1/13/2023

Clean code

Ch4 Comments

Comments

"Don't comment bad code—rewrite it."

—Brian W. Kernighan and P. J. Plaugher[1]

Abdurahman Gamal Ahmed
NO CAMPANY

Table of Contents

1-intro :

اول حاجة عنعلق عليها هى الغلاف عم بوب كاتب ف الغلاف متعملش comment للكود السئ عدله احسن .

مفيش حاجة مفيدى اكتر من comment مكتوب بصح ومفيش حاجة مضره اكتر من comment قديم كله مكتوب غلط . والcomment هو شر لابد منه ولو عندنا الموهبه الكافيه لستخدام لغات البرمجه للتعبير عن قصدنا مكناش احتجنا ل comment اصلا

- ف comment بتعوض فشلنا ف والتعبير عن قصدنا بالكود . وعم بوب هو الى استخدام كلمه فشل وبيقلك ان ال comment دايما فشل بس مش هنقدر نستغنا عنهم عشان نعوض فشلنا ف التعبير عن قصدنا بالكود
- واستخدام ال comment بختصار هو حاجة وحشه ومتفرحش لما تعمل comment .
- ولما تلاقى نفسك عاوز تكتب comment فكر كويس وشوف طريقه تعبر بيها بالكود ومتكتبش comment .
- وفى كل مرة تكتب فيها comment لازم تكون مضايق وزعلان من نفسك ولما تخلى الكود معبر وتشيل comment افتخر بنفسك يلا .

طيب ليه عم بوب بيكره ال comment : عشان غاالبا بيكون مضلل مش دايما بس فى حلات كتير وكل ما كان ال comment قديم وبعيد عن الكود كل مكان هيبقا غاالبا غلط وقصه بكلمه بعيد ان ف مكان تانى بعيد عن ال code .

- ودة مثال : ف الكود الى جاى دة ال كومنت مفروض يكون بيوصف اول سطر بتاع ال Http ف مع الوقت ال comment بعد عن الكود الى بيوصفه عشان اضاف حوالى 4 سطور

```
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;
// Example: "Tue, 02 Apr 2003 22:18:49 GMT"
```

- وهو حلو ان ال programmer يكتبه comment بس الاحسن ان يوفر الطاقه دى ف انه يكتب كود معبر من الاول
- وال comment غير الدقيقه اسواء بكتير من عدم وجود comment صلا عشان بتكون مضلله وبيكون فيه توقعات عمرها مل هتحصل والحقيقه داميا ف مكان واحد الى هو الكود بس برضو مش هنستغنى عن ال comment نهائيا احنا هنعمل مجهود عشان نقللها .

2- Comments Do Not Make Up for Bad Code:

واحد من الحجات المحفزه لكتابه ال comment هو الكود السئ . احنا بنكتب كود module وعرفين انه ملغبط وغير منظمه ف بنقول لنفسنا لازم تكتب comment بس دة مش صح الافضل انك تخليه هو clean

- الكود الواضح المنظم الى فيه comment قليله احسن بكتير من الكود الى فيه comment كتير وملغبط وبدال مضيع وقتك ف كتابه comment توصف اللغبطه الى انت اعملها اعمل كود نضيف من الاول ياخى

3- Explain Yourself in Code:

اكيد ف اوقات بيكون فيها الكود وسيله سيئه للتفسير وللاسف كتير من المبرمجين فاكرين ان نادرا لما الكود يكون وسيله كويسه للتفسير وده مش صح يعنى المبرمجين بيكونوا فاكرين ان الكود مش هينفع يكون معبر كفايه ودة مش صح

وعم بوب بيديك مثال عشان تقتنع ان ال comment هو مش افضل حاجة للتعبير عن الكود :

```
// Check to see if the employee is eligible for full benefits
if ((employee.flags & HOURLY_FLAG) &&
    (employee.age > 65))
```

Or this?

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

ف عم بوب بيسئلك تحب تشوف انهى كود من دول الاول ولا التانى ؟

- طبعا التانى عشان اصغر ومعبر اكتر اما الاول دة على الرغم ان فيه comment هو كبير ومش واضح وهاخد وقت عشان اقره الكود والكومنت .
- الموضع بيستغر ثوانى عشان تشرح قصدك بستخدام الكود نفسه وفى حلات كتير اواى انك انت هتسمى function بنفس اسم الcomment الى كنت هتعمله زى حلتنا هنا . فابعت عن الكومنت بصحبى .

