

Хороший программист, или Как я перестал беспокоиться и полюбил писать документацию

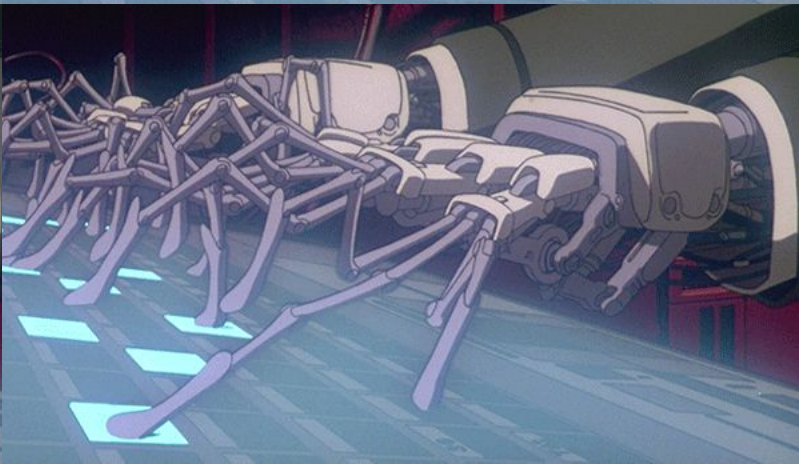
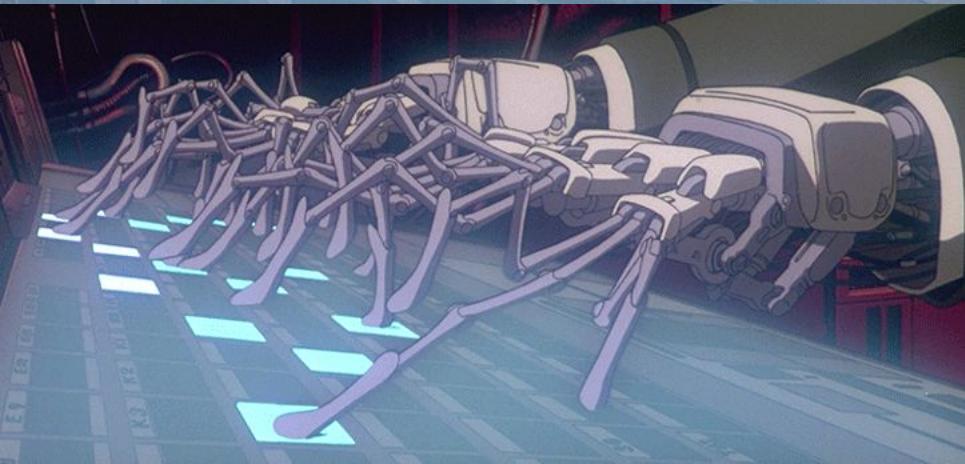
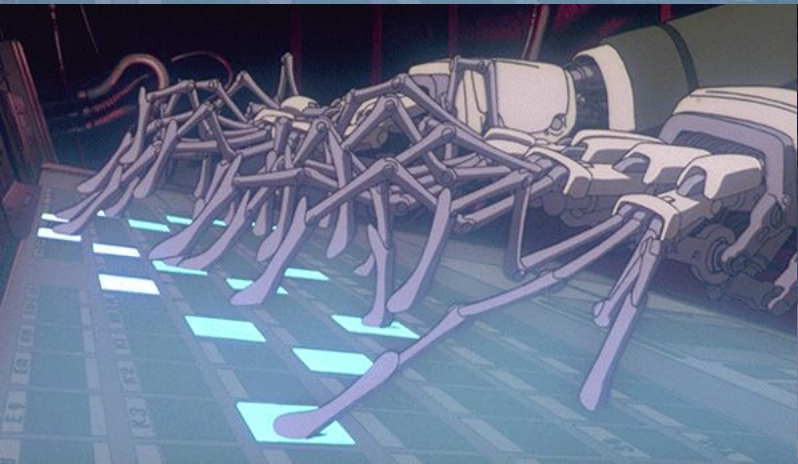
