Kill Useless Code:

Under developers it is sometimes common practice to leave code without a task in thesource file. Usually this is also code found by Coverage Tools and cannot be found by anytest.

Meaningful Names:

Names of the classes, variables, and methods must be meaningful and clearly indicate what a method does or what an attribute is.

Create pronounceable names to facilitate communication.

Delete Useless Comments:

- One of the most common reasons for the comments is because the code is bad.
- If you're thinking about writing a comment, then the code should be refactored.
- Comments do not save a bad code.
- Try to explain what the code causes to happen.
- Comments can be useful when placed in certain places.

Invest in a Precise Naming:
This is not only about variables, but also about fields, methods, interfaces, classes orpackages. Here too, it should be borne in mind that•the code is mostly read and therefore has to transport as much information andcontext as possible.•On the other hand, however, it must not be too long, so that manual refactoring orenhancements do not take up too much time or make the line unreadably long.

Explanatry variables:

If there are two different things in the same scope, you might be tempted to change one name in an arbitrary way. What is stored in cust, which is different from customer?

example:
String cust;

String customer;

Keep it simple, stupid (KISS):
KISS is a term also used in developing solutions to programming problems. Literally translated, KISS means “keep it simple, stupid” or “keep it stupid simple“. However, it’s assumed, the statement was not supposed to sound negative, rather to suggest
a simple design makes service and maintenance child’s play.

Don´t Repeat Yourself (DRY):

Don't repeat yourself (DRY, or sometimes do not repeat yourself) is a principle of software development aimed at reducing repetition of software patterns, replacing it with abstractions or using data normalization to avoid redundancy.

Avoid pointless function naming:

Obviously, the production code should never contain functions named without strict meaning. Or named just for the sake of naming. Some examples of pointless function names: foo() , bar() , baz() , buzz() - without reasonable sense.

Avoid negative conditions:
Negatives are just a bit harder to understand than positives. So, when possible, conditionals should be expressed as positives. For example:
if (buffer.shouldCompact())
is preferable to
if (!buffer.shouldNotCompact())