

Project:

StopCoding

Scope:

Project 'StopCoding'

Profile:

Default

Results:

Enabled coding rules: 30

- Checkstyle: 30

Problems found: 46

- Checkstyle: 46

Time statistics:

- Started: Tue Apr 02 14:45:14 EDT 2024
 - Initialization: 00:00:00.043
 - Checkstyle: 00:00:00.516
 - Gathering results: 00:00:00.331
- Finished: Tue Apr 02 14:45:15 EDT 2024

Detailed Results:

StopCoding

- Efficiency
 - Hide Utility Class Constructor
 - DataCenter
 - Utility classes should not have a public or default constructor. - line: 7
 - TimerService
 - Utility classes should not have a public or default constructor. - line: 12
- Maintainability
 - Unused Imports
 - TipsDialog.java
 - Unused import - java.time.LocalDateTime. - line: 10
 - TipsDialog.java
 - Unused import - icu.jokeen.stopcoding.task.WorkTask. - line: 5
 - TipsDialog.java
 - Unused import - icu.jokeen.stopcoding.service.TimerService. - line: 3
 - TipsDialog.java
 - Unused import - java.time.ZoneId. - line: 11
 - TipsDialog.java
 - Unused import - java.time.format.DateTimeFormatter. - line: 12
 - Visibility Modifier
 - DataCenter
 - Variable 'settingData' must be private and have accessor methods. - line: 12
 - Variable 'status' must be private and have accessor methods. - line: 13
 - Variable 'nextWorkTime' must be private and have accessor methods. - line: 14
 - Variable 'nextRestTime' must be private and have accessor methods. - line: 15
 - Variable 'workTimer' must be private and have accessor methods. - line: 17
 - Variable 'reskTimer' must be private and have accessor methods. - line: 18
 - Variable 'restCountDownSecond' must be private and have accessor methods. - line: 20

- RestTask
 - Variable 'tipsDialog' must be private and have accessor methods. - line: 15
- Reliability
 - Design For Extension
 - RestTask
 - Class 'RestTask' looks like designed for extension (can be subclassed), but the method 'run' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'RestTask' final or making the method 'run' static/final/abstract/empty, or adding allowed annotation for the method. - line: 22
 - SettingData
 - Class 'SettingData' looks like designed for extension (can be subclassed), but the method 'isOpen' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'SettingData' final or making the method 'isOpen' static/final/abstract/empty, or adding allowed annotation for the method. - line: 17
 - Class 'SettingData' looks like designed for extension (can be subclassed), but the method 'setOpen' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'SettingData' final or making the method 'setOpen' static/final/abstract/empty, or adding allowed annotation for the method. - line: 21
 - Class 'SettingData' looks like designed for extension (can be subclassed), but the method 'getWorkTime' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'SettingData' final or making the method 'getWorkTime' static/final/abstract/empty, or adding allowed annotation for the method. - line: 25
 - Class 'SettingData' looks like designed for extension (can be subclassed), but the method 'setWorkTime' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'SettingData' final or making the method 'setWorkTime' static/final/abstract/empty, or adding allowed annotation for the method. - line: 29
 - Class 'SettingData' looks like designed for extension (can be subclassed), but the method 'getRestTime' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'SettingData' final or making the method 'getRestTime' static/final/abstract/empty, or adding allowed annotation for the method. - line: 33
 - Class 'SettingData' looks like designed for extension (can be subclassed), but the method 'setRestTime' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'SettingData' final or making the method 'setRestTime' static/final/abstract/empty, or adding allowed annotation for the method. - line: 37
 - StopCodingInitializer
 - Class 'StopCodingInitializer' looks like designed for extension (can be subclassed), but the method 'preload' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'StopCodingInitializer' final or making the method 'preload' static/final/abstract/empty, or adding allowed annotation for the method. - line: 14
 - StopCodingSettingAction
 - Class 'StopCodingSettingAction' looks like designed for extension (can be subclassed), but the method 'actionPerformed' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'StopCodingSettingAction' final or making the method 'actionPerformed' static/final/abstract/empty, or adding allowed annotation for the method. - line: 9
 - TipsDialog

- Class 'TipsDialog' looks like designed for extension (can be subclassed), but the method 'setDesc' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'TipsDialog' final or making the method 'setDesc' static/final/abstract/empty, or adding allowed annotation for the method. - line: 63
- WorkTask
 - Class 'WorkTask' looks like designed for extension (can be subclassed), but the method 'run' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'WorkTask' final or making the method 'run' static/final/abstract/empty, or adding allowed annotation for the method. - line: 9
- Magic Number
 - SettingDialog
 - '400' is a magic number. - line: 42
 - '200' is a magic number. - line: 42
 - '300' is a magic number. - line: 43
 - '500' is a magic number. - line: 43
 - TimerService
 - '60' is a magic number. - line: 34
 - '60' is a magic number. - line: 78
 - '60' is a magic number. - line: 78
 - '60' is a magic number. - line: 79
 - '60' is a magic number. - line: 79
 - '60' is a magic number. - line: 79
 - '60' is a magic number. - line: 80
 - '10' is a magic number. - line: 88
 - TipsDialog
 - '400' is a magic number. - line: 26
 - '200' is a magic number. - line: 26
 - '200' is a magic number. - line: 27
 - '500' is a magic number. - line: 27
 - '1000' is a magic number. - line: 47
- Usability
 - Hidden Field
 - RestTask
 - 'tipsDialog' hides a field. - line: 18
 - SettingData
 - 'workTime' hides a field. - line: 29
 - 'restTime' hides a field. - line: 37