



# **Чистый код**

## **Приемы написания красивого и понятного кода**

**Алексей Аверьянов**



# Немного обо мне




- В 1С с 2003 года. Со времен 7.7
- Работал в франчайзи и в компаниях-пользователях 1С, на маленьких внедрениях и в федеральных компаниях.
- Прошел через разные роли: внедренец, программист, аналитик, руководитель проектов, тимлид
- Сейчас разработчик 1С в компании Автомакон (г.Москва) на проекте "ВкусВилл": сеть магазинов здорового питания, 1300 магазинов в разных городах и странах.
- Ведущий консультант в компании Автоматизация (г.Омск)
- Живу в Омске.

# О чем доклад?



# Книги





**Для кого мы пишем  
код?**



# Что делает этот код?

```
// Плохой пример
```

```
функция функцияХ(Парам1, Парам2)
    X = Новый Структура("Эл1, Эл2, Эл3", Истина, Неопределено, "
    Таб = функция2(Парам1, Парам2);
    м = Новый Массив;
    Для Каждого стр Из Таб Цикл
        А = Новый Структура;
        А.Вставить("product_id", стр.a);
        А.Вставить("cell", СокрЛП(стр.b));
        А.Вставить("rack", СокрЛП(стр.c));
        м.Добавить(А);
    КонецЦикла;
    X.Эл2 = м; Возврат X;
Конецфункции
```

# А теперь? Что делает этот код?

☐ функция СформироватьОтветПоТовару(НомерТочки, КодТовара)

```
Результат = Новый Структура("Успешно, Ответ, ТекстОшибки",  
Истина, Неопределено, "");
```

```
ТаблицаЯчеек = ПолучитьЯчейкиРазмещенияПоТовару(НомерТочки, КодТовара);
```

```
Ответ = Новый Массив;
```

```
Для Каждого СтрокаЯчейки Из ТаблицаЯчеек Цикл
```

```
    Элемент = Новый Структура;
```

```
    Элемент.Вставить("product_id", СтрокаЯчейки.id_tov);
```

```
    Элемент.Вставить("cell", СокрЛП(СтрокаЯчейки.Ячейка));
```

```
    Элемент.Вставить("rack", СокрЛП(СтрокаЯчейки.Стеллаж));
```

```
    Ответ.Добавить(Элемент);
```

```
КонецЦикла;
```

```
Результат.Ответ = Ответ;
```

```
Возврат Результат;
```

☐ Конецфункции




# Что такое чистый код?

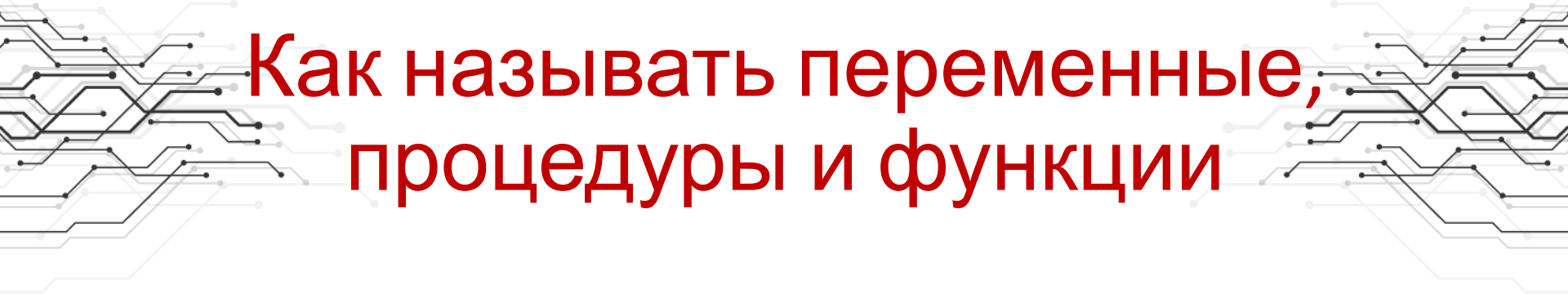


- Элегантный, понятный
- Приятно читать
- Хорошо решает одну задачу
- Не содержит дублирования
- Четко выражает все концепции архитектуры
- Предсказуемый
- Выглядит так, словно его автор над ним хорошо потрудился





