Умение разбираться в чужом коде — Lurkmore

«Умение разбираться в чужом коде» — строка-детектор, которая содержится чуть более, чем во всех вакансиях на должность программиста и смежные должности. Чаще всего акцент на этой фразе делается для вакансий разработчика на C++ и PHP, где «чужой» не только код.

```
template <class TList, template <class> class Unit>
class GenScatterHierarchy;
// Специализация класса GenScatterHierarchy:
// преобразование класа Typelist в класс Unit
template <class T1, class T2, template <class> class Unit>
class GenScatterHierarchy<TypeList<T1, T2>, Unit>
   : public GenScatterHierarchy<T1, Unit>,
     public GenScatterHierarchy<T1, Unit>
{
public:
   typedef Typelist<T1, T2> TList;
   typedef GenScatterHierarchy<T1, Unit> LeftBase;
   typedef GenScatterHierarchy<T2, Unit> RightBase;
};

// Передача атомарного типа (не списка типов) классу Unit
template <class AtomicType, template <class> class Unit>
class GenScatterHierarchy : public Unit<AtomicType>
{
   typedef Unit<AtomicType> LeftBase;
};

// С классом NullType не делаем ничего
template <template <class> class Unit>
class GenScatterHierarchy<NullType, Unit>
{
};
```

Умение разбираться в чужом коде реквестед[1]

Значение фразы

- Вроде как компания ищет высококлассного специалиста.
- Этот специалист должен быть настолько хорошим, что мог бы даже разбираться в чужом коде.

На самом деле никакого такого умения не существует. Любой человек, способный писать, может и читать. Другое дело, что можно писать так, что потом сам хрен прочитаешь. Усугубляется это тем, что в сравнении с натуральными языками языки программирования, на человеческий взгляд, перегружены «пунктуацией», которая в них крайне важна и не терпит ошибок, а небольшие «слова» могут иметь очень большую (и неочевидную) смысловую нагрузку.

Потому и существуют гайдлайны, они же правила оформления. Например, не лениться использовать «говорящие» названия переменных и функций, даже если они длинноваты, оставлять комментарии в ключевых местах (то есть почти везде), не использовать goto-подобные операторы и так далее. И если подобных единых правил придерживаешься, то код хоть немного, но можно читать. Однако код, написанный в спешке, почему-то получается нечитабельным у большинства разработчиков.

Настоящее значение фразы

- Нужен идиот, который согласен разгребать это дерьмо.
- Руководитель свято верит, что причина получающегося дерьма в нерадивых программистах, а не в нём.
- Разгрести это дерьмо у тебя не получится при любом желании, так как от тебя будут требовать писать всё больше и больше кода, который непонятно как придётся прикручивать к уже существующему, на вдумчивое разгребание времени не останется совсем.
- В разработке архитектуры ПО используется метод «снизу вверх», требования меняются постоянно в зависимости от настроения левой пятки руководителя.
- С 2011 года среди Java-разработчиков фраза приобрела новый оттенок: программисту подсовывают («посмотреть», доделать, переделать, заставить работать…) под видом собственной разработки довольно качественный код, декомпилированный из какого-нибудь коммерческого продукта. Ну и…

Что мы узнаём об организации по такой вакансии

- Большая текучка кадров.
- Некому толком поставить процесс разработки в верную позу (при грамотной организации процесса разработки ПО код получается таким, что разобраться в нём не составит труда для любого программиста).
- Темпы разработки постоянно подгоняются начальством, качество кода ужасно, переписывать приходится больше, чем писать.
- Качество ПО, производимого этой конторой — дерьмо.
- Вам будут перманентно ебать мозг.

Что мы узнаём о работнике

Если фраза встречается в резюме, то означает симметричное:

- я готов копаться в любом дерьме;
- я буду писать такой код, разобраться в котором можно, только обладая данным умением;
- я умею заставить работать то, что накодили разработчики.

Когда сабж оправдан

Встречаются случаи, когда такое требование размещают в вакансиях оправданно. Это могут сделать после опыта работы с ковбоями-идеалистами. Когда такие люди сталкиваются с любой программой, они сразу же начинают истерить и пытаться переписать всё с нуля по своим лекалам. Тем не менее «умение

разбираться в чужом коде» — это ОСНЕ плохая формулировка. Годная альтернатива — это «терпимое отношение к чуждому коду». И всё равно строка, скорее всего, лишняя, так как должна подразумеваться в каждой вакансии программиста по умолчанию, что-то вроде столь же неимоверно популярного в вакансиях «умения работать в команде», которое на деле означает лишь «не быть мудаком» или «быть податливым мудаком».

Также есть конторы, специализирующиеся на поддержке чужих программ. Обычно востребованы в случаях, когда:

- за давностью лет авторов нельзя найти, или они отказываются от работы;
- документация армейски проёбана, и никто не знает, как оно работает;
- реверс-инжиниринг с целью сделать свой аналог с чем положено.

Здесь уже сабж действительно один из ключевых навыков. Но это уже, скорее, компетенция программиста-аналитика 80-го уровня, а не школьника-быдлокодера, и написано в таких терминах, что не сразу узнаешь, что это оно.

А на самом деле

А на самом деле это базовый навык, без которого нельзя работать программистом. Даже если чужой код был написан твоим коллегой буквально вчера и вы вместе работаете над одним проектом. Кстати, даже если с этим коллегой тесно общаетесь, то вам всё равно придётся разбираться в его коде без него хотя бы потому, что чисто физически тяжело запомнить все правки и то, зачем они были сделаны, особенно если проект большой. Вообще, умение разбираться в чужих наработках свойственно всем профессиям со времён индустриализации общества: новый разработчик твоего кредитного фокуса выкурил много старых чертежей перед тем, как сделать в них свои правки.

См. также

- Reverse Engineering
- Корпоративная культура
- История успеха
- ЖРАТ

Примечания

1. ↑ На самом деле, код из книжки Александреску и из либы Loki и при вдумчивом чтении всё совершенно понятно.

> — Я делаю особую, шаблонную магию
> — Не-не-не, Андрей Александреску, не-не-не
>
> — *Анекдот*

Ссылки

- Что делать, если сотрудник не пишет комментарии к коду.

Языки программирования
++i + ++i 1C AJAX BrainFuck C Sharp C++ Dummy mode Erlang Forth FUBAR God is real, unless explicitly declared as integer GOTO Haskell Ifconfig Java JavaScript LISP My other car Oracle Pascal Perl PHP Prolog Pure C Python RegExp Reverse Engineering Ruby SAP SICP Tcl TeX Xyzzy Анти-паттерн Ассемблер Быдлокодер Выстрелить себе в ногу Грязный хак Дискета ЕГГОГ Индусский код Инжалид дежице Капча КОИ-8 Костыль Лог Метод научного тыка Очередь Помолясь Проблема 2000 Программист Процент эс Рекурсия Свистелки и перделки Спортивное программирование СУБД Тестировщик Умение разбираться в чужом коде Фаза Луны Фортран Хакер
Языки программирования