

Коментарі, документація,
а також все що з цим пов'язане
і зможе зробити наш код ще кращим

Назар Герасимчук

December 7, 2018

Що і до чого?

Не коментуйте поганий код –
перепишіть його.

Б. Керніган і П. Плауер

- **Найкраща допомога** розробнику – хороший коментар в коді.
- **Найбільше засмічують** код беззмістовні коментарі.

Що і до чого?

Не коментуйте поганий код –
перепишіть його.

Б. Керніган і П. Плауер

- **Найкраща допомога** розробнику – хороший коментар в коді.
- **Найбільше засмічують** код беззмістовні коментарі.
- **Найбільшу шкоду** приносить коментар який втратив свою актуальність (хибний, дезінформативний).

Що і до чого?

Не коментуйте поганий код –
перепишіть його.

Б. Керніган і П. Плауер

- **Найкраща допомога** розробнику – хороший коментар в коді.
- **Найбільше засмічують** код беззмістовні коментарі.
- **Найбільшу шкоду** приносить коментар який втратив свою актуальність (хибний, дезінформативний).

Філософський погляд на коментарі

Насправді коментарі –
неуникне зло.

- Мови програмування недостатньо виразні.
- Програмування – процес невдалого виразу власних думок.

Філософський погляд на коментарі

Насправді коментарі –
неуникне зло.

- Мови програмування недостатньо виразні.
- Програмування – процес невдалого виразу власних думок.
- Коментарі – ознака невдачі. Гордитися тут нічим.

Філософський погляд на коментарі

Насправді коментарі –
неуникне зло.

- Мови програмування недостатньо виразні.
- Програмування – процес невдалого виразу власних думок.
- Коментарі – ознака невдачі. Гордитися тут нічим.

Приклад до обговорення

```
MockRequest request;  
private final String HTTP_DATE_REGEXP =  
    "[SMTWF][a-z]{2}\\.\s[0-9]{2}\s[JFMASOND][a-z]{2}\s"+  
    "[0-9]{4}\s[0-9]{2}:[0-9]{2}:[0-9]{2}\sGMT";  
private Response response;  
private FitNesseContext context;  
private FileResponder responder;  
private Locale saveLocale;  
// Пример: "Tue. 02 Apr 2003 22:18:49 GMT"
```

Що сталося з коментарем і з тією стрічкою яку він повинен був описувати?

Коментарі не компенсують поганого коду

Ви пишете модуль, і розумієте що він неконтрольовано розрісся, став дуже заплутаним. Тут, згадуєте цю презентацію^a і думаєте «А давай-но я зараз понаписую коментарів, щоб код був зрозуміліший».

Ні! Так не буде краще. Потрібно переписати код модуля!

^aну як вам рекурсія?

Пояснюйте **що** ви робите в коді

```
// Проверить, положена ли работнику полная премия  
if ((employee.flags & HOURLY_FLAG) &&  
    (employee.age > 65))
```

```
if (employee.isEligibleForFullBenefits())
```

Юридичні нотації

```

/*****
**
** Copyright (C) 2016 The Qt Company Ltd.
** Copyright (C) 2013 Olivier Goffart <ogoffart@woboq.com>
** Contact: https://www.qt.io/licensing/
**
** This file is part of the QtCore module of the Qt Toolkit.
**
** $QT_BEGIN_LICENSE:LGPL$
** Commercial License Usage
** Licensees holding valid commercial Qt licenses may use this file in
** accordance with the commercial license agreement provided with the
** Software or, alternatively, in accordance with the terms contained in
** a written agreement between you and The Qt Company. For licensing terms
** and conditions see https://www.qt.io/terms-conditions. For further
** information use the contact form at https://www.qt.io/contact-us.
**
** GNU Lesser General Public License Usage
** Alternatively, this file may be used under the terms of the GNU Lesser
** General Public License version 3 as published by the Free Software
** Foundation and appearing in the file LICENSE.LGPL3 included in the
** packaging of this file. Please review the following information to
** ensure the GNU Lesser General Public License version 3 requirements
** will be met: https://www.gnu.org/licenses/lgpl-3.0.html.
**
** GNU General Public License Usage
** Alternatively, this file may be used under the terms of the GNU
** General Public License version 2.0 or (at your option) the GNU General
** Public license version 3 or any later version approved by the KDE Free
** Qt Foundation. The licenses are as published by the Free Software
** Foundation and appearing in the file LICENSE.GPL2 and LICENSE.GPL3
** included in the packaging of this file. Please review the following
** information to ensure the GNU General Public License requirements will
** be met: https://www.gnu.org/licenses/gpl-2.0.html and
** https://www.gnu.org/licenses/gpl-3.0.html.
**
** $QT_END_LICENSE$
**
*****/
```