# Приемы написания ЧИСТОГО И ПОНЯТНОГО кода



# Как называть переменные, процедуры и функции

Имена должны передавать намерения программиста

Имя переменной, процедуры или функции должно отвечать на все вопросы:

1. Почему эта переменная существует
2. Что она делает
3. Как она используется

# Как называть переменные, процедуры и функции?

// Плохой пример

```
☐ функция функцияХ(Парам1, Парам2)
    X = Новый Структура("Эл1, Эл2, Эл3", Истина, Неопределено,
    Таб = функция2(Парам1, Парам2);
    м = Новый Массив;
    Для Каждого стр Из Таб Цикл
        А = Новый Структура;
        А.Вставить("product_id", стр.a);
        А.Вставить("cell", СокрЛП(стр.b));
        А.Вставить("rack", СокрЛП(стр.c));
        м.Добавить(А);
    КонецЦикла;
    X.Эл2 = м; Возврат X;
Конецфункции
```

# Как называть переменные, процедуры и функции

Функция СформироватьОтветПоТовару(НомерТочки, КодТовара)

```
Результат = Новый Структура("Успешно, Ответ, ТекстОшибки",  
Истина, Неопределено, "");
```

```
ТаблицаЯчеек = ПолучитьЯчейкиРазмещенияПоТовару(НомерТочки, КодТовара);
```

```
Ответ = Новый Массив;
```

```
Для Каждого СтрокаЯчейки Из ТаблицаЯчеек Цикл
```

```
    Элемент = Новый Структура;
```

```
    Элемент.Вставить("product_id", СтрокаЯчейки.id_tov);
```

```
    Элемент.Вставить("cell", СокрЛП(СтрокаЯчейки.Ячейка));
```

```
    Элемент.Вставить("rack", СокрЛП(СтрокаЯчейки.Стеллаж));
```

```
    Ответ.Добавить(Элемент);
```

```
КонецЦикла;
```

```
Результат.Ответ = Ответ;
```

```
Возврат Результат;
```

Конецфункции



# 3 принципа сразу



1. Компактные процедуры и функции
2. Один уровень абстракции на одну процедуру и функцию
3. Исключение дублирования

# 3 принципа

```
□ Процедура ПостроитьДомСУчастком()  
    СоставитьОбщийПланУчастка();  
    ПостроитьДом();  
    ОблагородитьУчасток();  
КонецПроцедуры
```

```
□ Процедура ПостроитьДом()  
  
    ПодготовитьПроектДома();  
    ПоэтажныеПланы = РазбитьПроектНаПоэтажныеПланы();  
  
    Для каждого ПланЭтажа Из ПоэтажныеПланы Цикл  
        ВыполнитьСтроительствоЭтажа(ПланЭтажа);  
    КонецЦикла  
  
КонецПроцедуры
```

---



# 3 принципа

функция orderGET(Запрос)

```
ИмяМетода = "order";
```

Попытка

```
НомерТочки = ПреобразоватьПараметрВЦелоеЧисло(Запрос.ПараметрыURL["shop_id"]);
```

```
МассивНомераЗаказов = ПреобразоватьПараметрВМассивЦелыхЧисел(Запрос.ПараметрыURL["order_id"]);
```

```
РежимОтладки = ЗапросВРежимеОтладки(Запрос.ПараметрыЗапроса.Получить("debug"));
```

```
МенеджерЗаписиЛога = РегистрыСведений.ЛогированиеЗапросов.ЗаписатьЛогНачало(Запрос,  
НомерТочки, ИмяМетода, РежимОтладки);
```

```
Результат = СформироватьОтветПоЗаказам(НомерТочки, МассивНомераЗаказов);
```

```
Возврат СформироватьНТТРОтвет(Результат, МенеджерЗаписиЛога);
```

Исключение

```
ЗафиксироватьОшибкуВызватьИсключение(ОписаниеОшибки(), ИмяМетода, МенеджерЗаписиЛога);
```

КонецПопытки;

