**Обработка исключений в Java (try-catch)**

В Java исключения обрабатываются с помощью конструкции try-catch-finally, где finally – опциональная ветка, при наличии выполняется как после try, так и любой из веток catch. Иногда исключения перебрасываются выше по стеку вызовов. В этом случае метод в своем заголовке должен содержать throws класс\_исключения. Если может быть выброшено несколько исключений, они перечисляются через запятую.

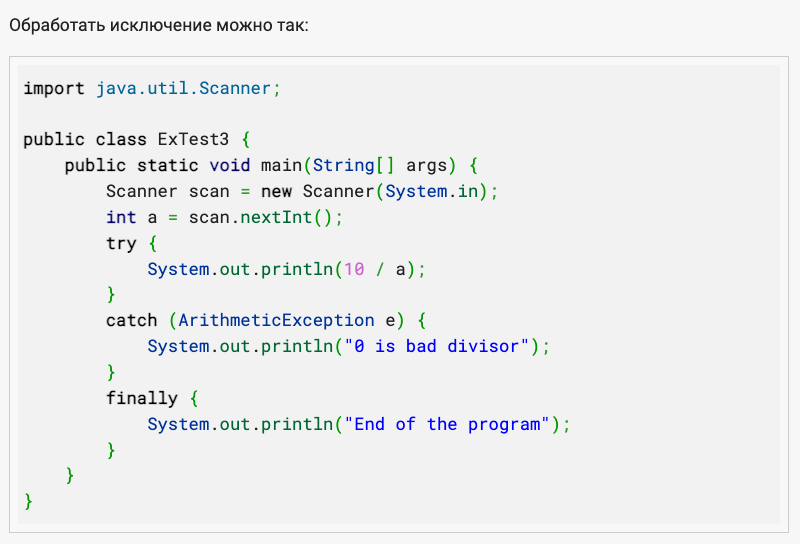


Во втором примере исключение можно было бы обработать в try-catch в методе main(). Можно указывать родительский класс исключения. Так вместо FileNotFoundException допустимо писать IOExctption или вообще класс Exception.

В Java исключения делятся на контролируемые (проверяемые) компилятором и неконтролируемые (непроверяемые). Приведенный выше исключения, которые могут возникать при вводе данных, – пример контролируемых. Если их не обработать или не передать вызывающему методу, то программу нельзя будет скомплировать.

Непроверяемые компилятором исключения должен контролировать сам программист. Так приведенная ниже программа удачно скомпилируется. Ошибка обнаружится лишь в процессе ее выполнения (runtime), если значение a окажется равным нулю.





Классов исключений в Java несколько сотен. По большей части они связаны между собой наследственными связями. Исходным является дочерний от Object класс Throwable. У него есть два наследника: Error и собственно Exception.

Непосредственно от класса Exсeption происходят классы RuntimeException и IOException. Первая группа относится к непроверяемым, вторая – к проверяемым.