Юридичні нотації

```
// Copyright (C) 2003,2004,2005 by Object Mentor, Inc. All rights reserved.  
// Публикуется на условиях лицензии GNU General Public License версии 2 и выше.
```

Інформативні коментарі

```
// Возвращает тестируемый экземпляр Responder.  
protected abstract Responder responderInstance(); 1
```

```
// Поиск по формату: kk:mm:ss EEE, MMM dd, yyyy  
Pattern timeMatcher = Pattern.compile(  
    "\\d*:\\d*:\\d* \\w*, \\w* \\d*, \\d*"); 2
```

¹Хіба не краще перейменувати метод?

²Хіба не краще винести в спеціальний клас, що перетворює формати дати і часу?

Пояснення намірів

```
public int compareTo(Object o)
{
    if(o instanceof WikiPagePath)
    {
        WikiPagePath p = (WikiPagePath) o;
        String compressedName = StringUtil.join(names, "");
        String compressedArgumentName = StringUtil.join(p.names, "");
        return compressedName.compareTo(compressedArgumentName);
    }
    return 1; // Больше, потому что относится к правильному типу.
}
```

3

³Автор вирішив, що при порівнянні двох об'єктів об'єкти його класу повинні знаходитися в порядку сортування вище, чим об'єкти будь-якого іншого класу.

Пояснення намірів

```
public void testConcurrentAddWidgets() throws Exception {
    WidgetBuilder widgetBuilder =
        new WidgetBuilder(new Class[]{BoldWidget.class});
    String text = ""'"bold text'"";
    ParentWidget parent =
        new BoldWidget(new MockWidgetRoot(), ""'"bold text'"");
    AtomicBoolean failFlag = new AtomicBoolean();
    failFlag.set(false);

    // Мы пытаемся спровоцировать "состояние гонки",
    // создавая большое количество программных потоков.
    for (int i = 0; i < 25000; i++) {
        WidgetBuilderThread widgetBuilderThread =
            new WidgetBuilderThread(widgetBuilder, text, parent, failFlag);
        Thread thread = new Thread(widgetBuilderThread);
        thread.start();
    }
    assertEquals(false, failFlag.get());
}
```

4

⁴Можна не погодитися з тим як автор вирішує проблему, проте в крайньому разі ви знаєте що саме автор намагається зробити.

Пояснення

```
public void testCompareTo() throws Exception
{
    WikiPagePath a = PathParser.parse("PageA");
    WikiPagePath ab = PathParser.parse("PageA.PageB");
    WikiPagePath b = PathParser.parse("PageB");
    WikiPagePath aa = PathParser.parse("PageA.PageA");
    WikiPagePath bb = PathParser.parse("PageB.PageB");
    WikiPagePath ba = PathParser.parse("PageB.PageA");
    assertTrue(a.compareTo(a) == 0);    // a == a
    assertTrue(a.compareTo(b) != 0);    // a != b
    assertTrue(ab.compareTo(ab) == 0);  // ab == ab
    assertTrue(a.compareTo(b) == -1);   // a < b
    assertTrue(aa.compareTo(ab) == -1); // aa = ab
    assertTrue(ba.compareTo(bb) == -1); // ba < bb
    assertTrue(b.compareTo(a) == 1);    // b > a
    assertTrue(ab.compareTo(aa) == 1);  // ab > aa
    assertTrue(bb.compareTo(ba) == 1);  // bb > ba
}
```

Пояснення загадкового аргументу, або значення що повертається. Загалом варто обдумати як краще це *зобразити* засобами мови програмування. Хоча часто подібна ситуація трапляється коли ви працюєте з готовою бібліотекою чи модулем.

Пояснення і ризики

```
public void testCompareTo() throws Exception
{
    WikiPagePath a = PathParser.parse("PageA");
    WikiPagePath ab = PathParser.parse("PageA.PageB");
    WikiPagePath b = PathParser.parse("PageB");
    WikiPagePath aa = PathParser.parse("PageA.PageA");
    WikiPagePath bb = PathParser.parse("PageB.PageB");
    WikiPagePath ba = PathParser.parse("PageB.PageA");
    assertTrue(a.compareTo(a) == 0);    // a == a
    assertTrue(a.compareTo(b) != 0);    // a != b
    assertTrue(ab.compareTo(ab) == 0);  // ab == ab
    assertTrue(a.compareTo(b) == -1);   // a < b
    assertTrue(aa.compareTo(ab) == -1); // aa = ab
    assertTrue(ba.compareTo(bb) == -1); // ba < bb
    assertTrue(b.compareTo(a) == 1);    // b > a
    assertTrue(ab.compareTo(aa) == 1);  // ab > aa
    assertTrue(bb.compareTo(ba) == 1);  // bb > ba
}
```

Виникає ризик що коментар буде неправильним. Це справді важко перевірити, самі переконайтеся.

