

7/23/2022

Clean code

Ch2 Meaningful Names

Meaningful Names

by Tim Ottinger



Activate V

Abdurrahman Gamal Ahmed Gaber
NO COMPANY

Table of Contents

1-Introduction.....	2
2-Use Intention-Revealing Names.....	2
3-Avoid Disinformation :.....	4
4-Make Meaningful Distinctions.....	4
5-Use Pronounceable Names	5
6-Use Searchable Names	5
7-Avoid Encodings.....	6
7.1-Hungarian Notation:	6
7.2-Member Prefixes :.....	6
7.3- Interfaces and Implementations:	6
8-Avoid Mental Mapping	6
9-Class Names	6
10-Method Names	6
11-Don't Be Cute	7
12-Pick One Word per Concept.....	7
13-Don't Pun.....	7
14-Use Solution Domain Names	7
15-Use Problem Domain Names	7
16-Add Meaningful Context.....	7
Words I do not know in this chapter:.....	10

1-Introduction

الاسماء موجودة ف كل مكان ف ال SW . Variables سيكون ليها اسم وكمان ال Function, class, arguments, package, source files and the directories . ف عشان احنا بنستخدم المتغيرات كثير ف محتاجين نستخدمهم بشكل كويس , عشان كدة هنشوف بعض القواعد لتسميه ال variables . وصورة الغلاف بتحاول تقلك ان تسميه ال Variables او التسميه عموما ف ال SW زى باضبط انك تختار اسم لابنك .

2-Use Intention-Revealing Names

يعنى اختار اسم اول مشوفه اعرف ده بيستخدم ف ايه . وده ممكن تفكره حاجة سهله ف تابع معنا وهنشوف سهل ولا لا . وطبعاً انك تحدد اسم مناسب ده هياخد وقت بس صدقنى مش هيكون اكثر من الوقت الى هياحفظ عليه بعد كدة لما تحب تفهم الكود او تعمل عليه تعديلات .

الاسماء الى بنختارها لازم تكون بتجاوب عن 3 اسئله :

1. ليه ال " variable او ال class .. الخ " ده موجود ؟
2. هو بيعمل ايه او ايه الوظيفة بتاعته ؟
3. ازاي اقدر استخدمه ؟

ولو الاسم بيحتاج comment كدة الاسم ده مش بيعكس النيه name does not reveal its intent . يعنى مفروض الاسم ميكونش معاه .comment

تعاله ناخذ كام مثال :

EX 2.1: `int d; // elapsed time in days`

هنا الاسم d مش بيعبر او مش بيعكس اى حاجة ف هو مش بيعبر عن ال day ولا حتى بيعبر عن الوقت الى فات . ف لما تختار اسم ف الحالة مفروض كان يعبر عن

1. الحاجة الى هتقسها .
2. وحدة القياس .

ممكن كنت تسميه اى حاجة من الاسماء ده : والاسماء ده بتعبر عن الحاجة الى هتقسها الى هي الوقت ووحدة القياس الى هي يوم او سنه .

```
int elapsedTimeInDays;  
int daysSinceCreation;  
int daysSinceModification;  
int fileAgeInDays;
```

EX 2.2: What is the purpose of this code?

```
public List<int[]> getThem() {  
    List<int[]> list1 = new ArrayList<int[]>();  
    for (int[] x : theList)  
        if (x[0] == 4)  
            list1.add(x);  
    return list1;  
}
```

ف بطل هبد كل الى قلته غلط ⑤ : الكود مش واضح هو بيعمل ايه او ايه الغرض منه مع العلم ان هو كود بسيط خالص . ف الفكرة هنا مش ف ان الكود صعب ولا سهل الفكرة ف التفاصيل الى مخفيه . طيب ايه الى محتاج تعرفه عشان تعرف عشان تفهم الكود او ايه الاسئله الى جات ف بالك اكيد سئلت نفسك ايه الصفر ده اشمعنى الصفر طيب الاربعه ده بتعبر عن ايه وال theList كمان بتعبر عن ايه والي list1 لانه لما ترجع هعمل بيها ايه صح ! ؟ هي فعلا ده الحجات الى محتاجها الكود عشان تفهمه تعاله نصلح ف الكود سيكا .

