

# Объектная модель AgavaPLC/AgavaSCADA

---



## Содержание

---

Введение

Узлы

Приложение

Типы вложенных узлов

Класс

Вложенные узлы

Структура

Функция

Программа

Секция

Наследование

Иерархия классов/объектов/узлов

Взаимодействие с узлами проекта.

Описание свойств, полей и методов базовых классов

Использование в дереве проекта узлов, унаследованных от базовых классов

## 1 Введение

---

Данный документ описывает объектную модель AgavaSCADA/AgavaPLC версии 1.5 и старше (новее).

## 2 Узлы

---

### 2.1 Приложение

Узел для организации алгоритмов.

Позволяет хранить внутри себя алгоритмы и структуры, а также узлы (окна, формы, регистры и т. д.).

#### 2.1.1 Типы вложенных узлов

- Класс (C++).
- Структура (C++).

- Функция (C++).
- Программа (C++).
- Секция (C++).

### **2.1.2 Класс**

Класс - это основополагающая сущность объектно-ориентированного программирования.

Узлы типа «Класс» позволяют реализовывать создание объектов определённого типа, описывая их структуру (набор полей и их начальное состояние) и определять алгоритмы (функции или методы) для работы с этими объектами.

Класс можно использовать для объявления экземпляров в функции, программе или методе другого класса. Возможно объявление экземпляров класса в дереве проекта и доступ к его полям, методам и свойствам из программ и функций.

Аналог класса в Codesys - функциональный блок.

#### **2.1.2.1 Вложенные узлы**

- Метод.
- Свойство. Специальный метод, предоставляющий доступ к полям через функции-сеттеры/геттеры.
- Действие. Специальный метод, доступный для использования в проекте как другие узлы типа "Действие".

### **2.1.3 Структура**

Структура - композитный тип данных, инкапсулирующий без сокрытия набор значений различных типов.

Структуру можно использовать для объявления в функции, процедуре и классе. Возможно объявление экземпляров структуры в дереве проекта и доступ к ее полям из программ и функций..

### **2.1.4 Функция**

Функция - фрагмент программного алгоритма, к которому можно обратиться из другого алгоритма - программы, метода или функции.

Функцию можно использовать для вызова в другой функции, процедуре или методах класса.

### **2.1.5 Программа**

Программа - новый элемент объектной модели, отсутствующий в языке C++. Наиболее близок к функции, имеет возможность объявления внутренних объектов (аналог блока

VAR), которые сохраняют свое значение при циклическом выполнении алгоритма. Позволяет задавать входные переменные (аналог блока VAR\_IN). В программе возможно использование всех декларативных типов узлов (функций, структур и классов). Аналог в Codesys – программа. Возможно добавление дополнительных вложенных функций, не доступных вне процедуры.

### **2.1.6 Секция**

Секция позволяет разместить в приложении часть программного кода без создания классов и функций в приложении. Секция компилируется вместе с приложением и все определенные в ней элементы будут доступны в других узлах приложения.

## **3 Наследование**

---

### **3.1 Иерархия классов/объектов/узлов**

- BasicNode – базовый класс узла.
- BasicSource – базовый класс узла-источника.
- BasicSignal – базовый класс узла-сигнала.
- BasicWindow – базовый класс узла-окна.
- BasicWidget – базовый класс узла-виджета.

и т.д.

### **3.2 Взаимодействие с узлами проекта.**

В функции, программы и методы можно передавать ссылки и указатели на узлы проекта и вызывать методы, определенные в узлах.

### **3.3 Описание свойств, полей и методов базовых классов**

Описание свойств, полей и методов базовых классов AgavaPLC/AgavaSCADA приведено в [описании базовых классов AgavaPLC/AgavaSCADA](#).

### **3.4 Использование в дереве проекта узлов, унаследованных от базовых классов**

При создании в программе собственного класса основанного на переопределении имеющихся базовых появляется возможность использования экземпляров этого класса в дереве проекта. При этом логика работы экземпляра такого класса определяется реализацией его собственных методов, а не методов базового класса.

---

Эта страница в последний раз была отредактирована 16 апреля 2024 в 16:02.