ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ГОСТ Р 60.0.7.6— 2025**

# Роботы и робототехнические устройства

# ОБЩАЯ ИНФОРМАЦИОННАЯ МОДЕЛЬ

## Описание общей информационной модели

Издание официальное

Москва
Российский институт стандартизации
2025

gost.e-ecolog.ru

## Предисловие

1 РАЗРАБОТАН Обществом с ограниченной ответственностью «Яндекс» (ООО «Яндекс»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 141 «Робототехника»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 27 ноября 2025 г. № 1547-ст

4 ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Оформление. ФГБУ «Институт стандартизации», 2025

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

## Содержание

1 Область применения . . . . . 1
2 Нормативные ссылки . . . . . 1
3 Термины, определения и сокращения . . . . . 2
4 Общая информационная модель роботов . . . . . 3
Приложение А (обязательное) Правила присвоения имен . . . . . 27
Приложение Б (справочное) Формат представления класса . . . . . 28
Приложение В (справочное) Рекомендации по применению общей информационной модели. . . . . 30

## Введение

Целью стандартов комплекса «Роботы и робототехнические устройства» является повышение интероперабельности роботов и их компонентов, а также снижение затрат на их разработку, производство и обслуживание за счет стандартизации и унификации процессов, интерфейсов и параметров.

Стандарты комплекса «Роботы и робототехнические устройства» представляют собой совокупность отдельно издаваемых стандартов. Стандарты данного комплекса относятся к одной из следующих тематических групп: «Общие положения, основные понятия, термины и определения», «Технические и эксплуатационные характеристики», «Безопасность», «Виды и методы испытаний», «Механические интерфейсы», «Электрические интерфейсы», «Коммуникационные интерфейсы», «Методы программирования», «Методы построения траектории движения (навигация)», «Конструктивные элементы». Стандарты любой тематической группы могут относиться как ко всем роботам и робототехническим устройствам, так и к отдельным группам объектов стандартизации — промышленным роботам в целом, промышленным манипуляционным роботам, промышленным транспортным роботам, сервисным роботам в целом, сервисным манипуляционным роботам и сервисным мобильным роботам.

Настоящий стандарт относится к тематической группе «Методы моделирования и программирования» и распространяется на все виды роботов.

Настоящий стандарт устанавливает требования к описанию общей информационной модели (ОИМ) роботов. Стандарт разработан с целью установления единой информационной модели данных для интеграции роботов в автоматизированные производственные, складские и иные виды систем.

Стандарт устанавливает структуру данных, объекты и их взаимосвязи, которые должны быть использованы для описания компонентов роботов.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Роботы и робототехнические устройства**

**ОБЩАЯ ИНФОРМАЦИОННАЯ МОДЕЛЬ**

**Описание общей информационной модели**

Robots and robotic devices. Integration of robotic systems.
Description of the general information model

**Дата введения — 2026—01—01**

## 1 Область применения

Настоящий стандарт устанавливает требования к ОИМ роботов различного назначения. ОИМ представляет собой формализованную структуру описания состава, компонентов, параметров и состояния робота, предназначенную для его интеграции в информационные системы, автоматизированные системы управления, цифровые двойники, а также для взаимодействия с другими роботами.

Стандарт определяет архитектуру ОИМ, включая:

- состав и структуру классов и объектов модели;
- системные и прикладные атрибуты компонентов робота;
- правила построения иерархии и связей между частями робота.

ОИМ, на которую распространяется настоящий стандарт, рекомендуется применять при разработке, внедрении и эксплуатации роботов, в том числе в рамках цифровизации, автоматизации производств, интеллектуального управления и киберфизических систем.

Настоящий стандарт предназначен для разработчиков, изготовителей, интеграторов и пользователей роботов, а также для специалистов, занимающихся проектированием и обслуживанием автоматизированных производственных комплексов, в которых используются роботы.

Требования настоящего стандарта основаны на требованиях стандартов комплекса «Роботы и робототехнические устройства».

## 2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 7.0.64 (ИСО 8601:2004) Система стандартов по информации, библиотечному и издательскому делу. Представление дат и времени. Общие требования

ГОСТ Р 60.0.0.4/ИСО 8373:2021 Роботы и робототехнические устройства. Термины и определения

ГОСТ Р 60.0.0.22 Роботы и робототехнические устройства. Общая информационная модель. Термины и определения

ГОСТ Р ИСО/МЭК 11179-5 Информационная технология. Регистры метаданных (РМД). Часть 5. Принципы наименования и идентификация

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого

**Издание официальное**

стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

## 3 Термины, определения и сокращения

### 3.1 Термины и определения

В настоящем стандарте применены термины по ГОСТ Р 60.0.0.4 и ГОСТ Р 60.0.0.22.

### 3.2 Сокращения

В настоящем стандарте применены следующие сокращения:

АСУ — автоматизированная система управления;

ИМ — информационная модель;

ИС — информационная система;

ОИМ — общая информационная модель;

API — интерфейс прикладного программирования (application programming interface);

DHCP — протокол динамической конфигурации узлов сети, предназначенный для автоматического присвоения IP-адресов и сопутствующих параметров сетевого соединения клиентским устройствам в компьютерной сети (Dynamic Host Configuration Protocol);

DNS — распределенная иерархическая система доменных имен, обеспечивающая преобразование человекочитаемых доменных имен в IP-адреса и обратное преобразование, используемая для упрощения доступа к ресурсам компьютерных сетей (Domain Name System);

EAP — сетевой протокол, обеспечивающий гибкий механизм аутентификации и поддержки различных методов проверки подлинности, применяемый в рамках протоколов доступа к сетям (Extensible Authentication Protocol);

IP — интернет-протокол (Internet protocol);

JSON — текстовый формат обмена данными, основанный на языке JavaScript (JavaScript object notation);

NTP — сетевой протокол, предназначенный для синхронизации системных часов компьютеров и других устройств в распределенных вычислительных системах с высокой точностью посредством обмена временными метками по сети (Network Time Protocol);

PEAP — защищенный расширяемый протокол аутентификации, обеспечивающий безопасный обмен учетными данными пользователя посредством создания зашифрованного TLS-туннеля между клиентом и сервером в сетях с поддержкой IEEE 802.1X (Protected Extensible Authentication Protocol);

PTP — сетевой протокол, предназначенный для синхронизации часов устройств в компьютерных сетях с высокой точностью, обеспечивающий передачу временных меток и коррекцию системного времени на основе иерархической структуры синхронизации (Precision Time Protocol);

RTC — специализированный таймер, встроенный в вычислительные системы, обеспечивающий постоянный и точный учет времени и даты, необходимый для корректного функционирования программного обеспечения и системного планирования (Real-Time Clock);

CSV — текстовый формат представления табличных данных, в котором значения отдельных полей разделяются запятыми, а каждая строка файла соответствует одной записи таблицы (Comma-Separated Values);

TLS — безопасность транспортного уровня (transport layer security);

TTLS — протокол аутентификации, обеспечивающий безопасный обмен учетными данными пользователя посредством создания зашифрованного TLS-туннеля между клиентом и сервером, используемый в сетях с поддержкой IEEE 802.1X для защиты передачи аутентификационной информации (Tunneled Transport Layer Security);

Wi-Fi — технология беспроводных локальных сетей, позволяющая электронным устройствам подключаться к сети, в основном используя диапазоны 2,4 ГГц и 5 ГГц (wireless fidelity).