1. What kinds of things are in theList?
2. What is the significance of the zeroth subscript of an item in theList?
3. What is the significance of the value 4?
4. How would I use the list being returned?


عشان اقدر اعدل الكود لازم تفهم الحاجة الى هتعملها كود ف الكاتب شرح الحاجة الى هعملها كود المهم هي لعبة mine sweeper game



- ودة شكل اللعبة. ف هو بيقلك ان ال board الى قدامك دي عبارة عن cell واحنا سمينها ف الكود الى فات list تعاله نغير اسمها ل game Board.
- وذلك ان كل cell هي عبارة عن Array.
- المكان رقم صفر ف ال array بكون فيه ال status values.
- والرقم اربعة ده بيعبر عن ان حد كتب على المكان المكان ده flagged. يعني لما يكون عندي $X[0] = 4$ " الى ف الكود الى فات" دي معناها ان ال cell الى انا فيها حالينا حد كتب عليها او علم عليها يعني ال cell دي flagged

بعد مقتللك الكام معلومه دول تعاله نكتب الكود تاني ونشوف هل لما نغير الاسماء شويه بناء على الى فهمتها هتفهم الكود ؟

```
public List<int[]> getFlaggedCells() {
    List<int[]> flaggedCells = new ArrayList<int[]>();
    for (int[] cell : gameBoard)
        if (cell[STATUS_VALUE] == FLAGGED)
            flaggedCells.add(cell);
    return flaggedCells;
}
```

بصراحه الكود كدة بقى احسن ومعبر بشكل عظيم . وانا شايف حتى ان مش محتاج تعليق او توضيح هو بيعمل ايه بسيط . بالك بقا لو علمنا class اسمه Cell بدال مبنستخدم array of int وعملنا فيه كمان داله اسمها isFlagged عشان نخفي الارقام او عشان نحوش جزء من ال logic بتاع الكود زي  هي الارقام دي ليه ومنشغلش بالنا بيها دلوقتي ودة هيساعد ف فهم احسن .

فيبدال مكنا عمليين constant flagged=5 ف ال main class كدة هنعمله جوة class تاني ف كدة خفيينا الارقام ي صحبي . وبصراحه الكود كدة بقى عظمه بدون منعمل اي تغيير ف الكود مجرد غيرت الاسماء ودي بقا الفائدة من الاسماء المعبرة .

ودة اخر شكل للكود .

```
public List<Cell> getFlaggedCells() {
    List<Cell> flaggedCells = new ArrayList<Cell>();
    for (Cell cell : gameBoard)
        if (cell.isFlagged())
            flaggedCells.add(cell);
    return flaggedCells;
}
```

3-Avoid Disinformation :

تجنب المعلومات المضللة. يعني لازم تتجنب الاسماء الى ممكن تدينا معنى غير الى عوزينه .

EX-3.1

لو عندك مجموعه من ال Accounts مينفعش تسميها مثلا accountList عشان المبرمجين عندهم مفهوم عن ال list معين زى ان ال list او ال array هي مجموعه من العناصر من نفس النوع وموجدين وري بعض ف ال memory. ف ربح دماغك يصحبي وسميها accountGroup او bunchOfAccounts او accounts.