4-Good Comments :

ف اوقات بتكون ال comment مفيدة ف هنشوف حلات منها ناو . وخليك فاكر ان احسن comment هو الى متكتبش .

4.1- Legal Comments:

احيانا بتجبرنا شركتنا على كتابه comment معينه زى بتاعه حقوق النشر والطبافه الى بتبقا ف اول كل file . ودة شكلها

```
// Copyright (C) 2003,2004,2005 by Object Mentor, Inc. All rights reserved.
// Released under the terms of the GNU General Public License version 2 or later.
```

بس خلى ال comment دى مش كبيره وخليها بتشير لحاجة موجوده بره الكودالى حابب يريجها براحته .

4.2- Informative Comments :

احيانا بيكون مفيد تقدم معلومات اساسيه عن طريق ال comment . زى المثال ده

```
// Returns an instance of the Responder being tested.
protected abstract Responder responderInstance();
```

- وهنا ال comment دة مفيد بيدينا معلومات عن الى راجع من ال abstract method بس برضو لو خليت اسم الداله معبر احسن يعنى لو خليتها كدة : `responderBeingTested`.

ودة مثال تانى : وهنا ال comment مفيد اكتر من المثال الى فات عشان ببعرفنا غرض ال regular expression ان هو بيعمل match لل timeو ال data وشكل بتاع ال format .

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

- بس برضو الموضع هيكون احسن لو ال كود دة اتنقل وتحط لكلاس تانى بيحول ال data,time .
- يعنى هنا عم بوب بيقلك ان الكومنت مفيد بس كان فيه حل تانى يعوضك عن ال comment .

4.3-Explanation of Intent:

الكومنت احيانا بيبقا اكتر من معلومات مفيده عن ال Implementation وبيبقا بيوضح الغرض او النيه من قرار ما . زى الصورة دى

```
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; // we are greater because we are the right type.
}
```

- ف المثال دة بيعمل مقاره بين two object وبيقرر ان يرتب الكلاس بتاعه بس الى هو اعلى من اى object . مش مهم لو مفهمتش السطر دة المهم انك تفهم التعليق بتاع المبرمج هنا هو علق على النيه بتاعته معلقش على كود معين او جزء من كود.

ودة مثال تانى على كتابه comment لوصف من عمل كود او وصف فكرة : ف المثال هنا المبرمج بيقلك عشان يحصل على race condition هو هيعمل عدد كبير من ال thread ف انت كدة هتفهم ايه المشكله الى عاوز يحلها .

```
  String text = "'''bold text'''";
  ParentWidget parent =
    new BoldWidget(new MockWidgetRoot(), "'''bold text'''");
  AtomicBoolean failFlag = new AtomicBoolean();
  failFlag.set(false);

  //This is our best attempt to get a race condition
  //by creating large number of threads.
  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());
}
```

4.4-Clarification:

Clarification دى معنها الوضوح.

اوقات بيكون كويس انك تترجم معنى argument او return value . بس طبعا الاحسن انك تخلى ال argument واضح اصلا .

بس لو مش هينفع ف خلاص عمل comment توضيحى هيون مفيد .

- بس ال comment التوضحيى دة ف نسبه خطوره انه ميكونش صح زى الصورة دى.
- الكومنت الى موجود اصلا دة ممكن يكون غلط اكيد من غير متقصد ف لما تكتب comment زى دى اتاكد نك كتبتهم صح واتاكد ان مفيش طريقه احسن او بديله لل comment .

```
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
}
```

4.5- Warning of Consequences :

احيانا بيكون مفيد انك تحز المبرمجين من عواقب محتمله . زى ال comment ده .