# 4 Общая информационная модель роботов

## 4.1 Общие положения

Каждый объект ОИМ представляет собой элемент робота с набором атрибутов и методов, которые описывают его состояние и поведение. ОИМ должна поддерживать как статические данные (например, характеристики компонентов), так и динамические данные (например, актуальное состояние робота).

## 4.2 Компоненты информационной модели

ОИМ состоит из следующих ключевых компонентов:

- объекты — основные элементы ОИМ, которые описывают компоненты и/или совокупность компонентов робота;
- атрибуты — характеристики, которые описывают состояние объекта или его параметры (например, координаты робота, тип сенсора, статус системы);
- методы — функции, которые определяют действия или операции, которые может выполнить объект и/или совокупность объектов (например, движение робота, захват объекта, настройка сенсора);
- связи между объектами — описание взаимосвязей и взаимодействий между различными компонентами робота (например, связь между роботом и его манипулятором, между сенсором и системой управления).

## 4.3 Масштабируемость

ОИМ спроектирована с учетом возможности добавления новых типов объектов, самих объектов и их атрибутов для поддержки будущих технологий и новых типов роботов. С этой же целью каждый объект в ОИМ обладает возможностью расширения за счет добавления дополнительными атрибутами и методами. Основными принципами проектирования ОИМ являются:

- расширяемость — возможность интеграции с новыми типами роботизированных решений;
- модульность — возможность описать дополнительные компоненты и типы устройств, такие как новые типы сенсоров, приводы, а также новые системы управления;
- совместимость с современными программными средствами и платформами — возможность реализации информационной модели с помощью различных промышленных протоколов и интерфейсов обмена данными.

## 4.4 Взаимосвязь между общей информационной моделью и конкретными информационными моделями

ОИМ роботов необходимо использовать в информационных моделях всех типов роботов, к которым относятся промышленные и сервисные роботы. Взаимосвязи между информационными моделями типов роботов представлены на рисунке 1.




Рисунок 1 — Взаимосвязи между информационными моделями различных типов роботов

## 4.5 Класс для общей информационной модели

### 4.5.1 Общие положения

Корневым классом для ОИМ является класс Robot, который должен быть сформирован из четырех внутренних классов, представленных на рисунке 2. Имена характеристик и классов, используемых

в ОИМ, должны соответствовать правилам присвоения имен, установленным в приложении А. Таблицы классов и перечислений заполняются согласно формату, описанному в приложении Б. Рекомендации по применению ОИМ — в приложении В.

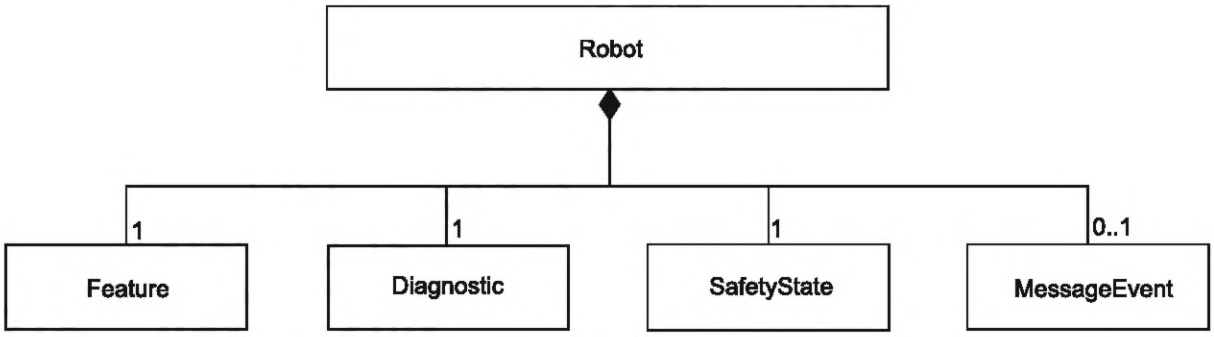


Рисунок 2 — Классы ОИМ

Диаграмма классов для класса Robot приведена на рисунке 3.

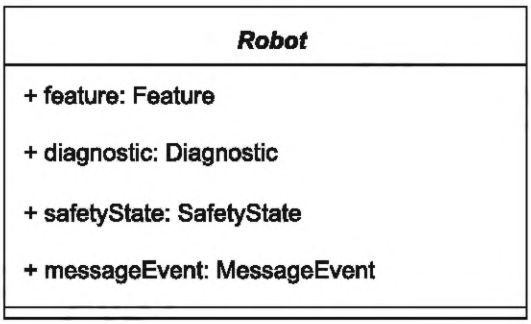


Рисунок 3 — Диаграмма классов для класса Robot

Класс Robot приведен в таблице 1.

Таблица 1 — Класс Robot

| Описание: Корневой класс Robot для ОИМ | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| feature | Feature | M | 1 | Общие параметры и характеристики робота (см. 4.5.2) |
| diagnostic | Diagnostic | M | 1 | Текущее состояние робота, включая диагностические параметры (см. 4.5.3) |
| safetyState | SafetyState | M | 1 | Текущее состояние систем безопасности робота (см. 4.5.4) |
| messageEvent | MessageEvent | O | 1 | Журнал текущих событий и сообщений робота (см. 4.5.5) |

### 4.5.2 Класс для характеристик робота

Класс Feature предназначен для хранения общих параметров и характеристик робота, а также информации, уникальной для каждого конкретного экземпляра робота. Этот объект служит основной точкой для описания основных характеристик, таких как модель, тип робота, а также специфической информации, необходимой для идентификации и настройки робота в системе. Класс Feature обладает атрибутами и методами, представленными на рисунке 4 и в таблице 2.

Таблица 2 — Класс Feature

| Описание: Информация об общих параметрах и характеристиках робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| name | String | O | 1 | Имя робота |
| serialNumber | String | M | 1 | Уникальный серийный номер робота |
| manufacturer | String | M | 1 | Информация о наименовании изготовителя робота |
| model | String | M | 1 | Информация с указанием модели робота |
| type | RobotType | M | 1 | Тип робота |
| description | String | O | 1 | Общее представление робота, что он собой представляет, что он делает и как его использовать |
| swVersion | String | M | 1 | Номер версии программного обеспечения в формате [Major].[Minor].[Patch] (например, 1.3.2) |
| firmwareVersion | String | M | 1 | Номер версии прошивки в формате [Major].[Minor].[Patch] (например, 1.3.2) |
| hwVersion | String | O | 1 | Номер версии аппаратного обеспечения робота в формате [Major] (например, 1) |
| cimVersion | String | M | 1 | Номер версии общей информационной модели |
| manufacturedDate | DateTime | O | 1 | Дата изготовления робота |
| time | DateTime | M | 1 | Текущее время, установленное на роботе |
| communication | Communication | M | 1 | Коммуникационные настройки робота |
| dateTimeSettings | DateTimeSettings | M | 1 | Настройки даты и времени на роботе |
| Методы: | | | | |
| setName (name: String) | MethodResult | O | Задание имени робота | |
| uploadFirmware (firmware: Byte[]) | MethodResult | O | Загрузка файла прошивки робота | |
| updateFirmware (fileName: String) | MethodResult | O | Обновление прошивки робота | |
| uploadSoftware (software: Byte[]) | MethodResult | O | Загрузка файла программного обеспечения робота | |
| updateSoftware (fileName: String) | MethodResult | O | Обновление программного обеспечения робота | |