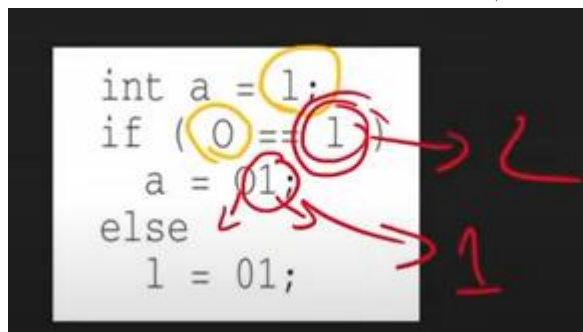
وعما بوب بيقلنا معلومه جامدة زحليقه : ان حتى لو الحاجة فعلا list ف مش مفضل تسميها AccountList يعني مش مفضل ان اكتب ال Container type ف الاسم .

EX-3.2

وعما بوب برضو بيقلك يسطا متستخدمش اسماء متغيرات زى hp, aix, and sco عشان دول اصلا اسماء موجودة ف Unix platforms.

EX-3.3

ومن اسواء الحجات الى ممكن تشوفها انك تستخدم حرف ال lower-case L او تستخدم حرف ال uppercase O. عشان حرف ال lower-case L ده شبه رقم واحد وال uppercase O شبه رقم صفر.



وانا راضى زمك ياخي فاهم حاجة من ام الكود ده . كذاب يسطا مش مفهوم الكود معفن . ومش واضح لو الى الصورة ده واحد ولا حرف. وعما بوب يقول ان الكود ده هو شاف حد كاتبه . ف انا حسيت ان عما بوب شويه وهيحسن ف الى كتب الكود ده .

4-Make Meaningful Distinctions

-اوقات المبرمج بيعمل مشاكل لنفسه لما يكتب كود عشان مجرد يشتغل .يعني هسمى اى حاجة عشان اخلى الكود يحصله compile . على سبيل المثال لو انت عندك scope معين variable ف انت مينفعش تكرر نفس الاسم تاني ف بضيف حرف او بتحذف حرف او ممكن بتزود رقم ف ده حاجة مش كويسه . لو لازم الاسماء تبقا مختلفة يبقا لازم معناها يبقا مختلف كمان. ف اسماء زى ... a1, a2 دى مش اسماء مضللة بس هي ملهاش معنى اصلا .

-وف حاجة اسمها Noise words: دى مكلمات فيها اختلافات بلا معنى .يعني كلمتين مختلفين بس معنهم واحد . تخيل ان عندك product class وعندك كلاس تاني اسمه productData او كلاس اسمه productInfo ف لو انا بقرة الكود مش هعرف الفرق بين ال class دى ومش هعرف استخدمها كل واحد فيهم امتى ف دى Noise words .

-وعيب انك تظهر كلمة variable ف اسم اى متغير عندك ، وبيقلك مثلا ازاي variable اسمه nameString احسن من variable اسمه name ؟يعني عمرك لقيم اسم حد float number ف ملهوش لزمه تكتب كلمة string . ومثال تاني لو عندك customer class و customerObject class ايه الفرق بينهم ؟ انهى واحد ممكن الاقى فيه معلومات عن customer's payment history.

-لو عندك التلت دوال هتعرف تعمل call لانهى واحدة فيهم امتى ؟

```
getActiveAccount();
getActiveAccounts();
getActiveAccountInfo();
```

الاسم ف الظاهر مختلف بس المعنى بتاع كل واحدة فيهم مش مختلف. ف الاسم مفروض يدلك تستخدم امتى اى واحد ف دول .

5-Use Pronounceable Names

لازم تستخدم اسماء تعرف تنطقها عشان تعرف تتناقش فيها . تخيل انك بتكلم واحد زميلك وبتقله شوف ال variable الى اسمه abstsntfId مش هتتعرف تنطق الاسم اصلا عشان تتناقش فيه . ف البرمجه نشاط جماعى لازم هتتناقش ف الكود مع حد من زميلك .

عنا بوب ببدينا مثال : ان ف شركه عنها متغير اسمه genymdhm وده اختصار ل

" generation date, year, month, day, hour, minute, and second " لو حابب تتناقش ف مشكله تخص المتغير ده هتعتبر عنها ازاي صحبى ؟ مش هتتعرف تنطق الاسم وهتكون مش موضع سخرية بسيط .