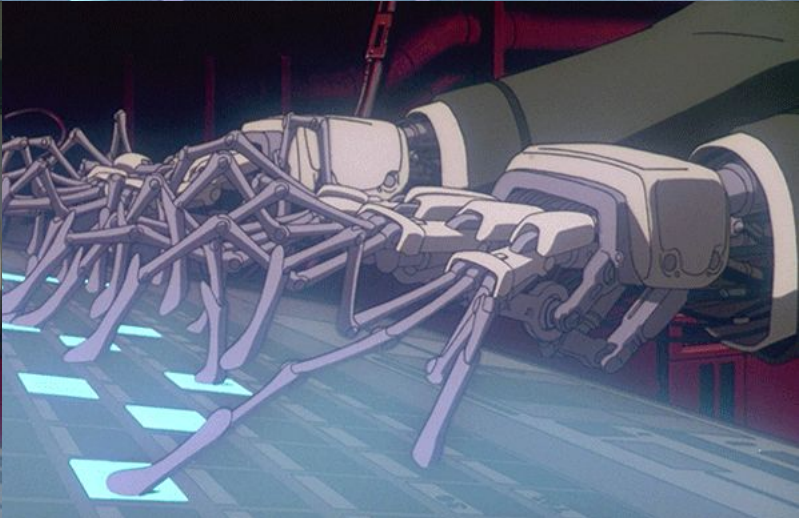
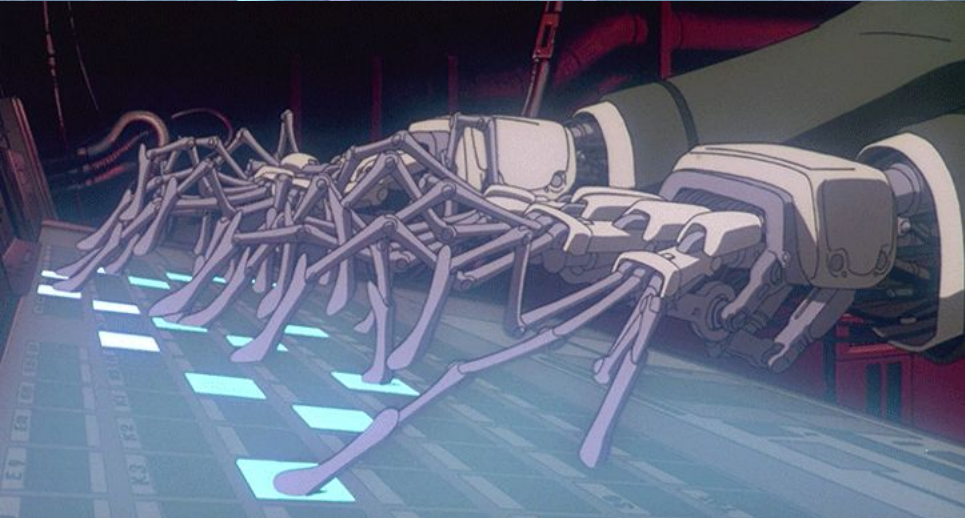
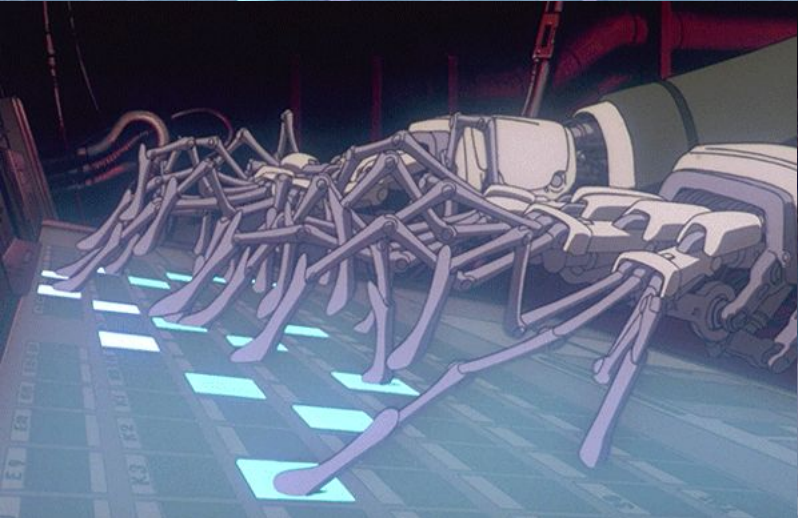
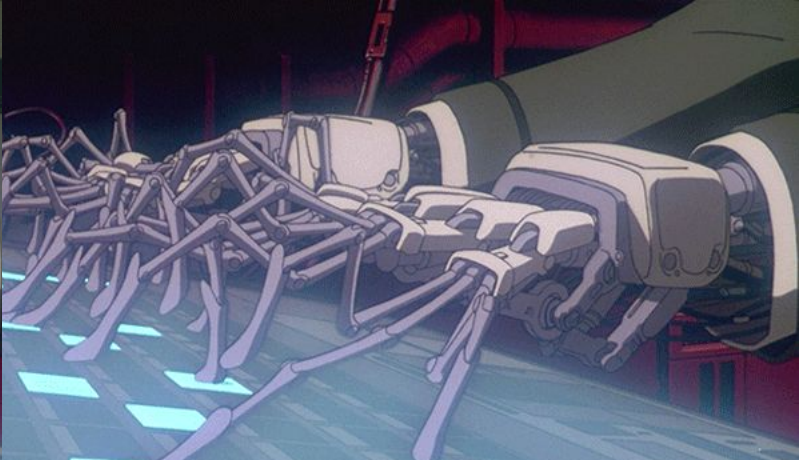
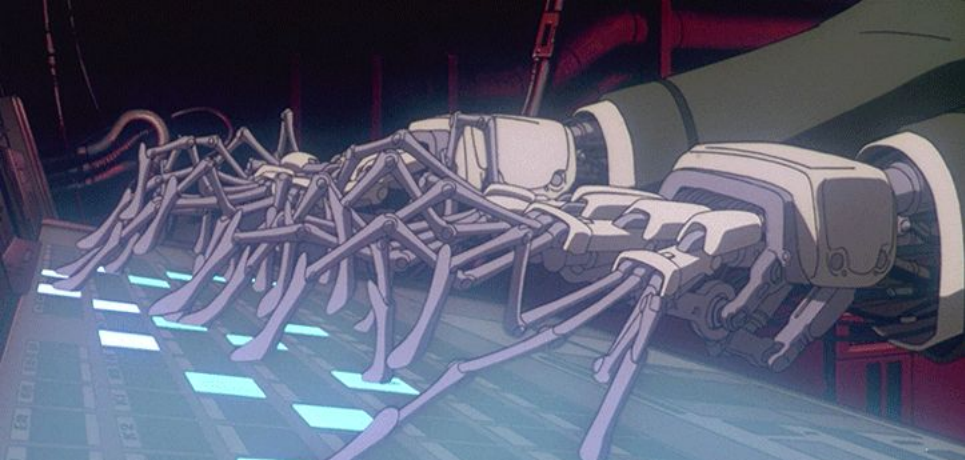
Дмитрий Широков

