**Обфуска́ция** или **запутывание кода** — приведение исходного текста или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.

**Обфускатор** - программы, производящие обфускацию.

* Затруднение декомпиляции/отладки и изучения программ с целью обнаружения функциональности.
* Затруднение декомпиляции проприетарных программ с целью предотвращения обратной разработки или обхода DRM и систем проверки лицензий.
* [Оптимизация](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) программы с целью уменьшения размера работающего [кода](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) и (если используется [некомпилируемый](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F" \o "Компилируемый язык программирования) язык) ускорения работы.
* Демонстрация неочевидных возможностей языка и квалификации программиста (если производится вручную, а не инструментальными средствами).

**Оптимизация**

* В интерпретируемых языках обфусцированный код занимает меньше места, чем исходный, и зачастую выполняется быстрее, чем исходный. Современные обфускаторы также заменяют константы числами, оптимизируют код инициализации массивов, и выполняют другую оптимизацию, которую на уровне исходного текста провести проблематично или невозможно.
* Проблема уменьшения размера важна, например, при программировании для сотовых телефонов на J2ME, где размер программы серьёзно ограничен. Обфускация JavaScript уменьшает размер HTML-файлов и, соответственно, ускоряет загрузку.

**Недостатки**

**Потеря гибкости кода**

* Код после обфускации может стать более зависимым от платформы или компилятора.

**Трудности**

* Обфускатор не даёт постороннему выяснить, что делает код, но и не даёт разработчику отлаживать его. При отладке приходится отключать обфускатор.

**Недостаточная безопасность**

* Хотя обфускация помогает сделать распределённую систему более безопасной, не стоит ограничиваться *только* ею. Обфускация — это безопасность через неясность. Ни один из существующих обфускаторов не гарантирует сложности декомпиляции и не обеспечивает безопасности на уровне современных криптографических схем. Вполне вероятно, что эффективная защита невозможна (по крайней мере в некотором конкретном классе решаемых задач).

**Ошибки в обфускаторах**

* Современный обфускатор — сложный программный комплекс. Зачастую в обфускаторы, несмотря на тщательное проектирование и тестирование, вкрадываются ошибки. Так что есть ненулевая вероятность, что прошедший через обфускатор код вообще не будет работать или будет работать не так , или так что лучше бы не работал. И чем сложнее разрабатываемая программа, тем больше эта вероятность.

**Вызов класса по имени**

* Большинство языков с промежуточным кодом могут создавать или вызывать объекты по именам их классов. Современные обфускаторы позволяют сохранить указанные классы от переименования, однако подобные ограничения сокращают гибкость программ.