Попередження про можливі наслідки

```
// Не запускайте, если только не располагаете  
// излишками свободного времени.  
public void _testWithReallyBigFile()  
{  
    writeLinesToFile(100000000);  
    response.setBody(testFile);  
    response.readyToSend(this);  
    String responseString = output.toString();  
    assertSubString("Content-Length: 1000000000", responseString);  
    assertTrue(bytesSent > 1000000000);  
}
```

Можливо не найкраще рішення, але тут все зрозуміло з цим тестом.

Попередження про можливі наслідки

```
//  
// Dear maintainer:  
//  
// Once you are done trying to 'optimize' this routine,  
// and have realized what a terrible mistake that was,  
// please increment the following counter as a warning  
// to the next guy:  
//  
// total_hours_wasted_here = 42  
//
```

Хрестоматійний приклад.

Попередження про можливі наслідки

Можливо це не найкращі коментарі, проте, їх наявність точно відіб'є бажання у типового програміста витратити робочий час на щось інше, крім як на вирішення своїх прямих задач.

Коментарі TODO

```
// TODO - На данный момент эта функция не используется.  
// Ситуация изменится при переходе к отладочной модели.  
protected VersionInfo makeVersion() throws Exception  
{  
    return null;  
}
```

- Чого ф-ія не реалізована?
- Реалізувати потрібно, проте *чому* не зараз.
- Залежність?
- Функції в IDE.

Коментарі TODO

```
// I am not sure if we need this, but too scared to delete. 5
```

⁵Тяжко жити, шкода вмерти. Не всі TODO-коментарі корисні.

Погані коментарі

```
public void loadProperties()
{
    try
    {
        String propertiesPath = propertiesLocation + "/" + PROPERTIES_FILE;
        FileInputStream propertiesStream = new FileInputStream(propertiesPath);
        loadedProperties.load(propertiesStream);
    }
    catch(IOException e)
    {
        // Если нет файла свойств, загружаются настройки по умолчанию
    }
}
```

6

⁶Що означає коментар? Хто цим буде займатися? Чого добивався автор? Чи може це такий TODO-коментар?

Надлишкові коментарі

```
// Вспомогательный метод; возвращает управление, когда значение this.closed истинно.  
// Иницирует исключение при достижении тайм-аута.  
public synchronized void waitForClose(final long timeoutMillis)  
throws Exception  
{  
    if(!closed)  
    {  
        wait(timeoutMillis);  
        if(!closed)  
            throw new Exception("MockResponseSender could not be closed");  
    }  
}
```

7

⁷Коментар читається не простіше ніж сам код. Коментар є неточним і нав'язує читачу певні ідеї які можуть бути відмінні від того що декларується в коді.

Надлишкові коментарі

```
public abstract class ContainerBase
    implements Container, Lifecycle, Pipeline,
    MBeanRegistration, Serializable {

    /**
     * Задержка процессора для этого компонента.
     */
    protected int backgroundProcessorDelay = -1;

    /**
     * Поддержка событий жизненного цикла для этого компонента.
     */
    protected LifecycleSupport lifecycle =
        new LifecycleSupport(this);

    /**
     * Слушатели контейнерных событий для этого контейнера.
     */
    protected ArrayList listeners = new ArrayList();

    /**
     * Реализация загрузчика, связанная с контейнером.
     */
    protected Loader loader = null;
```

```
    /**
     * Реализация журнального компонента, связанная с контейнером.
     */
    protected Log logger = null;

    /**
     * Имя журнального компонента.
     */
    protected String logName = null;

    /**
     * Реализация менеджера, связанная с контейнером.
     */
    protected Manager manager = null;

    /**
     * Кластер, связанный с контейнером.
     */
    protected Cluster cluster = null;

    /**
     * Удобочитаемое имя контейнера.
     */
    protected String name = null;
```

Недостовірні коментарі

- Код зі слайду 25.
- Метод **не** повертає керування **коли** `this.closed` стає істинним.

```
// Вспомогательный метод: возвращает управление, когда значение this.closed истинно.  
// Иницирует исключение при достижении тайм-аута.  
public synchronized void waitForClose(final long timeoutMillis)  
throws Exception  
{  
    if(!closed)  
    {  
        wait(timeoutMillis);  
        if(!closed)  
            throw new Exception("MockResponseSender could not be closed");  
    }  
}
```

