



**SAPLAND**

# Настройка шаблонов проверки АВАР-кода

Вебинар SAPLAND.

Весенняя сессия мастер-классов

[sapland.ru](http://sapland.ru)

# О чём поговорим на вебинаре?

На вебинаре разберем следующие вопросы:

1. Что такое АВАР Test Cockpit (АТС) и для чего он нужен.
2. Способы настройки АТС.
3. Настройка собственных вариантов проверки АВАР-кода
4. Объекты проверки и инспекции.
5. Обсудим темы для мастер-класса.

## ABAP Test Cockpit (ATC)

based on  
Code  
Inspector  
(SCI)

Syntax Check & Standard SCI Checks incl. SLIN

SAP HANA Checks (FUNCTIONAL\_DB)

SAP S/4HANA Checks

Security Checks (CVA)

ABAP Unit Tests

Customer specific checks...

АТС проверяет качество АВАР-объектов в нескольких областях, в том числе по производительности, безопасности, синтаксису, соглашениям по программированию, а также выполняет общие проверки. В зависимости от важности этих категорий проверок им присваивается приоритет. Эти проверки можно гибко настраивать

АТС объединяет несколько инструментов: Syntax Check (транзакции SE38 и SE80), Extended Syntax Check (SLIN), Code Inspector (SCI). АТС не просто интегрирует указанные инструменты, но функционирует как целостная платформа управления для проведения проверок качества кода с использованием функций отчетности, блокировки переносов и потока операций для обработки особых ситуаций возникающих в ходе проверки кода.

При автоматической блокировке переносов перед деблокированием выполняется автоматическая проверка переноса на наличие ошибок АТС в соответствующих АВАР-программах. При обнаружении ошибок АТС в таких АВАР-программах деблокирование переноса не выполняется, он остается в той же системе.

Локальная настройка АТС выполняется в системе разработки. После завершения настройки, отчеты АТС можно будет обрабатывать локально, вручную и периодически. Фоновые задания можно запланировать в рамках настройки для автоматического выполнения периодических отчетов.

Централизованная настройка осуществляется в системе обеспечения качества или в тестовой системе. Централизованная настройка позволяет проверять качество кода АВАР-программ из разных систем разработки, перенесенных в систему обеспечения качества. Результаты проверки при централизованной настройке переносятся обратно в соответствующие системы разработки. В этом случае по установленным критериям фильтрации для выполнения проверок при такой настройке учитываются только готовые программы, перенесенные в систему обеспечения качества.

Вопрос: Применяется ли АТС на вашем проекте?

Central Check System (SAP\_BASIS  $\geq$  7.51)

ABAP Test Cockpit (ATC)

RFC



Checked System A  
( $\geq$  7.00)

Customer Code



Checked System B  
( $\geq$  7.00)

Customer Code



Checked System C  
( $\geq$  7.00)

Customer Code



Remote Stubs

Вариант проверки АТС представляет собой набор правил с категоризацией по различным областям: общие проверки, проверки производительности, проверки безопасности, проверки синтаксиса, соглашения по программированию, функции поиска, проверки приложений, пользовательские интерфейсы и проверки удобства использования.

Для разных категорий вариантов проверки определяются ошибки с приоритетом 1, 2 и 3. Ошибки с приоритетом 1 и 2 являются критическими. Для обеспечения бесбойной работы продуктивной среды их необходимо устранить. Ошибки с приоритетом 3 несут в себе низкий риск. Их можно проигнорировать.

В настройке АТС используется вариант проверки Code Inspector. Стандартный вариант проверки SAP называется DEFAULT. Для пользовательской настройки варианта проверки существует определенный объем. Такая настройка выполняется на основе стандартов программирования и в соответствии с процессом проверки кода на данном проекте.

# Варианты проверки АТС

**Анализатор кодов: первый экран**

**И**

Ответственный RUEGOK

Инспекция

Имя  Верс.

Набор объектов

Имя  Верс.

Вариант проверки

Имя

Щелкните по пиктограмме копирования для создания пользовательского варианта проверки на основе стандартного варианта SAP. В результате этого действия все значения для варианта проверки DEFAULT переносятся в пользовательский вариант проверки.