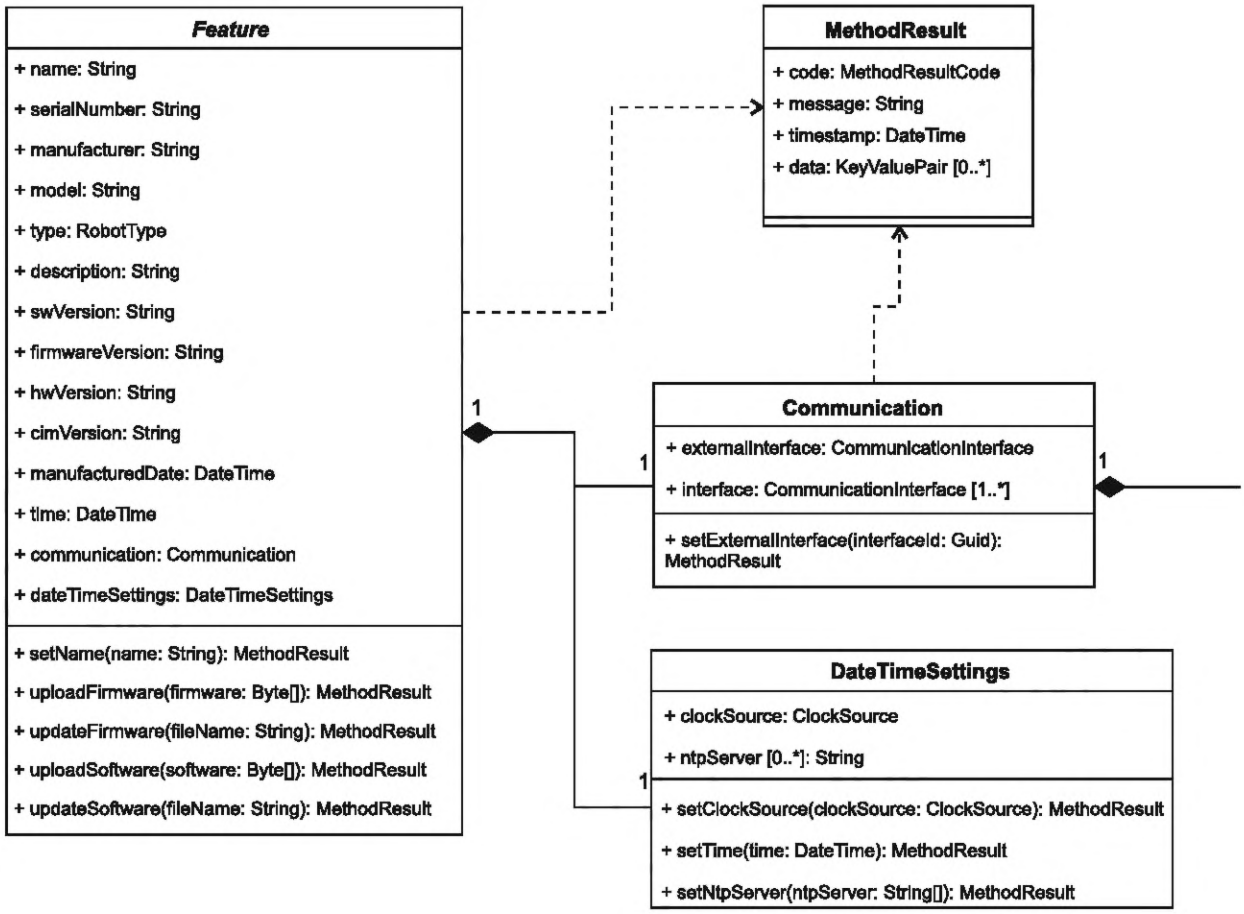


Рисунок 4 — Диаграмма классов для класса Feature, лист 1

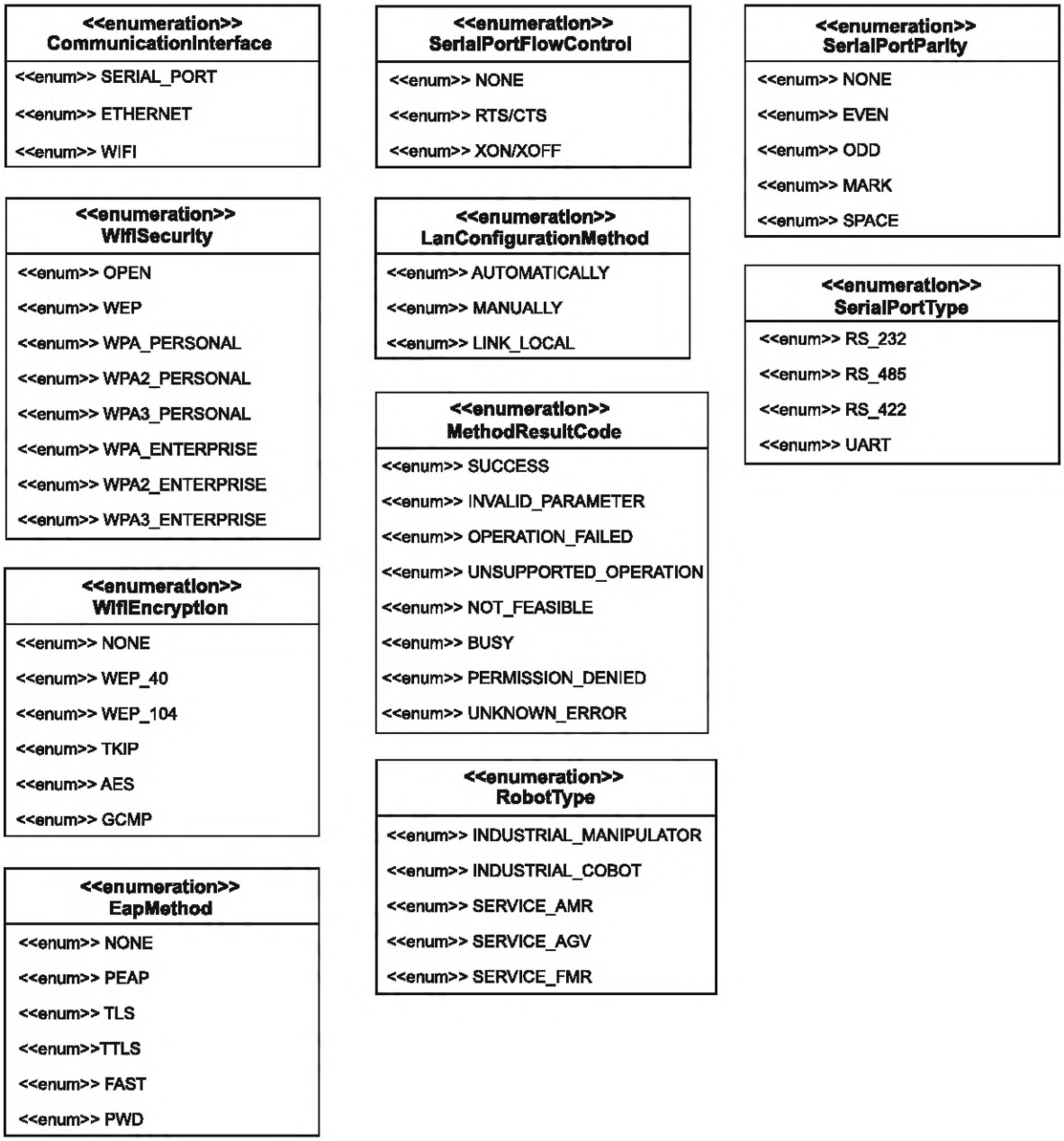


Рисунок 4, лист 2

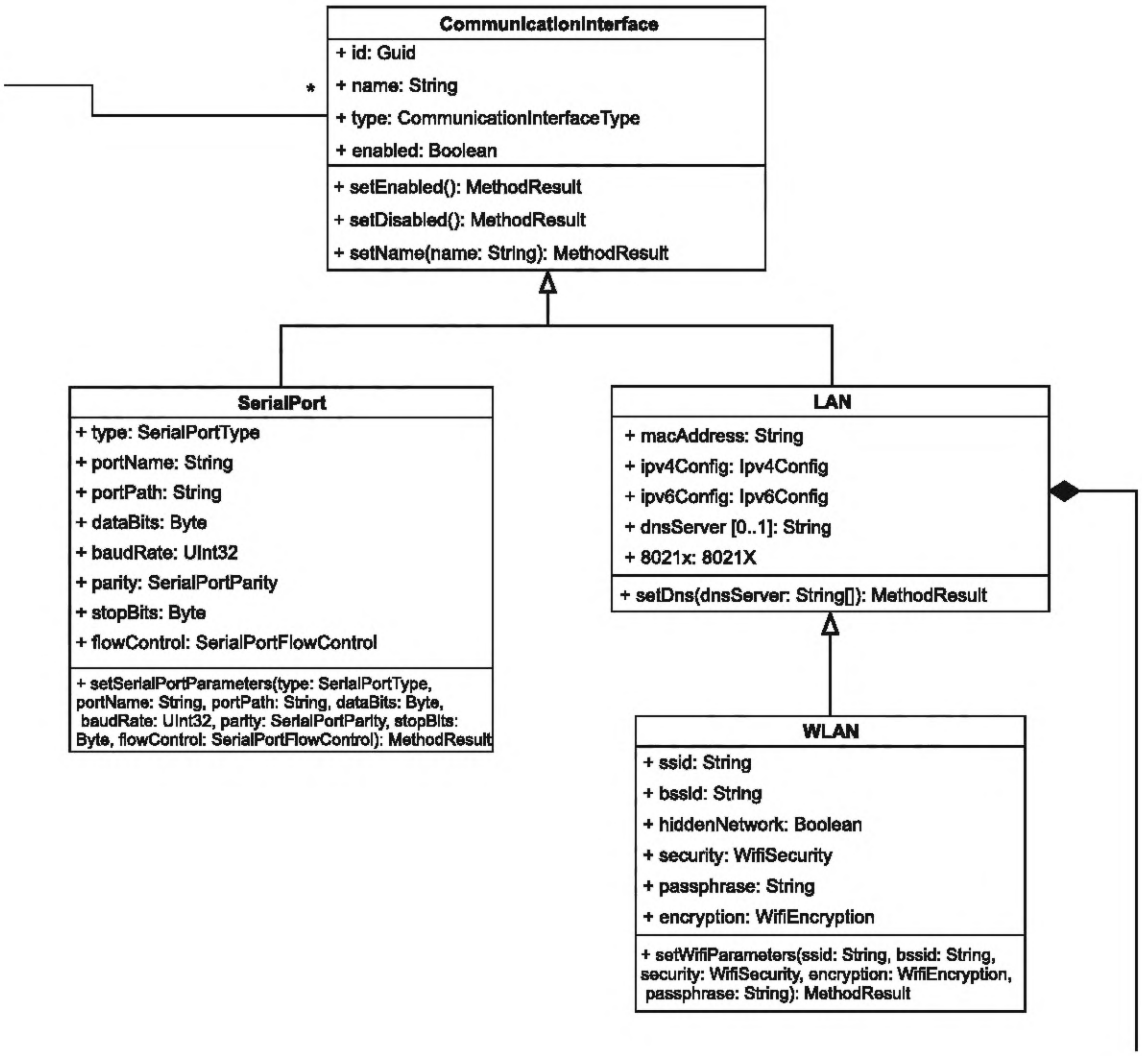


Рисунок 4, лист 3

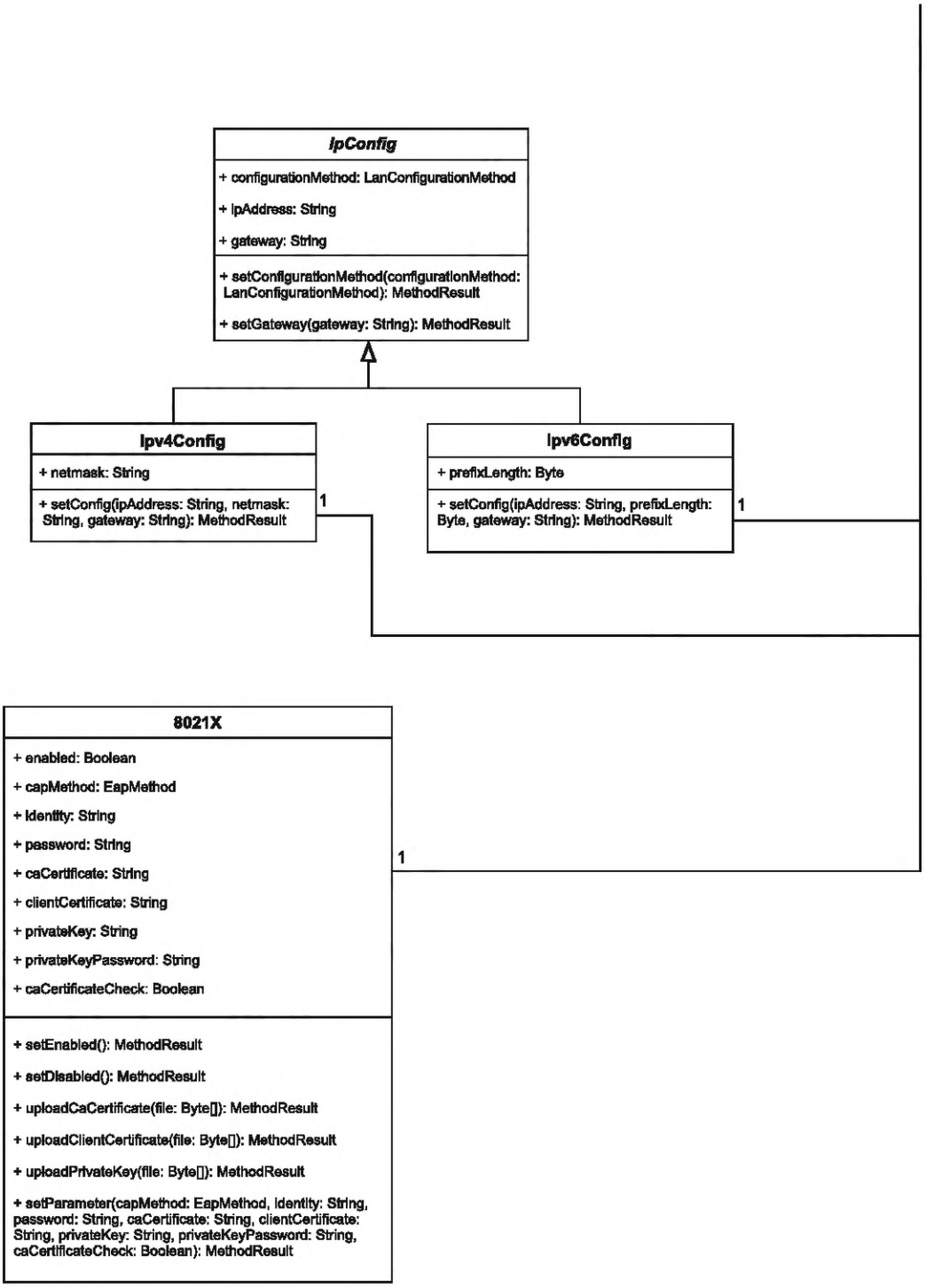


Рисунок 4, лист 4

Перечисление MethodResultCode приведено в таблице 3.

Таблица 3 — Перечисление MethodResultCode

| Описание: Коды статуса выполнения метода | |
|---|---|
| Атрибуты: | |
| SUCCESS | Операция выполнена успешно |
| INVALID_PARAMETER | Неверные входные данные |
| OPERATION_FAILED | Ошибка при выполнении операции |
| UNSUPPORTED_OPERATION | Запрошенная операция не реализована для данного типа робота |
| NOT_FEASIBLE | Операция не может быть выполнена в текущих условиях |
| BUSY | Робот занят другой операцией и не может начать новую |
| PERMISSION_DENIED | Недостаточно прав для выполнения операции |
| UNKNOWN_ERROR | Неклассифицированная ошибка |

Перечисление RobotType приведено в таблице 4.

Таблица 4 — Перечисление RobotType

| Описание: Типы роботов | |
|---|---|
| Атрибуты: | |
| INDUSTRIAL_ARM | Промышленный манипулятор |
| INDUSTRIAL_COBOT | Промышленный коллаборативный робот (робот, предназначенный для совместной работы с человеком) |
| SERVICE_AMR | Сервисный автономный мобильный робот |
| SERVICE_AGV | Сервисная автоматически управляемая тележка |
| Примечание — Перечисление может быть переопределено для информационных моделей каждого типа робота, учитывая актуальную классификацию роботов. | |

Перечисление CommunicationInterfaceType приведено в таблице 5.

Таблица 5 — Перечисление CommunicationInterfaceType

| Описание: Тип коммуникационного интерфейса | |
|---|---|