ف	شوف	الفرق
اول حاجة اسم الكلاس معبر خالص وصريحه		
تاني حاجة بدال كلمه genymdhm خلتها حاجة		
تعرف تنطقها generationTimestamp وانت		
بتتناقش ف الكود . وكم ان الاسم معبر		

```

class DtaRcd102 {
    private Date genymdhms;
    private Date modymdhms;
    private final String pszqint = "102";
    /* ... */
};

```

to

```

class Customer {
    private Date generationTimestamp;
    private Date modificationTimestamp;;
    private final String recordId = "102";
    /* ... */
};

```

6-Use Searchable Names

- عنا بوب ببقلك الاحسن نستخدم اسماء للمتغيرات يكون سهل البحث عنها . يعنى متستخدامش كلمه من حرف واحد او رقم مثلا ده هيكو صعب البحث عنه . وبقلك الاسماء الطويله المعبره احسن من الاسم القصير الغير معبر . وهنا لو الاسم قصير بس معبر قل استخدمه .
- وممكن تستخدم اسم متغير من حرف واحد لو كان المتغير ده local .
- ولو هتستخدم constant او اى متغير هتستخدمه كتير ف الكود الاحسن تبديله اسم يكون سهل البحث عنه .
- وده مثال : هحول الكود الاول للكود التانى وواضح ان الكود التانى سهل ف القراءه اكتر وكم ان سهل ابحت عن اى متغير .

<pre> for (int j=0; j<34; j++) { s += (t[j]*4)/5; } </pre>	<p>بس يصحبنى هو غير كل الارقام وخلصها</p> <p>Constant وده عشان تسهل عليه البحث</p>
---	--

to

<pre> int realDaysPerIdealDay = 4; const int WORK_DAYS_PER_WEEK = 5; int sum = 0; for (int j=0; j < NUMBER_OF_TASKS; j++) { int realTaskDays = taskEstimate[j] * realDaysPerIdealDay; int realTaskWeeks = (realdays / WORK_DAYS_PER_WEEK); sum += realTaskWeeks; } </pre>	<p>وكم ان عشان يسهل فهم الكود وكم ان عشان</p> <p>الاسماء بيقا معبر اكتر .</p> <p>الكود فيه غلط من الكاتب "من وجه نظرى"</p> <p>اكتشفها انت وهى غلط مشمهمه .</p> <p>اسم مكتوب مش ف مكانه</p>
--	--

7-Avoid Encodings

حاجة مش كويسه خالص انك تكتب كود بطريقه خاصه او طريقه مشفره . وعما بوب بيقول بالانجليزى "احنا نقصين يجدها هم مش كفايه الكود" وانا حسيته ببصيح الصراحه.

7.1-Hungarian Notation:

والسكشن ده حلو الصراحه.

Hungarian -- < ده اسم دوله المجر والسكشن ده باسم الدوله عشان الدوله دى كانت بتعكس السماء تقدر تبحث عن الموضع لو حابب . المهم عما بوب بيقولك زمان كان ال compiler مش بيعمل check لنوع المتغيرات ف عشان كده المبرمج كان بيضيف كام حرف عشان يعرف نوع الحاجه . زى مثلا strName ف انا ضفت str عشان افترق نوع المتغير ف ده الى اسمه Hungarian Notation . بس بعد كده الدنيا الدنيا اطورت وبقا مفيش داعى اكتب ال HN .

