

メソッド名をコピーペーストすると問題が起きるのはどういう場合か

2020 年 11 月 7 日 新出尚之

課題のプログラミングにおいて、メソッド名やクラス名・フィールド名などを、手で打ち込まずに Javadoc や問題の解説文などからコピーペーストするのは、**一般には勧められます** (長い場合は特に)。**綴りミスを減らせる**からです。

しかし、11 月 6 日の課題では、メソッド名を **Javadoc** からコピーペーストで Eclipse に入力したことが原因と思われる問題が発生した人が多数出ました。

この文書は、そのとき何が起こっていたのかと、その対策に関するものです。

1 起こっていたこと

例えば、図 1 は 11 月 6 日の課題の headfirst.observer.distanceobserver.DistSensor クラスの Javadoc (配布されていたもの) をブラウザで見たところです。



図 1: DistSensor クラスの Javadoc

この画面の「registerObserver(Observer o)」をコピーペーストで **Eclipse** の DistSensor.java の編集画面にコピーして、それを加工して以下のような行を作り、

```
public void registerObserver(Observer o){
```

これをもとに registerObserver() メソッドを実装すると、画面上では registerObserver() メソッドが実装されているように見えるのに、実行しようとする「DistSensor クラスに registerObserver() メソッドが**実装されていない**」というエラーが起きてしまいます。

一方、同じブラウザ画面から「notifyObservers()」のところを同様にコピーペーストでコピーして、これをもとに notifyObservers() メソッドを実装しても、同じようなエラーは起きません。

2 原因

原因は、図1の画面で「registerObserver」と「(Observer o)」の間に、「ゼロ幅スペース」という目に見えない1文字が入っていることによります。

これにより、ブラウザ画面で「registerObserver(Observer o)」に見える部分をマウスでコピーペーストすると、実際には「registerObserver (Observer o)」がコピーされます。従って、「registerObserver」ではなく「registerObserver 」という名前のメソッドを定義したことになってしまうわけです(図2)。

図2はEclipse IDEのスクリーンショットで、左側の「すべてのメソッド」タブで「registerObserver(Observer o)」がリストアップされています。このメソッド名が赤い枠で囲まれています。右側のエディタには、このメソッドのソースコードが表示されており、同じメソッド名が赤い枠で囲まれています。赤い矢印と注釈が、元のソースコードに「ゼロ幅スペース」が含まれているが、ブラウザ表示上では見えないことを示しています。

修飾子とタイプ	メソッド	説明
int	countObservers()	登録されているオブザーバーの数を取得
float	getDist()	距離を取得
void	measurementsChanged()	監視している値が変わった
void	notifyObservers()	オブザーバーに計測値を知らせる
void	registerObserver(Observer o)	オブザーバーを登録する
void	removeObserver(Observer o)	オブザーバーを削除する
void	setMeasurements(float dist)	計測値を設定する

```
public void registerObserver(Observer o) {
```

図 2: コピーペーストの様子

「notifyObservers」とその次の「()」の間には「ゼロ幅スペース」が入っていないので、「notifyObservers()」をコピーペーストした場合は問題は起きません。

Javadoc を、ブラウザで見るのではなく、ソースの HTML ファイルを直接見てみると、その違いが明確になります。図1の画面の HTML ファイルは、配布されたプロジェクト内の doc/headfirst/observer/distanceobserver/DistSensor.html にあり、このファイルの中に以下のような部分があります(長い行を紙面の都合で分けています)。

```
<tr id="i3" class="rowColor">
<td class="colFirst"><code>void</code></td>
<th class="colSecond" scope="row"><code><span class="memberNameLink"><a href="#notifyObservers()">notifyObservers</a></span>()</code></th>
<td class="colLast">
<div class="block"> オブザーバーに計測値を知らせる</div>
</td>
</tr>
<tr id="i4" class="altColor">
<td class="colFirst"><code>void</code></td>
<th class="colSecond" scope="row"><code><span class="memberNameLink"><a href="#registerObserver(headfirst.observer.distanceobserver.Observer)">registerObserver</a></span>(<a href="Observer.html" title="headfirst.observer.distanceobserver内のインタフェース">Observer</a> o)</code></th>
<td class="colLast">
<div class="block">オブザーバーを登録する</div>
</td>
</tr>
```

「registerObserver」とその次の「(」の間に「​」というのがありますが、これが「ゼロ幅スペース」のHTMLでの表記です。一方、「notifyObservers」とその次の「(」の間にはそれがありません。