| Атрибуты: | |
| SERIAL_PORT | Последовательный порт |
| ETHERNET | Ethernet |
| WIFI | Wi-Fi |

Перечисление SerialPortType приведено в таблице 6.

Таблица 6 — Перечисление SerialPortType

| Описание: Тип последовательного порта | |
|---|---|
| Атрибуты: | |
| RS_232 | RS-232 |
| RS_485 | RS-485 |
| RS_422 | RS_422 |
| UART | UART (универсальный асинхронный приемопередатчик) |

Перечисление SerialPortFlowControl приведено в таблице 7.

Таблица 7 — Перечисление SerialPortFlowControl

| Описание: Способ управления потоком данных | |
|---|---|
| Атрибуты: | |
| NONE | Управление потоком данных отсутствует |
| RTS/CTS | Аппаратное управление потоком данных |
| XON/XOFF | Программное управление потоком данных |

Перечисление SerialPortParity приведено в таблице 8.

Таблица 8 — Перечисление SerialPortParity

| Описание: Способ контроля четности для обнаружения ошибок | |
|---|---|
| Атрибуты: | |
| NONE | Без контроля бита четности |
| EVEN | С контролем бита четности (если в байте число единиц четное, то бит четности устанавливается в «1», в противном случае — в «0») |
| ODD | С контролем бита четности (если в байте число единиц нечетное, то бит четности устанавливается в «1», в противном случае — в «0») |
| MARK | Постоянно «1» (бит четности всегда устанавливается) |
| SPACE | Постоянно «0» (бит четности всегда сброшен) |

Перечисление LanConfigurationMethod приведено в таблице 9.

Таблица 9 — Перечисление LanConfigurationMethod

| Описание: Способ задания сетевых настроек | |
|---|---|
| Атрибуты: | |
| AUTOMATICALLY | Получение настроек в автоматическом режиме |
| MANUALLY | Ручное задание настроек |