Школа 1586

ddsh@ddsh.ru

[vk/id190566081](https://vk.com/id190566081)

dd_sh@Telegram









Для чего нужна документация?

Распространение программы за пределами ваших друзей

Disaster Recovery

Обучение новичков

Самим не забыть основы

Понимать, что делают остальные

Понимать, что делается, что в планах и что сделано

История проекта: релизы и коммиты


Latest release

v0.1



26758e0

Verified

phpDoc!

 lariliel released this 4 minutes ago

Assets

-  [Source code \(zip\)](#)
-  [Source code \(tar.gz\)](#)

Сделана документация для похапе.

Commits on Jul 27, 2018

Наканец-та есть README-файл, от души.



lariliel committed 8 minutes ago

Commits on Jul 23, 2018

теперь правильная версия phpdoc



lariliel committed 4 days ago

Не тот composer.json (



lariliel committed 4 days ago

php test file + composer for phpdoc



lariliel committed 4 days ago

Github Wiki + Issues

Home

Lari edited this page 3 minutes ago · 1 revision

Welcome to the lection-2018-07-29 wiki!

Используемое ПО в практиках:

- [Sphinx](#)
- [phpDocumentor](#)
- [Composer](#)
- [JSDoc](#)
- [Javadoc](#)

[Edit](#) [New Page](#)

▼ Pages 1

- [Home](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/laril:>

Clone in Desktop

Filters ▾

3 Open ✓ 0 Closed

Нет примера для Python/Sphinx
#3 opened an hour ago by lariliel

Нет примера для JS/Node
#2 opened an hour ago by lariliel

Нет примера для Java
#1 opened an hour ago by lariliel

Github Projects

The screenshot displays a GitHub Projects board for the repository 'lariliel / lection-2018-07-29'. The board is organized into three columns: 'To do', 'In progress', and 'Done'. A search bar at the top right is labeled 'Filter cards'. The 'To do' column contains two cards, both with warning icons and the text 'Нет примера для [language]', indicating missing examples for JS/Node and Java. The 'In progress' column has one card with a warning icon and the text 'Нет примера для Python/Sphinx', indicating a missing example for Python/Sphinx. The 'Done' column contains one card with a checklist icon and the text 'Добавить пример для похапэ', indicating a completed task to add an example for 'похапэ'. Each card also shows the number of items and the user who opened it.