ف الكومنت دة بيقلك مترنش اصلا ال function دى الا لو عندك وقت عشان تضيعه يعنى قصدة ان ال function دى بتاخد وقت طويل وكده

```
// Don't run unless you
// have some time to kill.
public void _testWithReallyBigFile()
{
  writeLinesToFile(10000000);

  response.setBody(testFile);
  response.readyToSend(this);
  String responseString = output.toString();
  assertSubString("Content-Length: 1000000000", responseString);
  assertTrue(bytesSent > 1000000000);
}
```

- طبعا ف ايمامنا دى ينفع نقفل الفانكشن بستخدام @Ignore() والبتاخد comment جوها وده شكلها كامل ومناسب لحلتنا دى @Ignore("Takes too long to run")
- وزمان برضو كان فيه اتفاقيه ان ال function الى تبداء ب _ كدة معنها متستخدمهاش

وده مثال تانى : المثال دة برضو بيحذر الشغص ان ممكن يحصل مشكله لو هتشتغل parallel .

```
public static SimpleDateFormat makeStandardHttpDateFormat()
{
  //SimpleDateFormat is not thread safe,
  //so we need to create each instance independently.
  SimpleDateFormat df = new SimpleDateFormat("EEE, dd MMM  yyyy HH:mm:ss z");
  df.setTimeZone(TimeZone.getTimeZone("GMT"));
  return df;
}
```

4.6- TODO Comments :

ال comment هنا او ال todo بيفسر سبب ال implementation وبيوضح شكل ال function شكلها هيكون ازاى ف المستقبل .

وال todo دى الحجات الى المبرمجين بيعتقدة لسبب ما انها مفروض تخلص بس لسبب ما مش هينفع تخلص دلوقتى ودى ممكن تبقا طلب لشخص ما ان يختار اسم احسن او يمسح كود عشان بقى متكرر او عشان يعمل تعديل معتمد على even .

- ومهما كان ال doto هيبقا ازاى او ليه مفيش مبرر انك تسيب كود ملهوش لزمه ف البرنامج .
- والايام دى كل ال IDE فيها مكان مخصص للDoto فمش هيضيعه ودة مش مبرر انك تمله الدنيا todo .

4.7- Amplification :

ممكن نستخدام ال comment عشان نضخم اهميه شئ ومن غير التضخيم ممكن الحاجة دى تبان مش منطقيه زى المثال ده :

```
String listItemContent = match.group(3).trim();
// the trim is real important.  It removes the starting
// spaces that could cause the item to be recognized
// as another list.
new ListItemWidget(this, listItemContent, this.level + 1);
return buildList(text.substring(match.end()));
```

4.8- Javadocs in Public APIs:

Javadocs دى toolبتعمل بطلعلك documentation على شكل كود html .

مفيش حاجة اجمل من public api مصوف كويس ولو بتكتب api المفروض تكتب Javadocs ليه كويسه . بس خلى بالك بروض ال Javadocs ممكن تبقا مضلله ومش صح زيها زى ال comment عادى يعنى ينطبق عليها كل قواعد ال comment عادى .

وهنشوف بعد كدة ان Javadocs هتكون مش كويسه ومضلله لو ال api بتاعك مكنش مفوض يبقا public اصلا .

5- Bad Comments:

عم بوب بيقلك ان اغلب ال comment ممكن نقول عليها bad وغالبا بتكون اعزار للكود السئ او مبررات لقرارات خاطئه .

5.1- Mumbling:

Mumblingمعنها التمتمه بكلام مش واضح او زى الراب كده .

انك تكتب comment عشان حسيت انك مفروض تكتب اوان ال process محتجاه ف دة مش صح ولوخلاص هتكتب comment خد وقتك عشان تتاكد ان ده افضل comment ممكن يتكتب.

- وفى الاسكرين دى ال comment مفيد بس الى كتبه كان مستعجل او مش مهتم ف الى كتبه مش مفهم او التمتمه الى كتبها مش مفهومه

```
public void loadProperties()
{
  try
  {
   String propertiesPath = propertiesLocation + "/" + PROPERTIES_FILE;
   FileInputStream propertiesStream = new FileInputStream(propertiesPath);
   loadedProperties.load(propertiesStream);
  }
  catch(IOException e)
  {
    // No properties files means all defaults are loaded
  }
}
```

- الكومنت هنا مش واضح معنها لما يحصل exception ف الى هيحصل ان مفش اى property وكمان معناه ال default اتحملت بس مين الى حملها وفين اصلا ال statement الى حملتها هل قبل loadProperties.load ولا بعدها ولا مفروض هو يكتبها اصلا ف ال catchبس لسه مكتبش الكود بتاع ال load ؟ الكومنت مش موضح
- ف الحل الوحيد عشان تكتشف هو قصده ايه ان افصح الكود دة ف اجزاءتانيه من ال system واى comment بيجبرك انك تفحص moduleتانيه عشان نفهم معنى ال comment نفسه ف ال comment دة ملهوش لزمه .