| LINK_LOCAL | Сетевые адреса, предназначенные для создания соединений в пределах одного сегмента локальной сети |

Перечисление WifiSecurity приведено в таблице 10.

Таблица 10 — Перечисление WifiSecurity

| Описание: Режим аутентификации подключения Wi-Fi | |
|---|---|
| Атрибуты: | |
| OPEN | Открытый доступ |
| WEP | Общий ключ |
| WPA_PERSONAL | Используют общий пароль для всей сети |
| WPA2_PERSONAL | Используют общий пароль для всей сети (алгоритм шифрования AES CCMP, длина ключа 128 бит) |
| WPA3_PERSONAL | Используют общий пароль для всей сети (при каждом соединении устанавливается новый шифрующий пароль, длина ключа 192 бита и т. д.) |
| WPA_ENTERPRISE | Используют полнофункциональный сервер аутентификации |
| WPA2_ENTERPRISE | Используют полнофункциональный сервер аутентификации (алгоритм шифрования AES CCMP) |
| WPA3_ENTERPRISE | Используют полнофункциональный сервер аутентификации (длина ключа 192 бита) |

Перечисление WifiEncryption приведено в таблице 11.

Таблица 11 — Перечисление WifiEncryption

| Описание: Способ шифрования пакетов подключения Wi-Fi | |
|---|---|
| Атрибуты: | |
| NONE | Без шифрования |
| WEP_40 | Использует 40-битный ключ, алгоритм шифрования RC4, длина ключа шифрования — 40 бит, ключ шифрования статический (не изменяется) |
| WEP_104 | Использует 104-битный ключ, алгоритм шифрования RC4, длина ключа шифрования — 104 бита. Ключ шифрования статический (не изменяется) |
| TKIP | Динамическое изменение ключа пакетов, длина ключа — до 128 бит, используется механизм проверки целостности сообщений (MIC) |
| AES | Симметричный блочный криптографический алгоритм шифрования |
| GCMP | Шифрование данных и их аутентификация |

Перечисление EapMethod приведено в таблице 12.