Выберите этот значок, чтобы сделать вариант проверки глобальным

DOG(2)/100 Скопировать

Ответственный RUEGOK

Вариант проверки

Скопировать в

Новый ВарПроверки

Описание

# Варианты проверки АТС

Вариант проверки

Дата изменения  Автор изменения

Описание

Переносимо  Публичное  На базе экстракта  На базе RFC

Проверки для выполнения

В ссылочной системе проверки  
 Доступные локально проверки

Выделение	Д...	А...	Н...	Н...	Проверки
▼ <input checked="" type="checkbox"/>					Список проверок
▶ <input type="checkbox"/>					Общие проверки
▶ <input type="checkbox"/>					Cloud Readiness
▶ <input checked="" type="checkbox"/>					Проверки производительности
▶ <input checked="" type="checkbox"/>					Проверки защиты
▶ <input checked="" type="checkbox"/>					Проверка синт./Генерация
▶ <input checked="" type="checkbox"/>					Устойчивое программирование
▼ <input checked="" type="checkbox"/>					СоглашПоПрограммир
• <input checked="" type="checkbox"/>					Соглаш. по именам
• <input checked="" type="checkbox"/>					Расширенные соглашения по присвоению имен для программ
• <input type="checkbox"/>					Check Pretty Print State
• <input type="checkbox"/>					ABAP Unit Test Conventions
• <input type="checkbox"/>					Extended Check for ABAP Doc Comments
▶ <input type="checkbox"/>					S/4HANA Readiness

На следующем шаге определяются правила проверки, добавляем правила в стандартный вариант, например, в именованя классов добавим ZCL\_\* и ZCX\_.

Настраиваем соглашения под свои нужды и сохраняем вариант проверки.

DOG(1)/100 Соглаш. по прис. имен и префиксы

Префиксы | **Общее/локальн. процедура** | Только функц. программирование | Только об.-ориент. программиро

Им. объекта	
CLASS (глобальный)	<input checked="" type="checkbox"/> (./.)?CL_
CLASS (локальный)	LCL_
INTERFACE (глобальный)	<input checked="" type="checkbox"/> (./.)?IF_
INTERFACE (локальный)	LIF_

Классы	
DATA	M[:type:]_
CLASS-DATA	G[:type:]_
CONSTANTS	C[:type:]?_
TYPES	

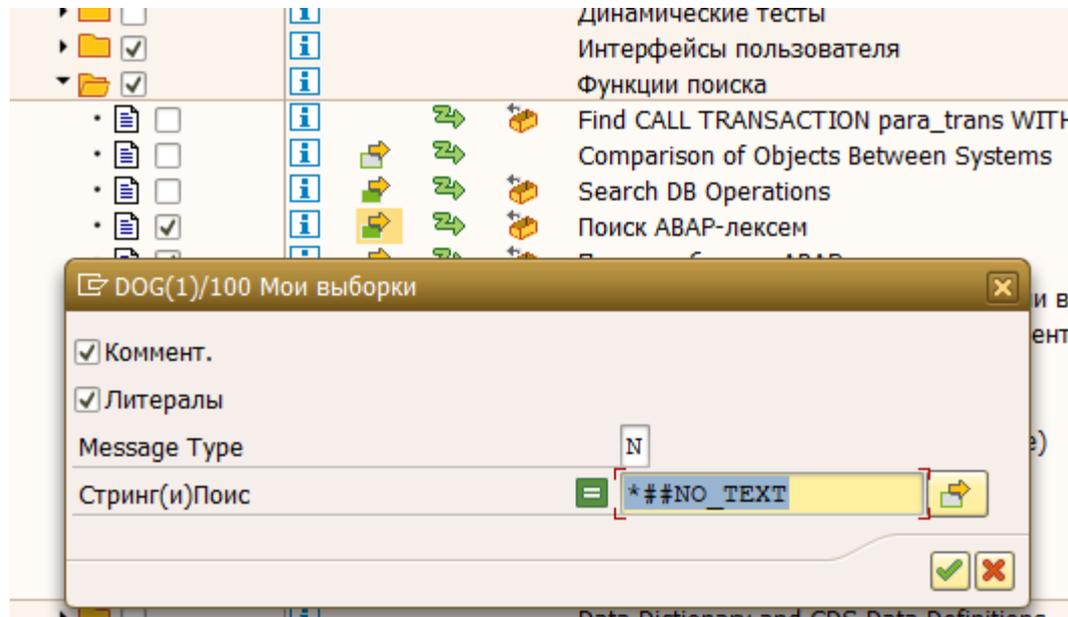
DOG(1)/100 Многократный выбор для CLASS (глобальный)

Выбрать отд. значения (2)

О.. ОтдельнЗначение	
<input checked="" type="checkbox"/>	(./.)?CL_
<input checked="" type="checkbox"/>	(./.)?CX_
<input type="checkbox"/>	(./.)?ZCL_
<input checked="" type="checkbox"/>	(./.)?ZCX_
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

Например, с помощью функции поиска в варианте проверки АТС можно выполнить явный поиск псевдокомментария в АВАР-программе. При использовании псевдокомментария ошибки АТС будут скрыты программными средствами без выполнения анализа. Такие ошибки могут являться критическими, в таком случае их необходимо устранить.

