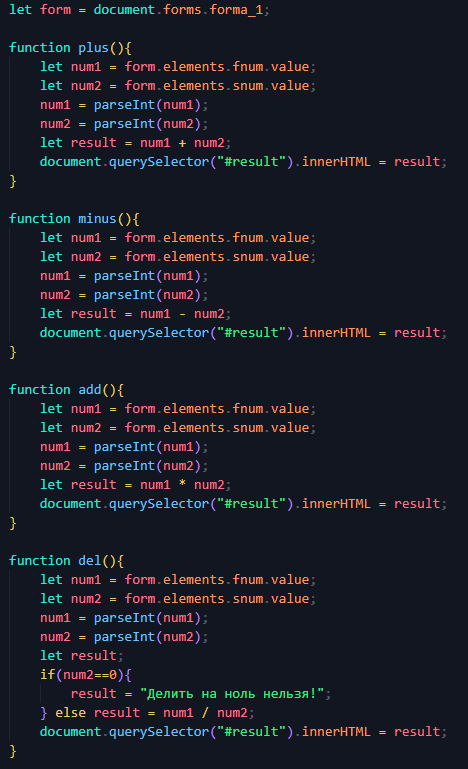


Рисунок 8. Код программы для обфускации:

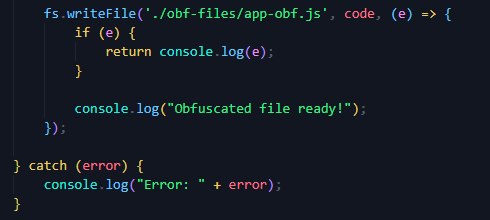
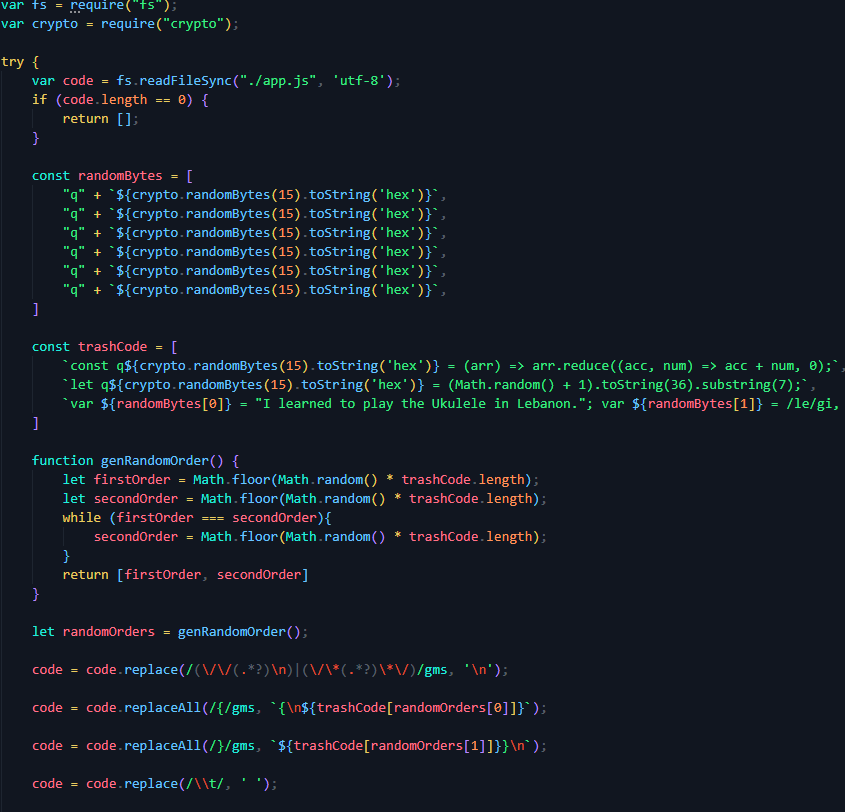


Рисунок 9. Создание обфусцированного файла.



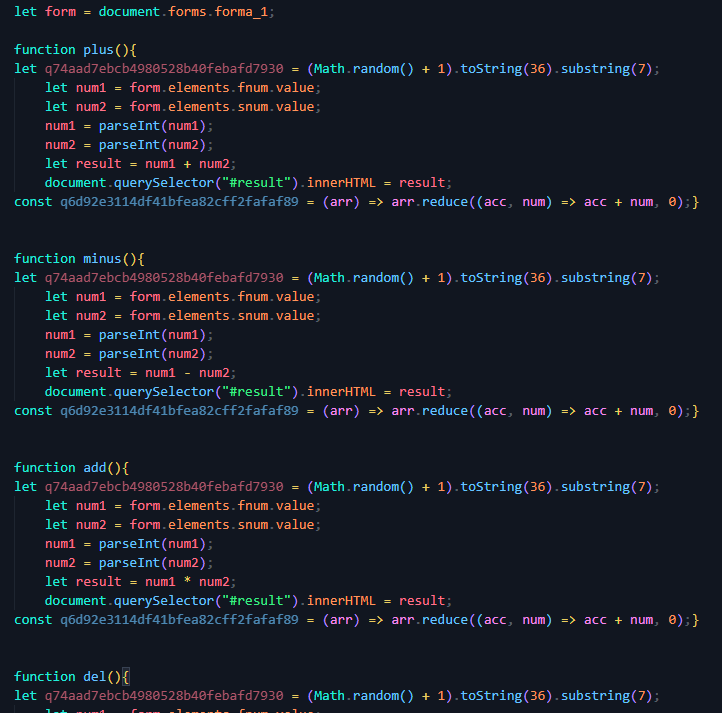


Рисунок 10. Полученный обфусцированный код.



Рисунок 11. Подключение обфусцированного кода.

Вывод: на выходе мы получаем обфусцированный и минимизированный код, в котором удалены все комментарии, добавлены запутывающие функции с рандомными переменными, а так же удалены символы. Причем после обфускации код работает исправно, что значит, что он эквивалентен исходному.