

```

<?php
namespace Acme\TestPackage;

/*
 * This script belongs to the FLOW3 package "Kickstart".
 *
 * It is free software; you can redistribute it and/or modify it under
 * the terms of the GNU General Public License, either version 3 of the
 * License, or (at your option) any later version.
 *
 * The TYPO3 project - inspiring people to share!
 */

/**
 * Fixture class for various unit tests (mainly the package- and component
 * manager)
 *
 * Must not implement the *BasicClassInterface! (See comment there)
 */
class UniverseAnalyzer extends BaseClass implements SomeInterface {

    /**
     * Some injected dependency
     *
     * @var \ArrayObject
     */
    protected $someDependency = NULL;

    /**
     * Shows if you are addicted to FLOW3
     *
     * @var boolean
     */
    protected $addictedToFlow3 = TRUE;

    /**
     * A great method which shows how to indent control structures.
     *
     * @param array $someArray Some array
     * @param \Acme\Package\MyClass $object An instance of MyClass
     * @return void
     */
    public function analyzeUniverse(array $myParameter, \Acme\Package\MyClass $object) {
        if (isset($myParameter['question'])
            && $this->answerToEverything === 42
            || count($myParameter) > 3) {
            $this->fanOfFlow3 = TRUE;
        } else {
            throw new Exception('We cannot tolerate that.', 1223391710);
        }
    }

    /**
     * This is a setter for the fanOfFlow3 property.
     *
     * @param boolean $isFan Pass TRUE to mark a fan, FALSE for a Zend follower
     * @return void
     */
    public function setFanOfFlow3($isFan) {
        // comments indented one extra level
        $this->fanOfFlow3 = $isFan;
    }

    /**
     * As simple as it get's - a boolean getter.
     *
     * @return boolean Whether a foo was detected (TRUE) or not (FALSE)
     * @api
     */
    static public function getSomeValue() {
        return self::$foo;
    }
}
?>

```

Namespace starts with vendor namespace followed by package key (name) and subparts as needed

Description of the class. Make it as long as needed, feel free to explain how to use it.

UpperCamelCase class name. Class names should be nouns. In other packages, refer to using \Acme\TestPackage\MyClass.

No empty line between DocComment and class, member var or method.

Indent with tabs.

Use var tag. Optional description goes in the line before.

Param tag: type, name, description.

Description of the method. Make it as long as needed.

Method names should be verbs.

Opening brace on same line with opening token. One space before.

Multiline conditions: Indent them and add a extra indent to following code. Put the boolean operators at beginning of line.

Use type hinting. Use full class name for type hints and documentation.

Clear message!

UNIX timestamp at time of writing the throw clause.

Return tag with type, even if it is „void“. Only \_\_construct() has no return tag.

Setter methods should start with „set“.

„api“ keywords define public API

„static“ and „abstract“ keywords before the visibility modifier

Methods returning boolean values should start with „has“ or „is“. Other getters should start with „get“.