Активируем функции поиска / Поиск АВАР-лексем и вводим псевдокомментарий



# Варианты проверки АТС

интерфейсы пользователя			2	0	0
Функции поиска			0	0	41
Поиск АВАР-лексем			0	0	3
Информация			0	0	3
Код сообщ. 0001			0	0	3
Прогр. ZPM_R_077 Include ZPM_R_077_CI Строк 1591 Столб. 55			0	0	1
Стринг поиска № 0001 найден в: ##NO_TEXT					
Прогр. ZPM R 077 Include ZPM R 077 CI Строк 1592 Столб. 55			0	0	1

Введем также поиск шаблонов АВАР-операторов

The screenshot shows the SAP search functions menu with the following items:

- Find CALL TRANSACTION para\_trans WITH AUT
- Comparison of Objects Between Systems
- Search DB Operations
- Поиск АВАР-лексем
- Поиск шаблонов АВАР-операторов

A dialog box titled "DOG(1)/100 Мои выборки" is open, showing the definition of a search template:

ОбразецОператора	=	SELECT * INTO TABLE
Вид сообщения		N



Определяем набор объектов, который необходимо проверить, в нашем случае это тестовая программа ZAR\_TEST со следующим содержимым:

**Анализатор кода: изменить набор объектов**

В инспекциях В наборах объектов

Набор объектов	SO_TEST	Верс.	001	Ответственный	RUEGOK
Дата удаления		Дт/изменения	31.03.2022	Автор изменения	RUEGOK
Описание	SO_TEST__001		Число элементов	0	

Выбор НабораОбъектов    ОбработкаНабораОбъектов    НаОбъект из результ.    НаОбъект из запроса    К..

Сохранить только КритВыбора

Присвоение объектов

Код компонента		По		
Компонент ПО		По		
Уровень переноса		По		
Пакет		По		
Система оригинала		По		
Ответственный		По		

Выбор объектов

Классы, ГрФункций, ...    СвобВыборОбъект

<input type="checkbox"/> Класс/интерфейс		По		
<input type="checkbox"/> Группа функций		По		
<input checked="" type="checkbox"/> Прогр.	ZAR_TEST	По		
<input type="checkbox"/> Компонент веб-экрана		По		
<input type="checkbox"/> Тип АВАР-словаря		По		
<input type="checkbox"/> Группа типов		По		

Создаем инспекцию

### Анализатор кодов: первый экран

Ответственный RUEGOK

Инспекция

Имя CI\_TEST рс.

Прописываем набор объектов, вариант проверки и сохраняем

### Анализатор кода: изменить инспекцию

Инспекция CI\_TEST1 Версия 001 Ответственный RUEGOK

Дата выполнения  Дата изменения 12.04.2022 Автор изменения RUEGOK

Дата удаления 01.06.2022

Описание CI\_TEST1\_\_001

Выбор объекта

Набор объектов SO\_TEST Версия 001

Запрос/Задача

Отдельн

Вариант проверки

ZSPDS\_VAR\_TEST

## После выполнения смотрим результаты инспекции

The screenshot shows the SAP inspection results screen. A context menu is open over the 'Результаты' (Results) option, which is highlighted in yellow. The menu items are:

- Просмотреть <->Изменить Ctrl+F1
- Выполнить
- Выполнить...
- Результаты Shift+F6**
- Статистика Shift+F8
- Выход Shift+F3

The main screen displays the following information:

**Инспекция** | Обработать | Перейти к | Утилиты | Система | Справка

**треть инспекцию**

Инспекция: 1 | Дата: 12.04.2022 | Версия: 001 | Ответственный: RUEGOK

Дата выполнения: 12.04.2022 | Дата изменения: 12.04.2022 | Автор изменения: RUEGOK

Дата удаления: 01.06.2022

Описание: CI\_TEST1\_\_001

**Выбор объекта**

Набор объектов |  | Версия:

Запрос/Задача |

Отдельн |

**Вариант проверки**

**Анализатор кода: результаты CI\_TEST1 001 RUEGOK**

Инспекция: CI\_TEST1      Версия: 1      Ответственный: RUEGOK

Сообщения

	Д...	...	О...	Проверки	Ошибка	Предупр.	Инфор...
▼				Список проверок	1	35	3
▶				Проверки производительности	1	1	0
▶				Проверки защиты	0	0	0
▶				Проверка синт./Генерация	0	0	1
▶				Устойчивое программирование	0	0	2
▶				СоглашПоПрограммир	0	34	0
▶				Интерфейсы пользователя	0	0	0

Если ничего критичного нет, то в своих правилах можно удалить ненужные проверки или ограничить область проверок.