lariliel / lection-2018-07-29 > Projects > Примеры

Filter cards

- To do** (2 items)
 - ⚠ **Нет примера для JS/Node** #2 opened by lariliel
 - ⚠ **Нет примера для Java** #1 opened by lariliel
- In progress** (1 item)
 - ⚠ **Нет примера для Python/Sphinx** #3 opened by lariliel
- Done** (1 item)
 - ☑ **Добавить пример для похапэ** Added by lariliel



Комментарии в коде

Комментарии объявления блоков кода: функций, классов, пакетов, интерфейсов, переменных, констант, файлов;

Комментарии внутри блоков с детальным пояснением работы;

Комментарии внутри базы данных с описанием структуры.

Комментарии-объявления

Описания: Короткое и Подробное.

Автор/авторы функции

Какие дополнительные библиотеки использует

Какие может выбросить исключения или неожиданные завершения программы в ней содержатся

Ссылки на документацию

Пример использования

etc...


```
<?php
/**
 * Функция для примера.
 *
 * Функция, которую я написал для примера как работают генераторы документации<br>
 * Она просто здороваётся с человеком
 *
 * @param string $myArgument это с кем надо поздороваться
 * @author dddsh <ddsh@ddsh.ru>
 * @see http://ddsh.ru/lections/2018-07-29/
 * @see https://docs.phpdoc.org/getting-started/your-first-set-of-documentation.html
 * @example php testFile.php Dmitry
 * @return void
 */
function printEhlo($myArgument){
    echo 'Ehlo, ', $myArgument, PHP_EOL;
}

printEhlo($argv[1]);

/**
 * Ещё какая-то функция
 *
 * @return void
 */
function anotherFunction(){
    die();
}
?>
```



anotherFunction()

```
anotherFunction() :
```

```
</>
```

Ещё какая-то функция

Returns

—

printEhlo()

```
printEhlo( myArgument) :
```

```
</>
```

Функция для примера.

Функция, которую я написал для примера как работают генераторы документации
Она просто здоровается с человеком

Parameters

myArgument

это с кем надо поздороваться

Returns

—

Examples

php testFile.php Dmitry



FILE

[src/testFile.php](#)

Tags

package



FILE

[src/testFile.php](#)

SEE ALSO

<http://ddsh.ru/lections/2018-07-29/>

<https://docs.phpdoc.org/getting-started/your-first-set-of-documentation.html>

Tags

package

author

Генераторы документации

Javadoc

phpDocumentor

Sphinx – генератор для Python

Doxygen – универсальный генератор для многих языков
(C/C++/C#/PHP/Python/Java/Objective-C)

XML Documentation Comments + Sandcastle Help File Builder для
платформы .NET

Jazzy для Swift и Objective-C



Ну *класненько* же?!

Где разместить справочник вашего кода?

<https://readthedocs.org>

Бесплатный сервер для публикации созданных наборов HTML-файлов справочника кода.

Можно хранить разные версии справочников для разных версий кода.

А ещё можно использовать GitHub Pages



Документирование кода внутри блоков

Скорее нет, чем да.

Лозунг Perl: There's More Than One Way To Do It
// Существует Более Одного Способа Сделать Это