5.2- Redundant Comments :

Redundant Comments يعنى ال comment الزائده او ال ملهاش لزمه. وناخد مثال على طول

Listing 4-1
waitForClose

```
// Utility method that returns when this.closed is true. Throws an exception
// if the timeout is reached.
public synchronized void waitForClose(final long timeoutMillis)
throws Exception
{
  if(!closed)
  {
    wait(timeoutMillis);
    if(!closed)
      throw new Exception("MockResponseSender could not be closed");
  }
}
```

- ف الصوره الى فاتت ال comment الى فى اول الاسكرين ده ملهوش لزمه مش بيوضح نيه ولا بيفسر الكود . هو بيغرى الى هيقراء الكود انه يقبل عدم الدقه بدال الفهم الحقيقى الدقيق .

- ف المضوع عامل زى تاجر العربيات الى بيقلك خد العربيه ومتبصش ف الكبوت عشان الدنيا خربانه جوه .

ناخد مثال كمان : الكومنت الى ف الكود دى ملهاش اى لزمه دى كمان بتخفى الكود .ف شوف الاسكرين كويس ومتعملش زيها

ContainerBase.java (Tomcat)

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

  /**
   * The processor delay for this component.
   */
  protected int backgroundProcessorDelay = -1;


  /**
   * The lifecycle event support for this component.
   */
  protected LifecycleSupport lifecycle =
    new LifecycleSupport(this);


  /**
   * The container event listeners for this Container.
   */
  protected ArrayList listeners = new ArrayList();


  /**
   * The Loader implementation with which this Container is
   * associated.
   */
  protected Loader loader = null;


  /**
   * The Logger implementation with which this Container is
   * associated.
   */
  protected Log logger = null;


  /**
   * Associated logger name.
   */
  protected String logName = null;
```

5.3- Misleading Comments:

Misleading Comments يعنى الكومنت المضلله. مع كل النوايه الحسنه اوقات المبرمج بيكتب حجات مضلله

- زى الكومنت ده هو مكتوب غلط بيقلك ان ال function بتعمل return لما closed تبقا true وده غلط .وهيكون مضلل لاى حد بيستخدام الكود دة

Listing 4-1
waitForClose

```
// Utility method that returns when this.closed is true. Throws an exception
// if the timeout is reached.
public synchronized void waitForClose(final long timeoutMillis)
throws Exception
{
  if(!closed)
  {
    wait(timeoutMillis);
    if(!closed)
      throw new Exception("MockResponseSender could not be closed");
  }
}
```

5.4- **Mandated Comments**:

Mandated Comment ده معناه التعليقات المطلوبه .

عم بوب بيقلك هيكون الموضوع سخيف لو فيه قاعده بتقلك ان كل function لازم يبقا فيها comments ف ال comment دى هتبقا عقبه وعب على الفاضى زى الصورة دى .

- ف الcomments الى هنا ملهاش اى تلاتين لزمه .

Listing 4-3

```
/**
 *
 * @param title The title of the CD
 * @param author The author of the CD
 * @param tracks The number of tracks on the CD
 * @param durationInMinutes The duration of the CD in minutes
 */
public void addCD(String title, String author,
                  int tracks, int durationInMinutes) {
  CD cd = new CD();
  cd.title = title;
  cd.author = author;
  cd.tracks = tracks;
  cd.duration = duration;
  cdList.add(cd);
}
```

5.5 - **Journal Comments** :

Journal دى معنها ف المحاسبه دفتر اليوميه ودى بيتسجل فيها كل حاجه حرفين . فدة حاجة زى ال history بتاع الكود يعنى هتكتب comment بتاريع عمل ال code وتريخ كل التعديلات ف عم بوب بيقلك الحجات دى دلوقتى ملهاش اى لزمه بعد ظهرو ال gethub, git . وده الشكل الى بيقول عليه عم بوب .