Рассмотрим пример ошибки:

**Анализатор кода: результаты CI\_TEST1 005 RUEGOK**

Инспекция CI\_TEST1      Версия 5      Ответственный RUEGOK

Сообщения

	Д...	...	О...	Проверки	Ошибка	Предупр.	Инфор...
				Список проверок	1	35	3
				Проверки производительности	1	1	0
				Анализ WHERE-условия для SELECT	1	0	0
				Ошибка	1	0	0
				Код сообщ. 0501	1	0	0
				Прогр. ZAR_TEST Include ZAR_TEST Строк 78 Столб. 2	1	0	0
				Большая таблица AUFK: нет WHERE-условия			
				==> Большая таблица AUFK: нет WHERE-условия			
				Анализ WHERE-условия для UPDATE и DELETE	0	0	0

Видим, что в коде нет ничего критического, поэтому можем снять эту проверку.

```
SELECT * INTO TABLE t_aufk FROM aufk UP TO 10 ROWS.
```

## Анализатор кода: изменить вариант проверки



Вариант проверки:

Дата изменения: 
 Автор изменения:

Описание:

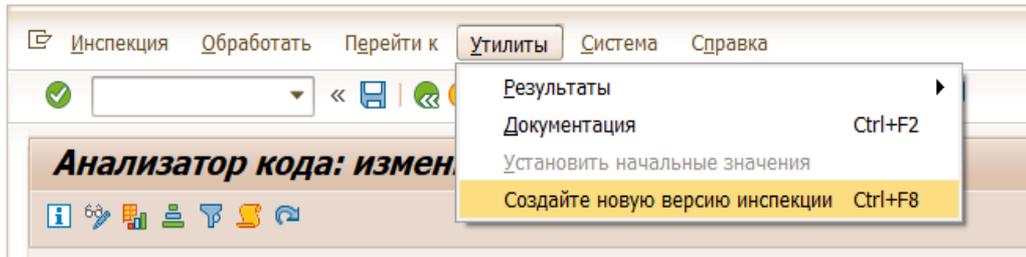
Переносимо
  Публичное
  На базе экстракта
  На базе RFC

### Проверки для выполнения

- В ссылочной системе проверки  
 Доступные локально проверки

Выделение	Д...	А...	Н...	Н...	Проверки
<ul style="list-style-type: none"> <li>▼ <input checked="" type="checkbox"/> </li> <li>▶ <input type="checkbox"/> </li> <li>▶ <input type="checkbox"/> </li> <li>▼ <input checked="" type="checkbox"/> </li> <li>• <input type="checkbox"/> </li> <li>• <input checked="" type="checkbox"/> </li> </ul>					Список проверок Общие проверки Cloud Readiness Проверки производительности Анализ WHERE-условия для SELECT Анализ WHERE-условия для UPDATE и DELETE SELECT-операторы, считывающие мимо табличного буфера Search problematic SELECT * statements Search SFLCT .. FOR ALL FNTRIES-clauses to be transformed

И снова запускаем инспекцию



После повторного прогона ошибок с приоритетом 1 больше нет

**Анализатор кода: результаты CI\_TEST1 006 RUEGOK**

Инспекция: CI\_TEST1    Версия: 6    Ответственный: RUEGOK

Сообщения

Д...	...	О...	Проверки	Ошибка	Предупр.	Инфор...
			Список проверок	0	35	3
▶			Проверки производительности	0	1	0
▶			Проверки защиты	0	0	0
▶			Проверка синт./Генерация	0	0	1
▶			Устойчивое программирование	0	0	2
▶			СоглашПоПрограммир	0	34	0
▶			Интерфейсы пользователя	0	0	0

# На кого ориентирован мастер-класс

Мастер-класс нужен вам, если вы:

1. Занимаетесь АВАР-разработкой и хотите писать качественный код.
2. Вы функциональный консультант, принимаете готовую разработку и хотите быть уверенным в ее качестве.
3. Вам нужно принимаете разработки у подрядчиков и вы хотите снизить риски для продуктивной системы, которые могут быть вызваны множественными ошибками разработчика и хотите автоматизировать процесс приемки кода.
4. Хотите научиться задавать правила приемки кода.  
Автоматизированно проверять код по расписанию, согласно заранее определенным правилам, и не допускать его переноса по ландшафту.

1. Обзор АТС.
2. Настройка АТС.
3. Варианты и объекты проверки, инспекции.
4. Автоматическая блокировка переносов
5. Особые ситуации АТС

Вопрос: какие темы вы хотели бы рассмотреть на мастер-классе?

# Промокод на мастер-класс

SAPLAND

На покупку любого мастер-класса весенней сессии SAPLAND действует промокод, дающий **скидку 35%**:

**ATC2022**

Промокод действует до 3 мая включительно при покупке **от физического лица**.

Вопрос: какие темы вы хотели бы рассмотреть на мастер-классе?



**Гофман Елена**

Моб.: +7 (909) 928-74-62

E-mail: [birdie172@mail.ru](mailto:birdie172@mail.ru)

**Настройка шаблонов  
проверки АВАР-кода**

**[sapland.ru](http://sapland.ru)**