Конецфункции

# 3 принципа

функция orderGET (Запрос)

```
ИмяМетода = "order";
Попытка

// shop_id
Попытка
    НомерТочки = Число(Запрос.ПараметрыURL["shop_id"]);
Исключение
    ВызватьИсключение "Bad input parameter: " + Запрос.ПараметрыURL["shop_id"];
КонецПопытки;

// число должно быть целым
Если Цел(НомерТочки) <> НомерТочки Тогда
    ВызватьИсключение "Bad input parameter: " + Запрос.ПараметрыURL["shop_id"];
КонецЕсли;

// теперь order_id
МассивНомераЗаказов = СтрРазделить(Запрос.ПараметрыURL["order_id"], ",", Ложь);
Если МассивНомераЗаказов.Количество() = 0 Тогда
    ВызватьИсключение "Bad input parameter: " + Запрос.ПараметрыURL["order_id"];
КонецЕсли;


Для Индекс = 0 По МассивНомераЗаказов.Количество()-1 Цикл
    Попытка
        МассивНомераЗаказов[Индекс] = Число(МассивНомераЗаказов[Индекс] );
    Исключение
        ВызватьИсключение "Bad input parameter: " + МассивНомераЗаказов[Индекс] ;
    КонецПопытки;

    // число должно быть целым
    Если Цел(МассивНомераЗаказов[Индекс] ) <> МассивНомераЗаказов[Индекс] Тогда
        ВызватьИсключение "Bad input parameter: " + МассивНомераЗаказов[Индекс] ;
    КонецЕсли;

    Если МассивНомераЗаказов[Индекс] = Неопределено Тогда
        ВызватьИсключение "Bad input parameter: " + Запрос.ПараметрыURL["order_id"];
    КонецЕсли;
КонецЦикла;

// .....
```

# Магические числа



15	♦	1 000 000
14	♦	500 000
13	♦	250 000
12	♦	125 000
11	♦	64 000
10	♦	32 000
9	♦	16 000
8	♦	8 000
7	♦	4 000
6	♦	2 000
5	♦	1 000
4	♦	500
3	♦	300
2	♦	200
1	♦	100

Что означает второй параметр?  
ПодписатьФоновымЗаданием(ТипыОбрабатываемыхДокументов, 86400)

?????

?????

?????

?????

# Магические числа

```
// 1
Запрос.УстановитьПараметр ("Контрагент",
    Справочники.Контрагенты.НайтиПоКоду ("0002123"));

|
// 2
ОбработатьДокументы (ДокументСсылка, 86400);

// 3
ОбработатьДокументы (ДокументСсылка, 259200);
```


# Магические числа

```
// 1
ОсновнойПоставщикМагазина = Справочники.Контрагенты.НайтиПоКоду ("0002123");
Запрос.УстановитьПараметр ("Контрагент", ОсновнойПоставщикМагазина);


// 2
СуткиВСекундах = 24*60*60;
ОбработатьДокументы (ДокументСсылка, СуткиВСекундах);

// 3
ТроеСутокВСекундах = 3*24*60*60;
ОбработатьДокументы (ДокументСсылка, ТроеСутокВСекундах);
```





# Хорошие и плохие комментарии



1. Лучший комментарий – который вы не написали
2. Комментарий – это всегда костыль нашему неумению более выразительно выразить наше намерение
3. Частая причина написания комментария – сложный код
4. Бывают полезные комментарии (не часто)
5. Закомментированный код...



# Хорошие и плохие комментарии


```
// Проверить, положена ли работнику премия  
Если Сотрудник.ТипЗанятности = Перечисления.ТипыЗанятостиСотрудников.ПолнаяЗанятность  
    И Сотрудник.Возраст > 65 Тогда
```

```
...  
КонецЕсли
```


```
// -----
```

```
Если РаботникуПоложенаПремия(Сотрудник) Тогда
```

```
....  
КонецЕсли
```



# Форматирование и оформление кода



Смотрите как типовых:

1. Отделяют мысли пустыми строками
2. Называют имена в CamelCase
3. Делают отступы Если, Пока, Попытка...
4. Объединяют функции по смыслу в Области

Делайте также.

# Как написать хороший ЧИСТЫЙ КОД?



# Как написать хороший ЧИСТЫЙ КОД?

```
МойКод = НаписатьГрязныйКод();
```

```
Пока НЕ КодСоответствуетМоимСтандартам()  
И НЕ КодМнеНравится() Цикл
```

```
МойКод = Рефакторить(МойКод);
```

```
КонецЦикла;
```

```
НакатитьВПродакшн(МойКод);
```



*Спасибо!*

**Контакты:**

Телеграмм: @avalexey

Почта:

[averyanovalexey@gmail.com](mailto:averyanovalexey@gmail.com)