Naming Variables

For each variable name, decide whether it is valid or invalid. If it is valid, decide whether it also obeys the conventions of Java programming. If it is invalid, try to think of a better name for the variable. If you don’t recall, these are the rules & conventions for Java variable naming:

1. Variable names begin with a letter, a dollar sign $, or an underscore _, and are followed by any sequence of alphanumeric characters (letters and numbers). Even though they *can* start with $ or _, it is not conventional to do so. The dollar sign $ is definitely never used at all by convention, but the underscore _ is sometimes used for private variables, but we haven’t talked about that yet.
2. Whitespace is not permitted
3. Cannot be named the same as any keywords or other reserved identifiers. See here.
4. Variable names should be short yet meaningful. The choice of a variable name should be mnemonic- that is, designed to indicate to the casual observer the intent of its use.
5. One-character variable names should be avoided except for temporary "throwaway" variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.

1. myName
2. pi (the constant, 3.1415…)
3. e (euler’s number, 2.71828...)
4. isItOn
5. numberOfChickensInTheFarm
6. active
7. total_sales
8. car deliveries
9. studentNames
10. HoursOfPractice
11. length
12. class

Solutions

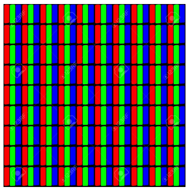
1. Valid & obeys conventions
2. Valid & obeys conventions
3. Valid & obeys conventions. This is one exception to a time when a legitimate variable (as opposed to a throw-away variable) should have a one-letter name. Since the variable refers to Euler's constant, which is commonly written as 'e' in math notation. Another exception would be naming the variables r, g, and b for the red, green, and blue components of an RGB pixel.
4. Valid, but not conventional. A better choice would be isOn. Try to not use 'is', but definitely never use a pronoun like 'it' in your variable name.
5. Valid, but too long. A better choice would be numChickens.
6. Valid & obeys conventions
7. Invalid, underscores aren't allowed anywhere but the beginning of the variable name
8. Invalid, whitespace is not allowed
9. Valid & obeysConventions
10. Valid, but not conventional. Variable names should be in camelcase, but should have a lowercase first letter.
11. Valid & obeys conventions (Even though 'length' is the name of a String method, you can use it as a variable name. This is because it is not a keyword, but merely the name of a method).
12. On the contrary, 'class' is a keyword and cannot be used to identify a variable

Using the right type

You should make sure to use the right primitive type. Java providers several built-in types for representing the same modality, but the use case differs for each one. One reason is that using an unnecessarily large type wastes storage. This may not make much of a difference at the lower level, but when dealing with tons of data this becomes crucial. Also, you need to use the right primitive type to over unexpected truncation and numerical overflow. These can be serious logical bugs that are hard to pick out.

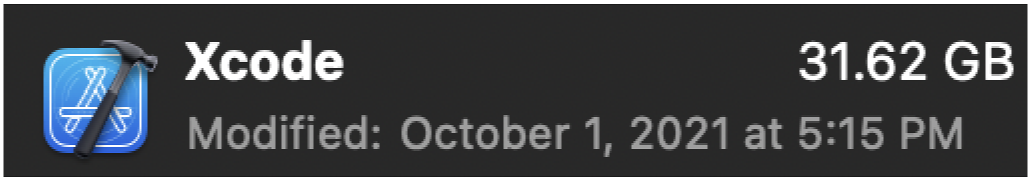
For each of the following variable names, decide which primitive type best suites it. Note that these are all valid and conventional variable names, so that part is not impotant in this activity. Answers can be one of the 7 primitive types or String.

1. worldPopulation
2. totalDonations
3. budget
4. serviceHours
5. myTestScore (assume 0-100 scale, with no fractional points)
6. averageTestScore
7. r (stores the value of the red color component of a pixel)



8. sqrt2 (stores the square root of 2, which is irrational: 1.414….)
9. weight (imagine a standard bathroom scale)

10. fileSize (measured in **bytes**, not KB, MB, or GB, think of the storage size of a large desktop application)



Xcode 31.62 GB

Modified: October 1, 2021 at 5:15 PM

11. fullName
12. middleInitial

Solutions

1. long; this is too big to fit in int
2. int, if you are measuring in just dollars, or float, if you record cents, too.
3. Again, int or float
4. short
5. byte
6. float
7. byte
8. double
9. int or float
10. long
11. String
12. char