CODEGYM

Java Developer in 12 months

Mentor-supported training program from CodeGym

MODULE 1. JAVA SYNTAX

Lesson 22 Elective

CODEGYM

Lesson plan

- Reserved words in Java
- Wrapping exceptions
- Error
- switch expression

Reserved words in Java

As in any programming language, Java has words that have special meaning. For example, return or if or while. These words are called keywords (keywords) and are considered reserved by the Java language.

You cannot use these words as a variable name, method name, or class name.

The compiler will always interpret them in a strictly defined way.

abstract	assert	boolean	break	byte
case	catch	char	class	const
continue	default	do	double	else
enum	extends	final	finally	float
for	goto	if	implements	import
instanceof	int	interface	long	native
new	package	private	protected	public
return	short	static	strictfp	super
switch	synchronize d	this	throw	throws
transient	try	void	volatile	while
var	true	null	false	

Primitive types

Java has 8 primitive types, and each of them has its own keyword:

- byte
- short
- int
- long
- char
- float
- double
- boolean
- void

Loops and branches

Loops and branches also give us a rather long list of keywords:

- if
- else
- switch
- case
- default
- while
- do
- for
- break
- continue

Exceptions

Exceptions give us 5 keywords:

- try
- catch
- finally
- throw
- throws

Visibility

There are three keywords:

- private
- protected
- public

Working with classes

There are 11 keywords in this category:

- class
- interface
- enum

- import
- package

- extends
- implements

- static
- final
- abstract
- default

CODEGYM

Working with objects and variables

Six more keywords are used when working with objects, methods, and variables:

- new
- instanceof
- this
- super
- return
- var (since Java 10)

Multithreading

At the level of Java syntax, multithreading is represented by just two words:

- synchronized
- volatile

Miscellaneous

There are another 4 special keywords:

- native
- transient
- assert
- strictfp

Reserved but not used

There are also two keywords that are reserved but not used.

- const
- goto

These are also a legacy of the C++ language, where they exist and are used.

Not keywords

Formally, the true, false and null constants are not keywords. That said, they each have their peculiarities. You cannot name a method true or a variable false. The compiler will not understand such code and will not compile it.

CODEGYM

Wrapping exceptions

However there is still a lot of code (including standard Java library code) that contains these checked exceptions. What is to be done with them? We can't ignore them, and we don't know how to handle them.

Java programmers proposed to wrap checked exceptions in RuntimeException. In other words, catch all the checked exceptions and then create unchecked exceptions (for example, RuntimeException) and throw them instead. Doing that looks something like this:

```
try {
  // Code where a checked exception might occur
}
catch(Exception exp) {
  throw new RuntimeException(exp);
}
```

CODEGYM

Error

Error is a critical runtime error related to the Java Virtual Machine.

In most cases, Error does not need to be handled, because it indicates some serious flaws in the code.

Here are the most famous Errors: StackOverflowError — occurs, for example, when a method endlessly calls itself, and OutOfMemoryError — occurs when there is not enough memory to create new objects.

As you can see, in these situations, there is generally nothing special you can do — the code is simply written incorrectly and must be redone.

CODEGYM

switch expression

The good old switch statement has been in Java since day one. But now things are starting to change: in Java 12 switch has become an expression instead of a statement:

```
public static int daysInMonth (Month month){
    return switch (month) {
        case JANUARY, MARCH, MAY, JULY, AUGUST, OCTOBER, DECEMBER -> 31;
        case APRIL, JUNE, SEPTEMBER, NOVEMBER -> 30;
        case FEBRUARY -> 28;
        default -> 0;
    };
}
```

The switch keyword gained the ability to return a result, which can then be assigned to a variable.

You can also use lambda-style syntax, which avoids passing through all case labels that do not have a break statement.

Arrow token (->)

The arrow token (->) indicates that now execution won't proceed to the next case label if the current case label does not have a break or return. Instead, the value to the right of the arrow will be returned.

You can also write code that is not an expression and does not return anything, but instead simply performs certain actions:

```
switch(condition) {
   case TRUE, FALSE -> System.out.println("True/false");

   default -> System.out.println("Another");
}
```

yield

This is a new keyword that will help you return a value from a switch, similar to the return keyword in methods.

The yield keyword was added in Java 13 for situations when case labels have more than one line of code and we want to return a result.

```
var result = switch (condition) {
  //...
  case "Hi" -> {
    // Your code
    yield "greeting";
  }
};
```

CODEGYM

Homework

Complete Level 23

MODULE 1. JAVA SYNTAX

CODEGYM

Answers to questions

MODULE 1. JAVA SYNTAX