```
* Changes (from 11-Oct-2001)
* --------------------------
* 11-Oct-2001 : Re-organised the class and moved it to new package
*                com.jrefinery.date (DG);
* 05-Nov-2001 : Added a getDescription() method, and eliminated NotableDate
*                class (DG);
* 12-Nov-2001 : IBD requires setDescription() method, now that NotableDate
*                class is gone (DG);  Changed getPreviousDayOfWeek(),
*                getFollowingDayOfWeek() and getNearestDayOfWeek() to correct
*                bugs (DG);
* 05-Dec-2001 : Fixed bug in SpreadsheetDate class (DG);
* 29-May-2002 : Moved the month constants into a separate interface
*                (MonthConstants) (DG);
* 27-Aug-2002 : Fixed bug in addMonths() method, thanks to N???levka Petr (DG);
* 03-Oct-2002 : Fixed errors reported by Checkstyle (DG);
* 13-Mar-2003 : Implemented Serializable (DG);
* 29-May-2003 : Fixed bug in addMonths method (DG);
* 04-Sep-2003 : Implemented Comparable.  Updated the isInRange javadocs (DG);
* 05-Jan-2005 : Fixed bug in addYears() method (1096282) (DG);
```

5.6- **Noise Comments**:

Noise Comments دى بتكون مزعجه ومش بضيف اى جديد. زى الكود ده :

```
/**
 * Default constructor.
 */
protected AnnualDateRule() {
}
```

- الكومنت دى ملهاش لزمه يعنى هنا عامل كومنت قبل ال constructor ف اى حد عارف oop عارف يعنى ايه constructor ف ملهوش لزمه تعمل كومنت

No, *really?* Or how about this:

```
/** The day of the month. */
    private int dayOfMonth;
```

And then there's this paragon of redundancy:

```
/**
 * Returns the day of the month.
 *
 * @return the day of the month.
 */
public int getDayOfMonth() {
  return dayOfMonth;
}
```

وده مثال تانى : عم بوب بيقول ف الصوره دى اول كومنت منطقى وبيوضح ليه تم تجاهل ال catch body بس تانى كومنت ده مش مفهوم والظاهر ان المبرمج كان عاوز يرح ف كتب الكومنت ده . والاحسن ليه انه كان يحسن الكود بالى الكومنت ده

Listing 4-4
startSending

```
private void startSending()
{
  try
  {
    doSending();
  }
  catch(SocketException e)
  {
    // normal. someone stopped the request.
  }
  catch(Exception e)
  {
    try
    {
      response.add(ErrorResponder.makeExceptionString(e));
      response.closeAll();
    }
    catch(Exception e1)
    {
      //Give me a break!
    }
  }
```

- وكان ممكن يخل ال body بتاع ال catch يكون function زى الكود ده :

- ف ده الحل للحاله الى فوق الى كان مفروض المبرمج يعملها كل الى حصل ان عملت function وديتها ف تانى catch عشان مسبش ال body بتاع تانى catch فاضى .

Listing 4-5
startSending (refactored)

```
private void startSending()
{
  try
  {
    doSending();
  }
  catch(SocketException e)
  {
    // normal. someone stopped the request.
  }
  catch(Exception e)
  {
    addExceptionAndCloseResponse(e);
  }
}

private void addExceptionAndCloseResponse(Exception e)
{
  try
  {
    response.add(ErrorResponder.makeExceptionString(e));
    response.closeAll();
  }
  catch(Exception e1)
  {
  }
}
```

5.6- Scary Noise:

الjavadoc اوقات ممكن تبقا مزعجه . خد مثال يحب

- ايه الغرض من ال comment دى ؟ولا حاجه هم ازعاج لرغبه خاظئه عشان تبقا documentation .

```
/** The name. */
private String name;

/** The version. */
private String version;

/** The licenceName. */
private String licenceName;

/** The version. */
private String info;
```

- ولو دققت ف الاسكرين هتلاقى كمان ف غلط ف اخر كومنت بدال info مكتوب version . ف اصلا الكومنت عملها غلط ملهاش لزمه يعنى وكمان كاتب غلط يعنى اخر عك

5.7- Don't Use a Comment When You Can Use a Function or a Variable :

متعملش comment طول م انت تقدر تستخدام function او variable .

Consider the following stretch of code:

```
// does the module from the global list <mod> depend on the
// subsystem we are part of?
if (smodule.getDependSubsystems().contains(subSysMod.getSubSystem()))
```

This could be rephrased without the comment as

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

- وفى المثال الى فات ده هو عمل صياغه للكود من غير الكومنت وبيقلك ممكن يكون المبرمج كتب الكومنت عشان يوصف الحاجه الى هيعملها قبل ميكتب كود اصلا بس كان مفروض يرجع ويعمل refactoring زى ما عم بوب عمل كدة .
- وشرحت من اول الشبتر ان الحاجة الوحديه الحقيقيه هى الكود عشان كده حولنا من كومنت لكود .

5.8- Position Markers:

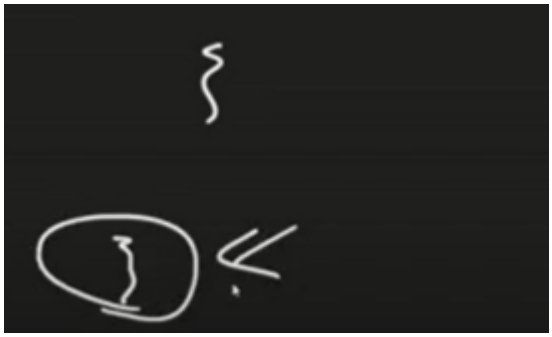
السكشن دة بيوضحلك ان الحركه دى ملهاش لزمه ف اعلب الاحبان . الحركه دى ممكن تعملها عشان تجمع شويه method تحتها ف عم بوب بيقلك نادرا لما يبقا منطقى انك تجمع تجمع method تحت ال banner ده . والbanner بيبقا واضح خاالص ف استعمله صح او ف موضعه وهو ف العموم فوضه لازم تتشال

Banner هو شويه ال / الى وره بعض.

5.9- Closing Brace Comments :

دى ال Closing Brace Comments الى عم بوب يقصدها .وبيقلك ان كتير من المبرمجين بيحطه comment ف اخر Closing Brace .

زى الصورة الى جايه



Listing 4-6
wc.java

```
public class wc {
  public static void main(String[] args) {
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    String line;
    int lineCount = 0;
    int charCount = 0;
    int wordCount = 0;
    try {
      while ((line = in.readLine()) != null) {
        lineCount++;
        charCount += line.length();
        String words[] = line.split("\\W");
        wordCount += words.length;
      } //while
      System.out.println("wordCount = " + wordCount);
      System.out.println("lineCount = " + lineCount);
      System.out.println("charCount = " + charCount);
    } // try
    catch (IOException e) {
      System.err.println("Error:" + e.getMessage());
    } //catch
  } //main
}
```

- ف الداله الى فاتت عند كل Closing Brace هتلاقيه كاتب comment باسم زى main و while الخ و فى حاله ال function الطويله الى فيها اكتر من nested structure زى حلتنا هنا دة ممكن يكون منطقى بس الكلام دة ضد ال فانكشن الصغيره الى احنا بنحاول نعملها وبنحبها ف لو لقيت نفسك محتاج تحدد ال Closing Brace بالطريقه دى صغر الفانكشن بتاعتك احسلك .

5.10- Attributions and Bylines:

مفيش داعى تعمل الشكل ده عشان تحدد مين الى ضاف جزء معين .

```
/* Added by Rick */
```

يعنى بختصار بعد ظهور ال git خلاص مبقيش ليه لزمه الموضع ده

5.11- Commented-Out Code :

حاجه مش كويسه ف انك تعمل commentلكود معين وتسيبه كده . زى الاسكرين دى

```
    InputStreamResponse response = new InputStreamResponse();
    response.setBody(formatter.getResultStream(), formatter.getByteCount());
//  InputStream resultsStream = formatter.getResultStream();
//  StreamReader reader = new StreamReader(resultsStream);
//  response.setContent(reader.read(formatter.getByteCount()));
```

- الى هيشوف الكومنت دى مش هيكون عنده الشجاعه ان يمسحهم وممكن يكونوا مش مهمين ف متعمل كده .

وف الكود الى جاى اشمعنا السطرين دول معملهم كومنت . هل هم مهمين وموجدين عشان يفكرونا بتعديل هيحصل . ولا هو كود ملهوش لزمه وحد كسل يمسحهم ؟ ف عم بوب بيقلك متعملش كده والموضوع ملهوش لزمه خاالص فى وجود ال git ف متخفش على الكود مش هيضيع وامسح الكومنت دول .