7.2-Member Prefixes :

مش محتاج اكتب اى Prefixes عشان تدل على نوع ان المتغير ده هو member in class ولا لا . عشان ال IDEs الحديثه بتخلي ف اللوان مختلفه لكل متغير .

```
public class Part {  
    private String m_desc; // The textual description  
    void setName(String name) {  
        m_desc = name;  
    }  
}
```

وده مثال : الكود الى ف الاول ده عامل m_dec

وده ال m اختصار ل member variable . واحنا

مش محتاجين لل Prefixes ده ولا غيره مع ال

IDEs الحديثه.

```
public class Part {  
    String description;  
    void setDescription(String description) {  
        this.description = description;  
    }  
}
```

7.3- Interfaces and Implementations :

ده ليه علاقه بال design pattern ف هنرجعه تانى ان شاء الله بعدين .

8-Avoid Mental Mapping

والسكشن ده كويس . وبيتكلم عن انك متستعرضش قضاياتك وتتفلك يلا . وغالبا المبرمجين بيكونوا ازكيا .

فى فرق واحد بين المبرمج professional و المبرمج الزكى : المبرمج ال professional بيكون المهم بالنسباله يكتب كود واضح عشان غيره يفهمه وده المهم . اما المبرمج الزكى ده بيكون فزلكه وده غلط . ف الوضوح اهم من انك تبين نفسك شاطر .

9-Class Names

اسم ال class او ال object الاحسن يكون اسماء مش فعل زى Customer او Account.

10-Method Names

-اسم ال function بيكون فعل زى save , postPayment, deletePage .

-عما بوب بيقول لو انت هتعمل constructors overload الاحسن انك تستخدم static factory methods ويكون ليها اسم معبر عن arguments ال method هتخدها . شوف المثال عشان تفهم

مثال : ف الصور الى تحت دى ببوا الصحاب بدال معمل constructors ياخذ قيمه float انا عمل static method واسمها اسم معبر وده احسن بكثير من ان اعمل constructors.

`Complex fulcrumPoint = Complex.FromRealNumber(23.0);`

is generally better than

`Complex fulcrumPoint = new Complex(23.0);`

11-Don't Be Cute

- عمنا بوب بيقلك متروش ف الكود عشان والله احنا مش نقصين بيني . يعني لو عندك `method` متسمهاش `HolyHandGrenade` معنى الكلمه دى قبله اليد المقدسه والاحسن تسمى ال `method` مثلا `DeleteItems` .
- متستخدمش لغه الشارع "لغه عاميه" .
- بعد كدة عمنا بوب بيقلك قول الى تقصده واقصد الى تقوله وخليك واضح يعنى .

12-Pick One Word per Concept

- اختار كلمه واحده لل `Concept` . يعنى مينفعش تستخدم الاسماء دى `fetch, retrieve, get` ف `class` مختلف مع انهم هيعمله نفس الحاجة كلهم مثلا هيرجه `object` الاحسن تلتزم باسم واحد . كل الدوال الى هترجع `object` تبتداء ب `get` مثلا . عشان مش هتعرف انهي `method` تتبع انهي `class` . حتى لو ال `IDEs` بيظهرك كل `method` لما تكتب `dot` بعد اسم ال `object` ف الى هيطهر هو اسم الداله بس من غير اى `comment` انت كاتبه او اى معلومه . تانى تساعدك عشان تفترك الداله بتعمل ايه .
- لو انت مسمى `object` مثلا `controller` و مسمى `object` تانى مثلا `manager` او `driver` ! لو كل ال `object` دى من نفس ال `class` الى هيقرة الكود بعدك مش هيعرف او مش هيكون واضح بانسياله ان كل ال `object` دى من نفس الكلاس .

13-Don't Pun

- متستخدماش كلمه ليها معنيين . والترجمه الحرفيه انك متعملش توريه بالكلام .
- تخيل ان عندك داله اسمها `add` ف اكثر من `class` : من اسمها هي بتعمل جمع لقميتين او بتعمل `concatenating` لحاجتين .
- تخيل بقا ان عندك `class` تانى جواه داله و الداله دى بتاخذ `parameter` واحد وبضيفه ف `list` مش صح خالص انك تسمى الداله دى `add` . دة اسمه `pun` يعنى توريه ولعب بالكلام يصحبي . والصح تسمى الداله دى `insert` مثلا . ف كلمه `add` هنا هتلتبظ والهدف بتاعنا اننا نخلى الكود سهل واضح قدر المكان .