ちなみに、この「ゼロ幅スペース」はおそらく、Javadoc 生成ツールが、該当の箇所で画面上改行できるようにと、意図的に入れたものと思われます。ブラウザの横幅を縮めてみると、図 3 のように当該箇所が画面上で改行されるようになります。メソッド名と「(」の間に「ゼロ幅スペース」の入っていないところは改行が起きません¹。



図 3: ブラウザ画面の横幅を縮めたところ

同様の問題は、headfirst.observer.distanceobserver.FarDistWarning クラスの update() メソッドでも起きます。図 4 の画面で「update(float dist)」の「update」と「(」の間に、見えないゼロ幅スペースがあるので、これをコピーペーストで Eclipse の編集画面にコピーし、それを加工して update() メソッドの定義を作ると、実際には update ではなく「update ゼロ幅スペース」という名前のメソッドが定義されてしまいます。



図 4: FarDistWarning クラスの Javadoc

¹ ちなみに、日本語の部分は「ゼロ幅スペース」がなくても改行が起きます。

この場合、FarDistWarning クラスで update() メソッドが定義されなくても、その親クラスである CurrentDistDisplay で update() メソッドが定義されており、それが**継承される**ため、FarDistWarning クラスで「update() メソッドが定義されていない」というエラーにならないので、より問題が見つけにくくなります。

なお、11 月 6 日の課題の Javadoc では、他にも同様の問題が起きるメソッドがあります。




ちなみに、**Javadoc** 内でメソッド名の直後にゼロ幅スペースが入っている全ての箇所をリストアップするには、ターミナルを開いて、プロジェクトがあるディレクトリ (脚注 2 参照) に (あるいは、Javadoc があるディレクトリに) cd で移動し、以下のようにすればできます (紙面では 2 行に分かれていますが、実際は 1 行で打ち込んで下さい)。

```
$ find . -name \*.html -print0 | xargs -0 perl -ne 's!(\w*)(</a>|</span>)*&#8203;!  
print "$ARGV: $1\n"!eg' | sort -u
```

こうすると、Javadoc にゼロ幅スペースが入っている箇所が全くなければ何も出力されませんが、あれば、以下のような出力が得られます。「:」の左がファイル名 (「.html」を取り除くとクラス名になります) で、右が直後にゼロ幅スペースの入っているメソッド名です。

```
./doc/headfirst/observer/distanceobserver/CurrentDistDisplay.html: CurrentDistDisplay  
./doc/headfirst/observer/distanceobserver/CurrentDistDisplay.html: update  
./doc/headfirst/observer/distanceobserver/DistSensor.html: registerObserver  
./doc/headfirst/observer/distanceobserver/DistSensor.html: removeObserver  
./doc/headfirst/observer/distanceobserver/DistSensor.html: setMeasurements  
...
```

3 見分け方

この現象によって、**Java のソース内**に現にゼロ幅スペースが入ってしまっているかどうか見分けるには、Eclipse の編集画面で図 5 (本資料 p. 4) のように、怪しいと思うメソッドの名前と「(」の間にマウスカーソルを持っていき、カーソルキーで    と操作してみましょう。

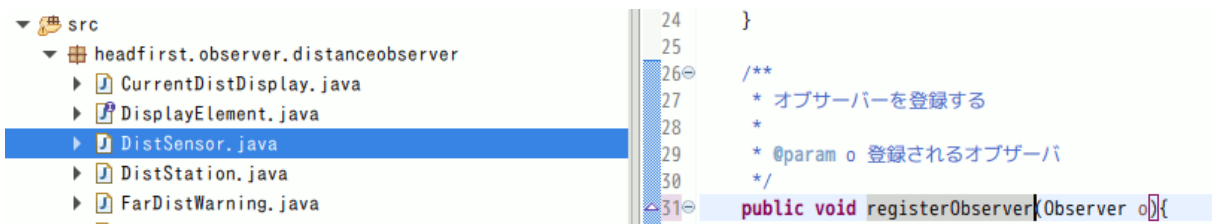







図 5: Eclipse 画面でゼロ幅スペースの有無を見分ける

そこにゼロ幅スペースがなければ、マウスカーソルは最初の  で「(」の右へ、その次の   で「r」(メソッド名の最後の文字) の左へ行くはずですが、もし、最初の  でカーソルが動かなかったり、あるいはその次に  を 2 度押してもカーソルが 1 度しか動かなかったりしたら、そこにゼロ幅スペースがあることになります。ゼロ幅スペースは 1 つの文字でしかも幅が 0 なので、カーソルを 1 文字ずつ動かしていくと、ゼロ幅スペースのあるところで 1 回カーソルが止まるように見えるのです。

