There are a number of things that can cause a loop to iterate infinitely. Generally, this is undesirable and an unintended consequence of something else. Often times, this is the result of a loop that never closes, either due to a condition never being met or there being no end condition to a particular statement.

The effects of an infinite loop are usually a “game breaker” in an application. They often prevent the user from further using the application. This may be a result of a user entering improper information which was not anticipated by the developer, or a logic error written by a developer. Either way, it is usually the developer’s fault.

Below is an example of an infinite loop.

public class infiniteLoop {

public static void main(String[] args) {

    int i = 0;

// there is nothing to increment our int so it runs infinitely

while (i < 5) {

    System.out.println("Will this goes on forever?");

}

}

}

In this example “i” stays at zero because there is nothing to change its value, thus causing an infinite loop.

Below is an example of the same code but we added an increment feature that will eventually close the loop after 5 iterations.

public class infiniteLoop {

public static void main(String[] args) {

    int i = 0;

// there is nothing to increment our int so it runs infinitely

while (i < 5) {

    System.out.println("Will this goes on forever?");

    //to fix this problem we can increment our variable

     ++i;

}

}

}

It’s worth noting that this is a very simple example and there are other, more complex situations, that will cause an infinite loop in a less obvious way. This is why constant testing is critical in the development process.