14-Use Solution Domain Names

- الناس الى هتقرا الكود هم مبرمجين . ف هم فاهمين مصطلحات ال `(CS) computer science` .
- ف المبرمج فاهم المصطلحات الى موجود ف `algorithm names, pattern names, math terms` . ف استخدم المصطلحات دى براحتك وبلاش تستخدم مصطلحات ليها علاقه ب ال `problem domain` عشان اى حد هيقرة الكود بعدك هيتحتاج يتواصل مع ال `customer` عشان يفهم الكود .
- ف عمنا بوب بيقصد ب `Use Solution Domain Names` : يعنى تستخدم اسم الحاجة " `design pattern` " الى هتحل بها المشكله .
- وملاحظه على جنب كدة ال `design pattern` دى زى حلول جاهزة لمشكله مشهوره او ا تكررت كثير .

15-Use Problem Domain Names

- السكشن دة عكس السكشن الى فات : هنا بيقلك لو الكود بتاعك هيقرأه شخص مش مبرمج ف استخدام مصطلحات تعبر عن المشكله مش عن الحل .
- وانك تعرف تفصل او تميز بين ال `Problem Domain` و `Solution Domain` دة جزء من دور المبرمج الشاطر .

16-Add Meaningful Context

- من الاحسن انك تخلى الاسماء قريه لمجال او استعاب الشخص الى هيقرة الكود . ولو فشلت ف انك تختار كلمات معبرة او مفهومه لشخص الى هيقره الكود . ف استخدام ال `prefixing` . تعاله ناخذ مثال على موضوع ال `prefixing` :

تخيل ان معايه ان معايه variables زى دول :

firstName, lastName, street, houseNumber, city, e state, and zipcod

لو شوفت ال variables دول كلهم مع بعض هتفهم ان دى متغيرات متاخدة او بتعبر عن address . ف انا فهمت من السياق
" Context " ان دة address . اما لو شوفت مثلا المتغير الى اسمه state موجود لوحدة ف method هتفهم منه حاجة ؟ اكيد مش هتفهم
ان دة جزء من address . ف عشان نحافظ على ال Context هنضيف prefixes معينى . فحللتنا هنا هيكون addr . ف المتغيرات هتبقا كدة :
addrFirstName, addrLastName, addrState وكدة لو ظهر اى متغير لوحدة ف اى method هيكون معروف دة تبع ايه.
وطبعا الحل الاصح انك تعمل class اسمه address ووف الحالة دى حتى ال compiler هيعرف ان المتغيرات دى تبع concept اكبر
تعاله ناخذ مثال ببو الصحاب ازاى نخلي الكود اوضح او نظبط ال context :

Listing 2-1

Variables with unclear context.

```
private void printGuessStatistics(char candidate, int count) {
    String number;
    String verb;
    String pluralModifier;
    if (count == 0) {
        number = "no";
        verb = "are";
        pluralModifier = "s";
    } else if (count == 1) {
        number = "1";
        verb = "is";
        pluralModifier = "";
    } else {
        number = Integer.toString(count);
        verb = "are";
        pluralModifier = "s";
    }
    String guessMessage = String.format(
        "There %s %s %s", verb, number, candidate, pluralModifier
    );
    print(guessMessage);
}
```

طبعا اسم ال method بيكون جزء من ال context وكمان الكود او الالجورزم بيكون جزء من context . ف method هنا كبيرة شويه
او فيها logic كتير مش هتفهم الغرض من الداله من النظرة الاولى .

ف احنا علوزين نظبط ال function دى عشان تكون واضحة اكثر او عشان يكون ال context واضح لكل الاجزاء.