不幸にしてゼロ幅スペースがあったなら、そのゼロ幅スペースだけを (普通に 1 文字削除するのと同じ操作で) 削除すればいいのですが、見えない文字なので消す操作がやりにくいし、消せたのかもわかりにくい (先述と同じ調べ方でわかるにはわかるのですが) ので、その 1 行を消して全部手で打ち直す方が安心できる解決策かもしれません。

今のところ、このケース (Javadoc 作成ツールがメソッド名とその次の「(」の間にゼロ幅スペースを入れてしまう) 以外に、同種の現象は観測されていないので、**メソッド名 (コンストラクタ名を含む) の直後**にだけ気をつけていれば大丈夫と思われます。

プロジェクト内の全ての **Java ソースファイル**について、ゼロ幅スペースが入っているかどうか一気に調べるには、(文字コードが UTF-8 であれば) ターミナルを開いて、プロジェクトがあるディレクトリ²に cd で移動し、以下のようなコマンドを実行すればできます (紙面では 2 行に分かれていますが、実際は 1 行で打ち込んで下さい。あるいはこの PDF からコピーペーストでも OK です)。

²大抵はホームディレクトリ下の Documents/workspace/プロジェクト名 というディレクトリだと思います。

```
$ find . -name \*.java -print0 |  
xargs -0 perl -ne 'print "$ARGV: $_" if s/\xe2\x80\x8b/@@@/g'
```

こうすると、ゼロ幅スペースが入っている Java ファイルがなければ何も出力されませんが、あれば

```
./src/headfirst/observer/distanceobserver/FarDistWarning.java:      public void  
update@@@(float dist){
```

のような出力が得られます (長い行を紙面の都合で分けています)。ゼロ幅スペースの入っている箇所は「@@@」に変換されて表示され³、「:」の左がファイル名です。

プロジェクト内の Java ファイルのゼロ幅スペースをコマンドラインで一挙に除去することも、一応可能ではあります。上と同様にターミナルを開いて、プロジェクトがあるディレクトリに cd で移動し、

```
$ find . -name \*.java -print0 | xargs -0 perl -i -pe 's/\xe2\x80\x8b//g'
```

とします。ただし、Eclipse で編集中にこれをやると、Eclipse の持っている編集内容と齟齬をきたして異常が起きる可能性があります (Eclipse はこの手の状況に極めて弱いのです)。また、Eclipse が動いていない間にやっても、その後立ち上げた Eclipse がこの操作で書き換わった Java ファイルを「書き換わっている」と認識してくれるか (そして必要時に自動的に再コンパイルしてくれるか) どうかわかりません。混乱が起きる可能性があるので、あまりお勧めはしません。開発に Eclipse のような IDE 類を使わず、Emacs などで開発するのなら、この手を使っても大丈夫でしょう。

4 予防

3 節で述べたように、この現象が観測されているのが今のところ Javadoc でしかもメソッド名の直後だけなので、コピーペーストするときにメソッド名の直後にだけ気をつけて、3 節に述べたカーソルキーを動かしてみる方法でペースト時に毎回確認すれば、おそらく防げるでしょう。あるいは、メソッド名をコピーペーストする際は名前の最後の 1 文字だけは手で打つなどの対策も有効かも知れません (引数リストをコピーペーストする場合はそちらの括弧の直前にも気をつけねばなりません)。

また、Javadoc 以外 (出題文など) からのコピーペーストではこの問題は起きないと考えられます。人間が編集時に意図的にゼロ幅スペースを入れることはあまり考えられないからです (特にテキストファイルでは)。

他の対策として、事前に **Javadoc 内の**ゼロ幅スペースを表す「​」を全部除去してしまうことも考えられます。それには、ターミナルを開いて、プロジェクトがあるディレクトリ (脚注 2 参照) に (あるいは、Javadoc があるディレクトリに) cd で移動し、このようにします。

```
$ find . -name \*.html -print0 | xargs -0 perl -i -pe 's/&#8203;//g'
```

こうしてからブラウザで閲覧すれば (すでにブラウザを開いていた場合はリロードする)、ゼロ幅スペースをコピーペーストすることによる問題は起きません。ただし、該当のところで自動改行ができなくなるので、ブラウザ画面の幅を小さくした場合の見栄えが変わってしまうことがあります。

なお、Javadoc を作り直した場合は、この操作をやり直す必要があります。

³同じ行に最初から「@@@」が入っている箇所があればそこも「@@@」と表示されてしまいますが、そのようなことはまずないでしょう。