Style

It's still hard to give rules, but we can give some examples.

Remember: this is to some extent a matter of taste.

One rule: use names

- If you name something – a quantity, a concept, a type – then you can avoid having to understand it more than once.

```
System.out.println(2 + 3);

System.out.println("2 + 3 = " + (2 + 3));

System.out.println("2 + 3 = " + (2 + 2));

int firstNum = 2;
int secondNum = 3;
System.out.println ("2 + 3 = " + (firstNum +
   secondNum));

System.out.println (firstNum + " + " + secondNum +
   " = " + (firstNum + secondNum));
```

Choosing names

Not too short

- i, j are acceptable only as loop indices
- similarly e for Enumerations
  - but many would use "enum" or longer

Not too long

- numberOfCurrentlyEnrolledUndergraduateStudents

Should a field be "public static final"?

```
public static final int STUDENT_NUMBER_LENGTH = 9;
```

Why is it public?

- Many parts of a program might want to know how long a student number must be.

Why is it static?

- The value isn't dependent on a particular object: all students share it.

Why is it final?

- No part of any program should change the length of a student number.

Should a public field be a method?

```
public static boolean isValidStudentNumber (...)
```

- We can check more than the length.
- We can handle a situation in which checking the length itself may no longer be appropriate.

So even if a field satisfies all the rules for being public, static and final, maybe it shouldn't be a field at all.

Comments 1

```
// Constructor for case 1
public Student (String name, String stunum) {...}
```

```
// Variable declarations
```

```
// public fields
```

This is actually a useful comment. Why?

```
// the sellTickets method
public void sellTickets (...) ...
```

```
// sellTickets: subtract numTickets from the
// number of tickets remaining
public void sellTickets (int numTickets) ...
```

Comments 2

This one's OK:

```
// sellTickets: record the sale of numTickets
// tickets
public void sellTickets (int numTickets) ...

// getTickets: return the number of tickets
// available
public void getTickets (int numTickets) {
  ticketsLeft -= numTickets;
}

int maxTickets = 100; // number of tickets left
```

Comments

Write your comments before your code.