```
this.bytePos = writeBytes(pngIdBytes, 0);
//hdrPos = bytePos;
writeHeader();
writeResolution();
//dataPos = bytePos;
if (writeImageData()) {
     writeEnd();
     this.pngBytes = resizeByteArray(this.pngBytes, this.maxPos);
}
```

5.12- HTML Comments:

ده ملهوش لزمه ف هسيبه ناو .

5.13- Nonlocal Information:

لو لازم تكتب كومنت اتاكد ان هو بيوصف الكود القريب منه متكتبش معلومات عن الsystem فى commentعادى فى اى فانكشن او كلاس

```
/**
 * Port on which fitnesse would run. Defaults to <b>8082</b>.
 *
 * @param fitnessePort
 */
public void setFitnessePort(int fitnessePort)
{
  this.fitnessePort = fitnessePort;
}
```

- ف الاسكرين الى فاتت دى بعض النظر عن حقيقه الكومنت ملهوش لزمه اصلا . هو بيوصف معلومه عن ال default portوالفانكشن ملهاش علاقه بال port ده اصلا الكومنت مش بيوصف الفانكشن دة بيوصف شئ ف ال system وكمان مفيش اى ضمان ان لما ال portيتغير فى system هيتغير كمان ف الكومنت.

5.14- Too Much Information :

متحطش مناقشات تارخيه وتافضيل زياده ف الكومنت . زى الاسكرين دى

- والكومنت ده معمول ل module عشان يعمل test for function بتعمل encode ,decode base 46 حاجه لها علاقه بالصور فكك منها . بغض النظر عن الكود بتاع PEC الى ف اول الكومنت الى هيقره الفانكشن مش محتاج يعرف المعلومات الغامضه الى ف الكومنت خاالص.

```
/*
  RFC 2045 - Multipurpose Internet Mail Extensions (MIME)
  Part One: Format of Internet Message Bodies
  section 6.8.  Base64 Content-Transfer-Encoding
  The encoding process represents 24-bit groups of input bits as output
  strings of 4 encoded characters. Proceeding from left to right, a
  24-bit input group is formed by concatenating 3 8-bit input groups.
  These 24 bits are then treated as 4 concatenated 6-bit groups, each
  of which is translated into a single digit in the base64 alphabet.
  When encoding a bit stream via the base64 encoding, the bit stream
  must be presumed to be ordered with the most-significant-bit first.
  That is, the first bit in the stream will be the high-order bit in
  the first 8-bit byte, and the eighth bit will be the low-order bit in
  the first 8-bit byte, and so on.
*/
```

5.15- Inobvious Connection:

الصله بين الكومنت والكود الى بيوصفه لازم تبقا واضحه او على الاقل الى هيقره الكومنت ويبوص على الكود يبقا فاهم الكود بيتكلم عن انهى جزء ف الكود بالتحديد .

- وخد مثال عشان تفهم :

```
/*
 * start with an array that is big enough to hold all the pixels
 * (plus filter bytes), and an extra 200 bytes for header info
 */
this.pngBytes = new byte[((this.width + 1) * this.height * 3) + 200];
```

- ايه ال filter bytes الى ف الكومنت دى هل ليها علاقه ب ال 1+ ف الكود ولا الضرب فى تلاته ؟ولا ليه علاقه بالاتنين ؟وهل ال pixel ده هو byte؟ وليه مزود 200 ؟
- غرض الكود انه يوصف كود عشان الكود مش عارف يوصف نفسه ف حرام اصلا يبقا الكومنت نفسه عاوز الى يفسره .

5.16- Function Headers:

عم بوب بيقلك الفانكشن الصغيره مش محتاجه وصف يعنى فانكشن صغيره ليها اسم معبر وبتعمل حاجه واحده بس احسن من Function Headers

Function Headers دى معنها كومنت اول الفانكشن.

5.17- Javadocs in Nonpublic Code:

ده هطنشه دلوقتى عشان مش مفيد بالنسبالى ف حاجه . هرجعله ان شاء الله

بس كدة يا مؤمن خلصنا الحمد لله سلاااااااااام .