Недостовірні коментарі

```
// Вспомогательный метод: возвращает управление, когда значение this.closed истинно.  
// Иницирует исключение при достижении тайм-аута.  
public synchronized void waitForClose(final long timeoutMillis)  
throws Exception  
{  
    if(!closed)  
    {  
        wait(timeoutMillis);  
        if(!closed)  
            throw new Exception("MockResponseSender could not be closed");  
    }  
}
```

- Код зі слайду 25.
- Метод **не** повертає керування **коли** `this.closed` стає істинним.
- Метод повертає керування **якщо** `this.closed` істинне, інакше, чекає таймауту і генерує виключення у випадку якщо `this.closed` так і не стало істинне.

Недостовірні коментарі

```
// Вспомогательный метод: возвращает управление, когда значение this.closed истинно.  
// Иницирует исключение при достижении тайм-аута.  
public synchronized void waitForClose(final long timeoutMillis)  
throws Exception  
{  
    if(!closed)  
    {  
        wait(timeoutMillis);  
        if(!closed)  
            throw new Exception("MockResponseSender could not be closed");  
    }  
}
```

- Код зі слайду 25.
- Метод **не** повертає керування **коли** `this.closed` стає істинним.
- Метод повертає керування **якщо** `this.closed` істинне, інакше, чекає таймауту і генерує виключення у випадку якщо `this.closed` так і не стало істинне.
- Проблеми будуть не в того хто це написав, а в того хто унаслідуює цей код.

Недостовірні коментарі

```
// Вспомогательный метод: возвращает управление, когда значение this.closed истинно.  
// Иницирует исключение при достижении тайм-аута.  
public synchronized void waitForClose(final long timeoutMillis)  
throws Exception  
{  
    if(!closed)  
    {  
        wait(timeoutMillis);  
        if(!closed)  
            throw new Exception("MockResponseSender could not be closed");  
    }  
}
```

- Код зі слайду 25.
- Метод **не** повертає керування **коли** `this.closed` стає істинним.
- Метод повертає керування **якщо** `this.closed` істинне, інакше, чекає таймауту і генерує виключення у випадку якщо `this.closed` так і не стало істинне.
- Проблеми будуть не в того хто це написав, а в того хто унаслідуює цей код.

Журнальні коментарі

- * Изменения (начиная с 11 октября 2001)
 - * -----
 - * 11.10.2001 : Реорганизация класса и его перемещение в новый пакет
* com.jrefinery.date (DG);
 - * 05.11.2001 : Добавление метода getDescription(), устранение класса
* NotableDate (DG);
 - * 12.11.2001 : С устранением класса NotableDate IBD требует включения
* метода setDescription() (DG); исправление ошибок
* в методах getPreviousDayOfWeek(), getFollowingDayOfWeek()
* и getNearestDayOfWeek() (DG);
 - * 05.12.2001 : Исправление ошибки в классе SpreadsheetDate (DG);
- 8

⁸З моменту коли почали використовувати системи контролю версій коду – про подібні речі треба одразу ж забувати.

Шум

```
/**
 * Конструктор по умолчанию.
 */
protected AnnualDateRule() {
}

/** День месяца. */
private int dayOfMonth;

/**
 * Возвращает день месяца.
 *
 * @return день месяца.
 */
public int getDayOfMonth() {
    return dayOfMonth;
}
```

Шум

```
int main(void)
/* Program starts here */
```


Небезпечний шум

```
/** Имя. */  
private String name;  
/** Версия. */  
private String version;  
/** Название лицензии. */  
private String licenceName;  
/** Версия. */  
private String info;
```

- Яка користь з коментарів такого типу?
- Це шумові коментарі, викликані бажанням когось покоментувати.

Небезпечний шум

```
/** Имя. */  
private String name;  
/** Версия. */  
private String version;  
/** Название лицензии. */  
private String licenceName;  
/** Версия. */  
private String info;
```

- Яка користь з коментарів такого типу?
- Це шумові коментарі, викликані бажанням когось покоментувати.
- Перечитайте уважніше.

Небезпечний шум

```
/** Имя. */  
private String name;  
/** Версия. */  
private String version;  
/** Название лицензии. */  
private String licenceName;  
/** Версия. */  
private String info;
```

- Яка користь з коментарів такого типу?
- Це шумові коментарі, викликані бажанням когось покоментувати.
- Перечитайте уважніше.
- Знайшли помилку?