Таблица 12 — Перечисление EapMethod

| Описание: Режим EAP-аутентификации подключения Wi-Fi | |
|---|---|
| Атрибуты: | |
| NONE | Аутентификация отсутствует |
| PEAP | Сертификат у сервера, у клиента логин/пароль |
| TLS | Криптографические сертификаты |
| TTLS | Сертификат только у сервера, у клиента — логин/пароль |
| FAST | Быстрая аутентификация от Cisco |
| PWD | Парольная аутентификация |

Класс MethodResult приведен в таблице 13.

Таблица 13 — Класс MethodResult

| Описание: Данные, возвращаемые методами классов | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| code | MethodResultCode | M | 1 | Код статуса выполнения метода |
| message | String | O | 1 | Описание возвращаемого методом результата |
| timestamp | DateTime | O | 1 | Метка времени формирования результата метода |
| data | KeyValuePair | O | N | Данные выполнения метода |

Класс CommunicationInterface приведен в таблице 14.

Таблица 14 — Класс CommunicationInterface

| Описание: Класс для коммуникационных интерфейсов робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| id | Guid | M | 1 | Активный коммуникационный интерфейс |
| name | String | O | 1 | Пользовательское имя интерфейса |
| type | CommunicationInterfaceType | M | 1 | Тип коммуникационного интерфейса |
| enabled | Boolean | M | 1 | Флаг активации интерфейса |
| Методы: | | | | |
| setEnabled() | MethodResult | M | Включение интерфейса | |
| setDisabled() | MethodResult | M | Отключение интерфейса | |
| setName (name: String) | MethodResult | O | Задание пользовательского имени коммуникационного интерфейса | |

Класс SerialPort приведен в таблице 15.

Таблица 15 — Класс SerialPort

| Описание: Конфигурация подключения по последовательному порту | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| type | SerialPortType | M | 1 | Тип последовательного порта |
| portName | String | O | 1 | Имя порта в контроллере |
| portPath | String | O | 1 | Путь порта в контроллере |
| dataBits | Byte | M | 1 | Количество бит данных |
| baudRate | SerialPortBaudRate | M | 1 | Скорость передачи данных |
| parity | SerialPortParity | M | 1 | Способ контроля четности для обнаружения ошибок |
| stopBits | Byte | M | 1 | Количество стоповых битов |
| flowControl | SerialPortFlow Control | M | 1 | Способ управления потоком данных |

*Окончание таблицы 15*

| | | | |
|---|---|---|---|
| Методы: | | | |
| setSerialPortParameters (type: SerialPortType, portName: String, portPath: String, dataBits: Byte, baudRate: UInt32, parity: SerialPortParity, stopBits: Byte, flowControl: SerialPortFlowControl) | MethodResult | M | Задание конфигурации подключения по последовательному порту |

Класс IpConfig приведен в таблице 16.

Таблица 16 — Класс IpConfig

| | | | | |
|---|---|---|---|---|
| Описание: Конфигурация сетевых настроек | | | | |
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| configurationMethod | LanConfigurationMethod | M | 1 | Способ задания сетевых настроек |
| ipAddress | String | M | 1 | IP-адрес |
| gateway | String | M | 1 | Адрес шлюза по умолчанию |
| Методы: | | | | |
| setConfigurationMethod (configurationMethod: LanConfigurationMethod) | MethodResult | O | Задание IP-конфигурации | |
| setGateway(gateway: String) | MethodResult | O | Задание адреса шлюза | |

Класс Ipv4Config приведен в таблице 17.

Таблица 17 — Класс Ipv4Config

| | | | | |
|---|---|---|---|---|
| Описание: Конфигурация сетевых настроек IPv4 | | | | |
| Выведен из: IpConfig | | | | |
| Атрибуты: | | | | |
| netmask | String | M | 1 | Маска подсети |
| Методы: | | | | |
| setConfig(ipAddress: String, netmask: String, gateway: String) | MethodResult | M | Задание параметров IPv4 | |

Класс Ipv6Config приведен в таблице 18.

Таблица 18 — Класс Ipv6Config

| | | | | |
|---|---|---|---|---|
| Описание: Конфигурация сетевых настроек IPv6 | | | | |
| Выведен из: IpConfig | | | | |
| Атрибуты: | | | | |
| prefixLength | Byte | M | 1 | Префикс IPv6 |
| Методы: | | | | |
| setConfig(ipAddress: String, prefixLength: Byte, gateway: String): MethodResult | MethodResult | M | Задание параметров IPv6 | |

Класс 8021X приведен в таблице 19.

Таблица 19 — Класс 8021X

| Описание: Конфигурация параметров аутентификации по 802.1X | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| enabled | Boolean | M | 1 | Включена ли аутентификация по 802.1X |
| eapMethod | EapMethod | M | 1 | Режим EAP-аутентификации |
| identity | String | O | 1 | Имя пользователя или идентификатор |
| password | String | O | 1 | Пароль для EAP-аутентификации |
| caCertificate | String | O | 1 | Путь к CA-сертификату |
| clientCertificate | String | M | 1 | Путь к клиентскому сертификату |
| privateKey | String | M | 1 | Путь к закрытому ключу |
| privateKeyPassword | String | M | 1 | Пароль от закрытого ключа |
| caCertificateCheck | Boolean | O | 1 | Обязательна ли проверка сертификата сервера |
| Методы: | | | | |
| setEnabled() | MethodResult | M | Включение аутентификации по 802.1X | |
| setDisabled() | MethodResult | M | Отключение аутентификации по 802.1X | |
| uploadCaCertificate(file: Byte[]) | MethodResult | M | Загрузка файла серверного сертификата | |
| uploadClientCertificate(file: Byte[]) | MethodResult | M | Загрузка файла клиентского сертификата | |
| uploadPrivateKey(file: Byte[]) | MethodResult | M | Загрузка файла закрытого ключа | |

Класс LAN приведен в таблице 20.

Таблица 20 — Класс LAN

| Описание: Класс для коммуникационных интерфейсов робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| macAddress | String | M | 1 | Способ получения настроек: DHCP или статический IP-адрес |
| ipv4Config | Ipv4Config | O | 1 | Конфигурация IPv4 |
| Ipv6Config | Ipv6Config | O | 1 | Конфигурация IPv6 |
| dnsServer | String | O | N | Список DNS-серверов |
| 8021x | 8021x | O | 1 | Конфигурация безопасности стандарта 802.1X |
| Методы: | | | | |
| setDns(dnsServer: String[]) | MethodResult | O | Задание списка DNS-серверов | |

Класс WLAN приведен в таблице 21.