- 1- اول حاجة مفروض اعمل class واحط فيه ال 3 variables عشان اوضح ال context لل variables واسم الكلاس هو مثلا
GuessStatisticsMessage
- 2- هقسم الالجورزم دة لمجموعه من method الصغيرة .

ف الكود الى هتشوفه تحت طبقنا الخطوتين الى فوق دول :

-عملنا class الى قلت عليه وكمان عملت 5 method . غيرت اسم الداله من printGuessStatistics ل createPlura.. وخليت
الكود الى جوة كل condition داله كامله .

-عملت داله اسمها make وناديت جوها الداله createPlura.. وبعد كدة رجعت الاسترنج الى اتكون من داله... createPlura..

موضع ازاى غيرت الكود دة انا شايفه مهم خالص ف ركذ ف كل جزء فيه .

ليه عملت داله اسمها make ؟ عشان احنا دلوقتي ف class واحنا عملنا داله make عشان نطبع فيها بدال ما كنا بنطبع ف اخر الداله ف
هنا مش هينفع نطبع ف اخر ال class ف حطينا سطر الطباعة ف داله . وكمان عشان يكون كل اسم داله معبر عن تاسك صغير هو بيعمله.
فمثلا داله make بتعمل استرنج ودى تاسك معبر ومناسب للاسم . وباقي الدوال نفس الفكرة ؟ اخر كام سطر هبد منى مش ف الكتاب

Variables have a context.

```
public class GuessStatisticsMessage {
    private String number;
    private String verb;
    private String pluralModifier;

    public String make(char candidate, int count) {
        createPluralDependentMessageParts(count);
        return String.format(
            "There %s %s %s%s",
            verb, number, candidate, pluralModifier );
    }

    private void createPluralDependentMessageParts(int count) {
        if (count == 0) {
            thereAreNoLetters();
        } else if (count == 1) {
            thereIsOneLetter();
        } else {
            thereAreManyLetters(count);
        }
    }

    private void thereAreManyLetters(int count) {
        number = Integer.toString(count);
        verb = "are";
        pluralModifier = "s";
    }

    private void thereIsOneLetter() {
        number = "1";
        verb = "is";
        pluralModifier = "";
    }

    private void thereAreNoLetters() {
        number = "no";
        verb = "are";
        pluralModifier = "s";
    }
}
```

17 —Don't Add Gratuitous Context

مضيف context بلا سبب . يعنى متكبرش اسم ال variable على الفاضى . لو اسم ال variable صغير ومعبر خلاص .

مثال لو عندى مثلا instance اسمه accountAddress او customerAddress من كلاس اسمه address. كدة زى الفل مش محتاج تخلى ال class اسمه accountAddress او customerAddress .

18-Final Words

اخبرنااا نهاية الشبتر ي صحبى . عما بوب ببفلك انك تعرف تسمى ال variable او ال class او اى حاجة ثانى عموما دى مهارة مع الوقت هتكتسبها ان شاء الله .

Words I do not know in this chapter:

1.	Names should reveal intent.	يكشف النية
2.	What we want to impress upon you	ما نريد اقناعك به
3.	evoke a sense of elapsed time	تثير الاحساس باوقت المنقضى
4.	he significance of the value 4	دلاله الرقم الرابع
5.	avoid leaving false clues that obscure the meaning of code	تجنب ترك أدلة خاطئة تحجب معنى الكود
6.	Beware of using names which vary in small ways.	احذر من استخدام الأسماء التي تختلف بطرق بسيطة.
7.	How long does it take to spot the subtle difference between	كم من الوقت يستغرق لاكتشاف الفرق الدقيق بين
8.		
9.		
10.		
11.		
12.		
13.		
14.		
15.		
16.		
17.		
18.		
19.		
20.		
21.		
22.		
23.		
24.		
25.		