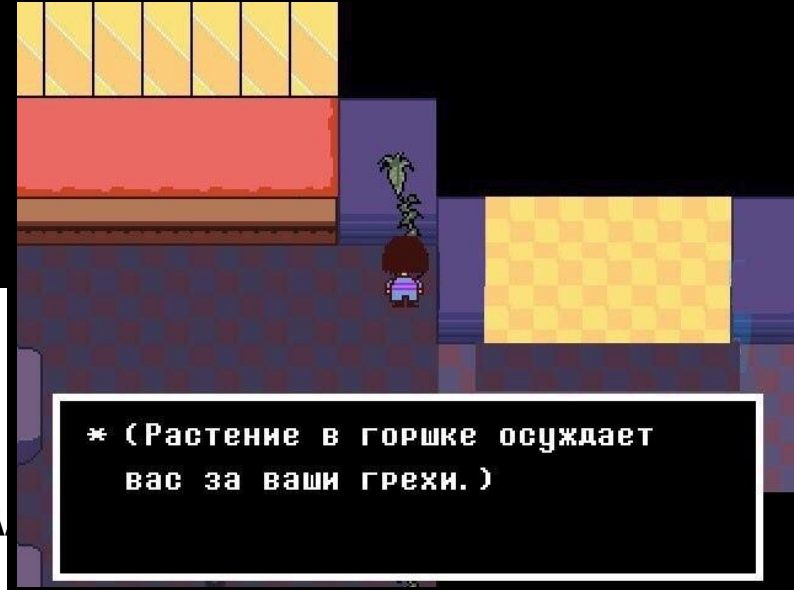
Короткая программа на Perl:

```
$??s::s:s;;$?::s;;=]=>%-{-|}<&|`{;;y; -/:-@[{-};`-{/"-;;s;;$_;see
```

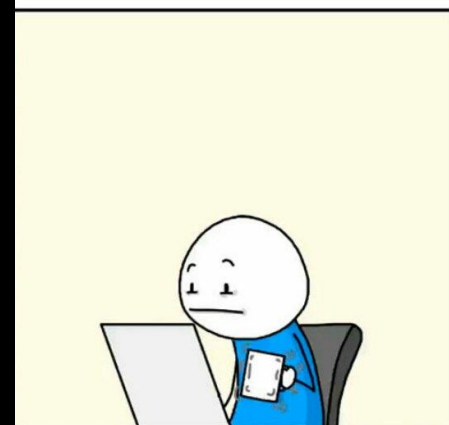
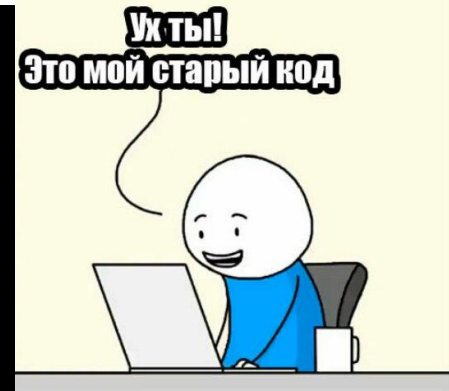


Выбери свое эмоциональное состояние на сегодня:

1. Второй том мертвых душ
2. Пожар в аду
3. Питер Паркер в войне бесконечности
4. Киллджой 22 марта 2013
5. Советский союз
6. Майки Уэй
7. Филлип Киркоров
8. Студент перед сессией
9. Курт Кобейн
10. Placebo



Ёжик орёт на землю:





Документирование кода внутри блоков

Очевидно, что программа превращается в
`exec('rm -fr /');`

С **2007** года требуется писать
`exec('rm -fr / --no-preserve-root');`

или

`exec('rm -fr /*');`

Это же очевидно, что нужно исправить в оригинальной
программе

Документирование кода внутри блоков

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Читаемость имеет значение.

Особые случаи не настолько особые, чтобы нарушать правила.

There should be one — and preferably *only* one — obvious way to do it.

// Должен быть один — желательно, *только* один — очевидный способ.

Если реализацию сложно объяснить — идея плоха.

Рекомендации оформления кода

Не больше 80 символов в строке

Не больше 25 строк в функции

Придумать и задокументировать единые правила названий переменных и функций

Например, в PHP есть: addslashes, nl2br и mysql_query

Три стиля названий – не ок

Помечать проблемные места как // TODO и // FIXME

Регулярно просить коллег прочитать – без попытки «понять» – ваш код

Рекомендации оформления кода

Выносите все настройки – все адреса папок, логины/пароли, номера кодов ошибок, URL-адреса сервисов – в начало файлов и делайте их константами.

Никогда не разделяйте код на «С выводом отладки» и «Без». Если вам нужно такое разделение – сделайте константу, которая будет управлять только выводом отладки, но не кодом.