Таблица 21 — Класс WLAN

| Описание: Конфигурация подключения по Wi-Fi | | | | |
|---|---|---|---|---|
| Выведен из: LAN | | | | |
| Атрибуты: | | | | |
| ssid | String | M | 1 | Уникальное имя сети |
| bssid | String | O | 1 | MAC-адрес точки доступа |
| hiddenNetwork | Boolean | O | 1 | Скрытая ли сеть |
| security | WifiSecurity | M | 1 | Режим аутентификации |
| encryption | WifiEncryption | M | 1 | Способ шифрования пакетов |
| passphrase | String | O | 1 | Пароль сети |
| Методы: | | | | |
| setWifiParameters(ssid: String, bssid: String, security: WifiSecurity, encryption: WifiEncryption, passphrase: String) | MethodResult | M | Задание параметров интерфейса Wi-Fi | |

Класс Communication приведен в таблице 22.

Таблица 22 — Класс Communication

| Описание: Класс для коммуникационных настроек робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| externalInterface | CommunacationInterface | M | 1 | Активный коммуникационный интерфейс |
| interface | CommunacationInterface | M | N | Последовательный порт |
| Методы: | | | | |
| setExternalInterface (interfaceId: Guid) | MethodResult | M | Задание коммуникационного интерфейса для внешнего взаимодействия | |

Примечания

1 Робот должен обладать не менее одним из возможных классов коммуникационного интерфейса

2 Класс Communication включает в себя базовые и наиболее распространенные интерфейсы взаимодействия с роботами. Более специфичные интерфейсы могут быть описаны в ИМ для конкретных типов роботов.

Перечисление ClockSource приведено в таблице 23.

Таблица 23 — Перечисление ClockSource

| Описание: Типы источников данных о времени | |
|---|---|
| Атрибуты: | |
| MANUAL | Задают вручную, поддерживается RTC контроллера |
| NTP | Получается через NTP |
| PTP | Получается через PTP |

Класс DateTimeSettings приведен в таблице 24.

Таблица 24 — Класс DateTimeSettings

| Описание: Класс для настроек даты и времени робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| clockSource | ClockSource | M | 1 | Тип источника установки времени |
| ntpServer | String | O | N | Список NTP-серверов |
| Методы: | | | | |
| setClockSource (clockSource: ClockSource) | MethodResult | M | Задание источника получения времени: NTP или ручная установка | |
| setTime(time: DateTime) | MethodResult | M | Установка текущего времени на роботе | |
| setNtpServer(ntpServer: String[]) | MethodResult | O | Задание списка NTP-серверов для синхронизации времени контроллера робота | |

### 4.5.3 Класс для состояния робота

Информация о состоянии, работоспособности и актуальных параметрах функционирования робота в целом и его компонентов должна быть определена в классе Diagnostic, представленном на рисунке 5 и в таблице 25.

Таблица 25 — Класс Diagnostic

| Описание: Класс диагностики робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| state | State | M | 1 | Текущее состояние робота и его компонентов |
| error | Error | M | N | Список текущих ошибок |
| maintenance | Maintenance | O | 1 | Параметры обслуживания робота |

