

# Theory: Errors in programs

⌚ 8 minutes

5 / 7 problems solved

Start practicing

8740 users solved this topic. Latest completion was 20 minutes ago.

Suppose, you're writing a Java program. Different errors may occur during its compilation or execution. We will divide all possible errors into two groups: **compile-time errors** and **run-time errors**.

Let's look at cases where errors occur and how to avoid them.

## §1. Compile-time errors

Compile-time errors are errors that prevent a java program to compile:

- a syntax error: incorrect keyword, a forgotten symbol `;` at the end of a statement;
- a bad source code file name;
- invoking a non-existing method;
- and many others.

Consider an example of compile-time errors. The following program should output the string "Hello!" but it does not compile.

```
1 public class MyClass {
2
3     public ztatic void main(String args[]) {
4         System.out.println("Hello!");
5     }
6 }
```

There are two errors in this program:

- a typo in the keyword `ztatic`;
- incorrect name of the method `println`.

If you fix these mistakes, it will be possible to compile this program.

To avoid such errors, programmers use modern IDE (Integrated Development Environment) with a static code analyzer. This tool allows programmers to identify compile-time errors before the compilation. In addition, it is able to highlight warnings about more complex errors and weak places in your code, as well as tips on how to improve the code.

Over time, you will write code that contains less or even none of the compile-time errors.

## §2. Run-time errors

Run-time errors (also known as "bugs") are errors that occur when the program is running. Run-time errors will cause your program to behave unexpectedly or may even stop the execution.

There are two subtypes of run-time errors:

- **logic errors** – when a program produces a wrong result because the code is not correct (for example, instead of "Hello!", your program outputs "Hi!");
- **unhandled exceptional events** like division by zero, not found files, and other unexpected cases.

We will learn how to handle exceptional events (exceptions) in further lessons.

Avoiding such run-time errors is a more difficult task than avoiding compile-time errors. If your program compiles successfully, there are no guarantees that it does not have bugs. There are different strategies to find such errors:

- to **debug** your program;
- to write **automatic tests** for your program;

Current topic:

✓ Errors in programs Stage 7 ...

Topic depends on:

✓ Scanning the input Stage 1 ...

✗ Write, compile, and run Stage 3 ...

Topic is required for:

What is an exception Stage 7 ...

Table of contents:

[1 Errors in programs](#)

[§1. Compile-time errors](#)

[§2. Run-time errors](#)

[§3. Conclusion](#)

[Feedback & Comments](#)

- to use **code review** practice as part of the development process. In general, this practice stands for a case, when one or more developers visually inspect the source code of a program.

### §3. Conclusion

- Compile-time errors happen when a program fails to compile. They include typos and incorrect method invocation.
- Static code analyzers in IDEs help to spot compile-time errors before program compilation.
- Run-time errors, or bugs, occur after compilation when a program is running. They may result in program unexpected behavior and crashes.
- Debugging is a useful instrument to identify compilation errors in your program.

 Report a typo

**652** users liked this theory. **4** didn't like it. What about you?



Start practicing

[Comments \(3\)](#)   [Hints \(0\)](#)   [Useful links \(0\)](#)   [Show discussion](#)