Небезпечний шум

```
/** Имя. */  
private String name;  
/** Версия. */  
private String version;  
/** Название лицензии. */  
private String licenceName;  
/** Версия. */  
private String info;
```

- Яка користь з коментарів такого типу?
- Це шумові коментарі, викликані бажанням когось покоментувати.
- Перечитайте уважніше.
- Знайшли помилку?
- Ctrl+C + Ctrl+V
- Якщо автори не слідкують за коментарями в момент написання, то як можна очікувати що такі коментарі принесуть користь читачам?

Небезпечний шум

```
/** Имя. */  
private String name;  
/** Версия. */  
private String version;  
/** Название лицензии. */  
private String licenceName;  
/** Версия. */  
private String info;
```

- Яка користь з коментарів такого типу?
- Це шумові коментарі, викликані бажанням когось покоментувати.
- Перечитайте уважніше.
- Знайшли помилку?
- Ctrl+C + Ctrl+V
- Якщо автори не слідкують за коментарями в момент написання, то як можна очікувати що такі коментарі принесуть користь читачам?

Небезпечний шум

```
/**  
 * Always returns true.  
 */  
public boolean isAvailable() {  
    return false;  
}
```

```
//private instance variable for storing age  
public static int age;
```

Коментар чи функція?

```
// Зависит ли модуль из глобального списка <mod> от подсистемы,  
// частью которой является наш код?  
if (smodule.getDependSubsystems().contains(subSysMod.getSubSystem())) 9
```

```
ArrayList moduleDependees = smodule.getDependSubsystems();  
String ourSubSystem = subSysMod.getSubSystem();  
if (moduleDependees.contains(ourSubSystem))
```

⁹Імовірно програміст спершу написав коментар, потім по коментарю написав відповідний код. Але він забув переглянути написаний код...

Закоментований код

В програмуванні навряд чи трапиться звичка більш огидна як залишений закоментований код.

- Іншим програмістам не вистачить хоробрості його видалити.
- Чи важливі закоментовані стрічки?
- Може це просто *мотлох*?
- Забули про систему контролю версій?

Нелокальна інформація

```
/**
 * Порт, на котором будет работать fitnessse. По умолчанию <b>8082</b>.
 *
 * @param fitnesssePort
 */
public void setFitnesssePort(int fitnesssePort)
{
    this.fitnesssePort = fitnesssePort;
}
```

10

¹⁰Хто прослідкує за зміною порта по замовчуванню?

Неочевидні коментарі

```
/*  
 * Начать с массива, размер которого достаточен для хранения  
 * всех пикселей (плюс байты фильтра), плюс еще 200 байт  
 * для данных заголовка  
 */  
this.pngBytes = new byte[((this.width + 1) * this.height * 3) + 200]; 11
```

¹¹Що таке байти фільтра? Як вони пов'язані з +1? Або з *3? Один піксель відповідає одному байту? Чого 200?..

Цікаві коментарі в історії

```
/*
 * Switch to stack of the new process and set up
 * his segmentation registers.
 */
retu(rp->p_addr);
sureg();
/*
 * If the new process paused because it was
 * swapped out, set the stack level to the last call
 * to savu(u_ssav). This means that the return
 * which is executed immediately after the call to aretu
 * actually returns from the last routine which did
 * the savu.
 *
 * You are not expected to understand this.
 */
if(rp->p_flag&SSWAP) {
    rp->p_flag =& ~SSWAP;
    aretu(u.u_ssav);
}
```

Цікаві коментарі в історії

```
/*  
 * 'schedule()' is the scheduler function. It's a very simple and nice  
 * scheduler: it's not perfect, but certainly works for most things.  
 * The one thing you might take a look at is the signal-handler code here.  
 *  
 * NOTE!! Task 0 is the 'idle' task, which gets called when no other  
 * tasks can run. It can not be killed, and it cannot sleep. The 'state'  
 * information in task[0] is never used.  
 *  
 * The "confuse_gcc" goto is used only to get better assembly code..  
 * Dijkstra probably hates me.  
 */  
asm linkage void schedule(void)
```

Практична частина

Джерела I



Robert C. Martin.

Clean Code - a Handbook of Agile Software Craftsmanship.
Prentice Hall, 2009.



B. W. Kernighan and P. J. Plauge

The Elements of Programming Style.
New York, 1978.



Dennis Ritchie.

Odd Comments and Strange Doings in Unix.

<http://web.archive.org/web/20040206202840/http://cm.bell-labs.com/cm/cs/who/dmr/odd.html>

Джерела II



Говнокод.ру

«По колено в коде» <https://www.govnokod.ru/>