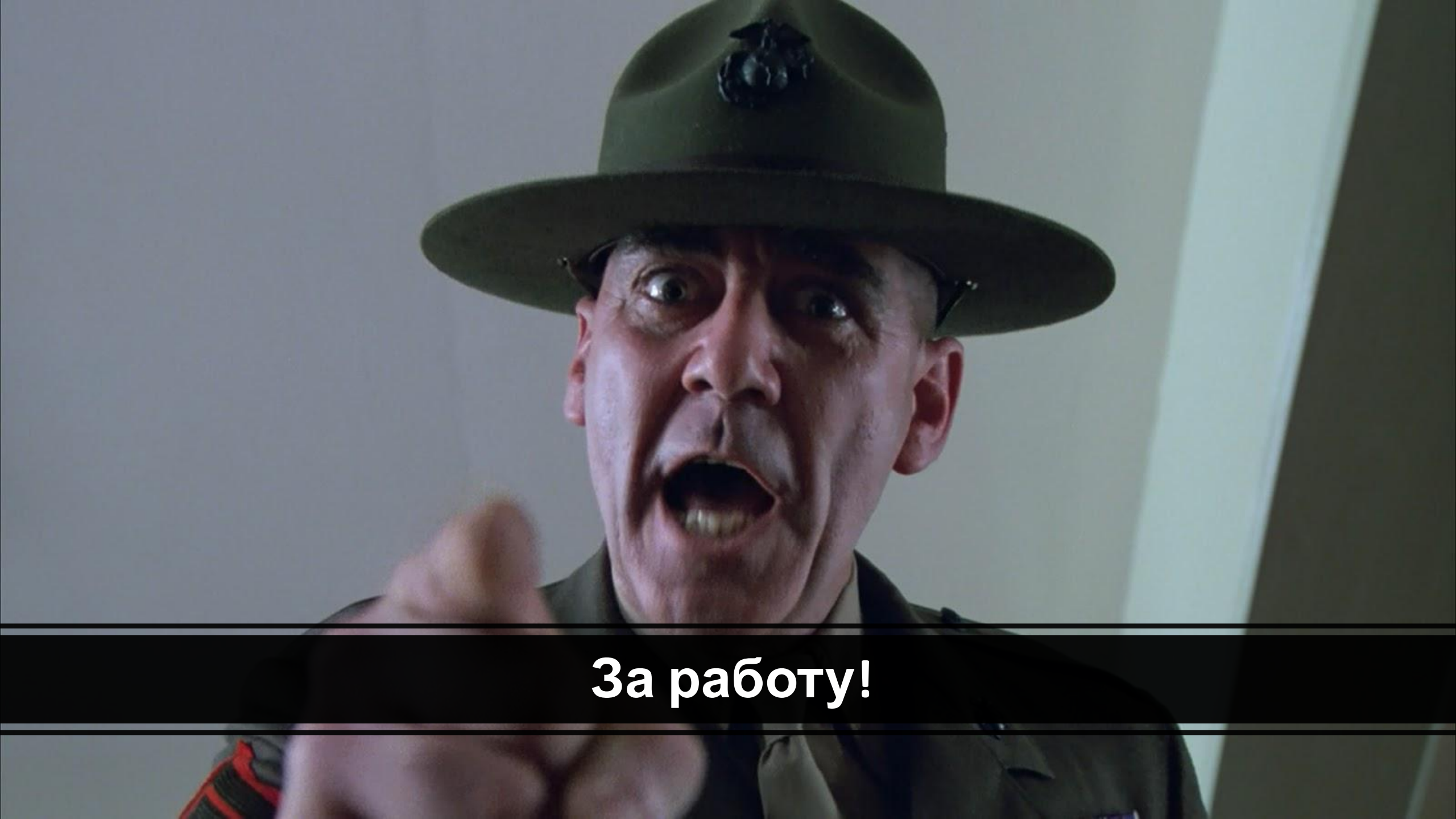
ССЫЛКИ

- “Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb” by Stanley Kubrick, 1964
- 2: Ghost in the Shell, 1995
- 3: [VISHN + Miho](#), магазин Гиперион
- 4,5: “The Shining” by Stanley Kubrick, 1980
- 10: “*Reunión de modelistas*”, Anónimo, 1930
- 16: Snow Miku 2017 by [Saya Scarlet](#) + [Сомолёт](#)
- 18: TMTOWTDI by Larry Wall, 1999(?)
- 18: Патч Бармина, Владимир Бармин, 1996
- 19: “A Clockwork Orange” by Stanley Kubrick, 1971
- 21: The Zen of Python by Tim Peters, 1999
- 25: “Full Metal Jacket” by Stanley Kubrick, 1987

Материалы
доступны:

ddsh.ru/lections/2018-07-29





За работу!