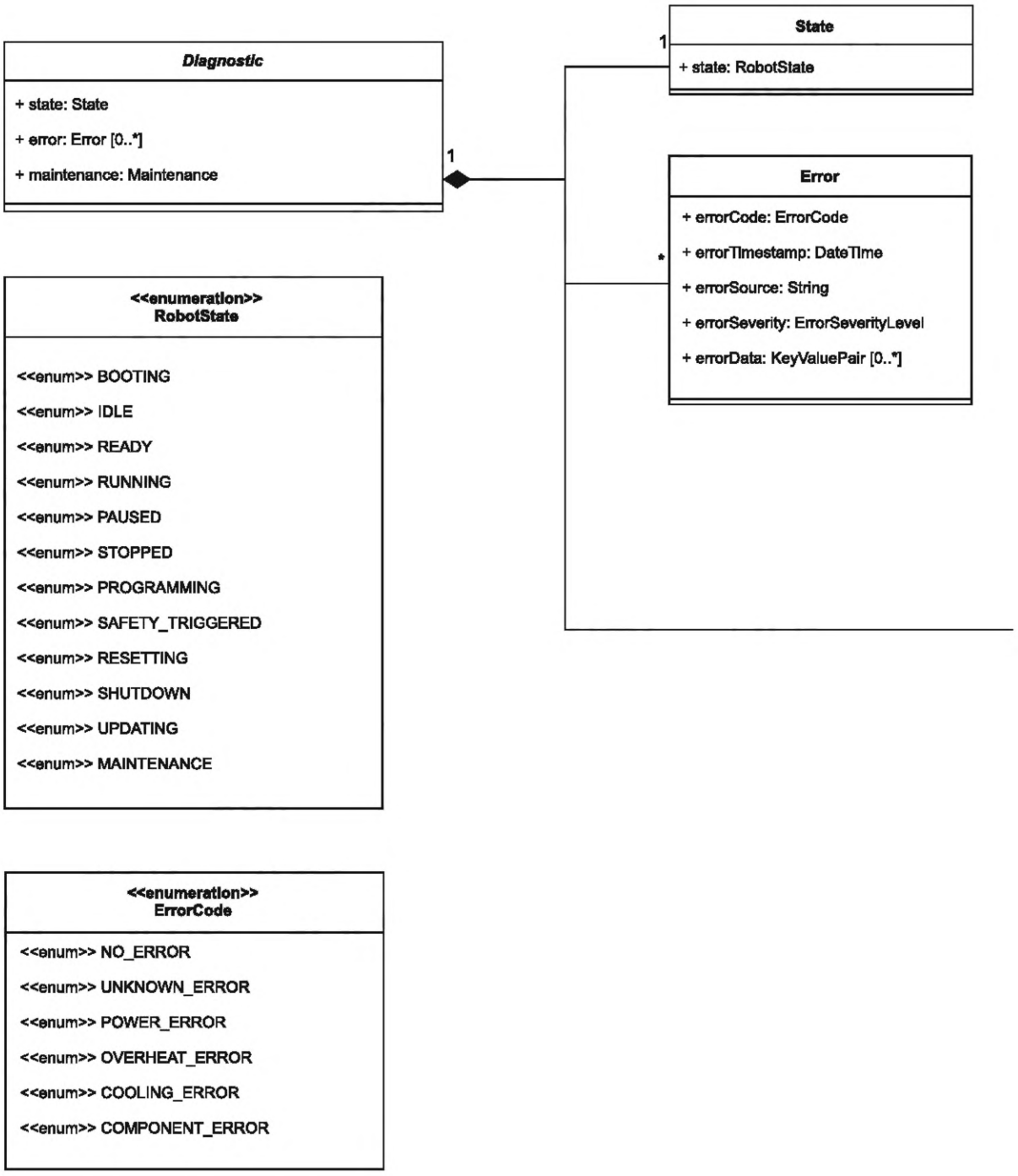


Рисунок 5 — Диаграмма классов для класса Diagnostic, лист 1

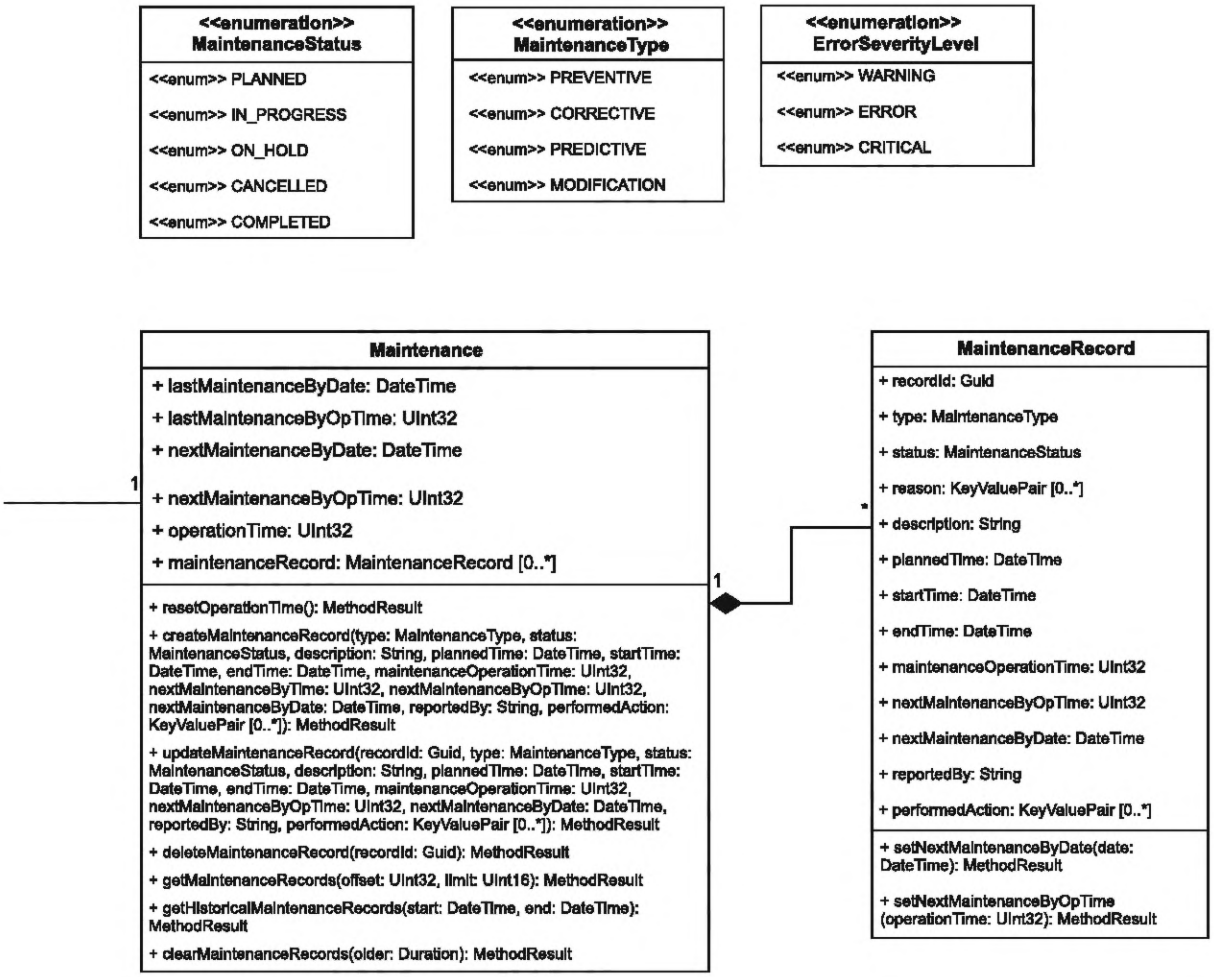


Рисунок 5, лист 2

Класс Diagnostic представляет текущее состояние компонентов и робота в целом, а также актуальные данные по обслуживанию и ошибкам, которые могут возникнуть в процессе работы робота.

Перечисление ErrorSeverityLevel представлено в таблице 26.

Таблица 26 — Перечисление ErrorSeverityLevel

| Описание: Уровни ошибок | |
|---|---|
| Атрибуты: | |
| WARNING | Предупреждение |
| ERROR | Ошибка |
| CRITICAL | Аварийная ситуация |

Перечисление ErrorCode представлено в таблице 27.

Таблица 27 — Перечисление ErrorCode

| Описание: Коды ошибок | |
|---|---|
| Атрибуты: | |
| NO_ERROR | Ошибки отсутствуют |
| UNKNOWN_ERROR | Неизвестная ошибка |
| POWER_ERROR | Ошибка питания |
| OVERHEAT_ERROR | Ошибка перегрева |
| COOLING_ERROR | Ошибка охладительной системы |
| COMPONENT_ERROR | Ошибка компонента робота |

Перечисление MaintenanceType представлено в таблице 28.

Таблица 28 — Перечисление MaintenanceType

| Описание: Тип обслуживания робота | |
|---|---|
| Атрибуты: | |
| PREVENTIVE | Плановое обслуживание |
| CORRECTIVE | Внеплановое (аварийное) обслуживание |
| PREDICTIVE | Прогнозируемое обслуживание |
| MODIFICATION | Плановая модификация |

Перечисление MaintenanceStatus представлено в таблице 29.

Таблица 29 — Перечисление MaintenanceStatus

| Описание: Статус обслуживания робота | |
|---|---|
| Атрибуты: | |
| PLANNED | Запланировано |
| IN_PROGRESS | В работе |
| ON_HOLD | На удержании/ожидание |
| CANCELLED | Отменено |
| COMPLETED | Выполнено |

Класс Error представлен в таблице 30.

Таблица 30 — Класс Error

| Описание: Ошибка робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| errorCode | ErrorCode | M | 1 | Код ошибки |
| errorTimestamp | DateTime | M | 1 | Время возникновения ошибки |

*Окончание таблицы 30*

| | | | | |
|---|---|---|---|---|
| errorSource | String | M | 1 | Источник возникновения ошибки |
| errorSeverity | ErrorSeverityLevel | M | 1 | Уровень ошибки |
| errorData | KeyValuePair | O | N | Данные ошибки |

Класс MaintenanceRecord представлен в таблице 31.

Таблица 31 — Класс MaintenanceRecord

| Описание: Запись о техническом обслуживании | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| recordId | Guid | M | 1 | Уникальный идентификатор записи |
| type | MaintenanceType | M | 1 | Тип обслуживания робота |
| status | MaintenanceStatus | M | 1 | Текущий статус обслуживания робота (конкретной записи) |
| reason | KeyValuePair [0..*] | M | 1 | Причина обслуживания робота |
| description | String | M | 1 | Описание текущего обслуживания |
| plannedTime | DateTime | O | 1 | Запланированное время обслуживания робота |
| startTime | DateTime | M | 1 | Время начала обслуживания робота |
| endTime | DateTime | O | 1 | Время окончания обслуживания робота |
| maintenanceOperationTime | UInt32 | O | 1 | Наработка робота при проведении обслуживания в секундах |
| nextMaintenanceByOpTime | DateTime | O | 1 | Время следующего обслуживания робота по наработке |
| nextMaintenanceByDate | DateTime | O | 1 | Время следующего обслуживания робота по времени |
| reportedBy | String | O | 1 | Ответственный за выполнение обслуживания |
| performedAction | KeyValuePair [0..*] | O | 1 | Выполненные действия в рамках проведения обслуживания |
| Методы: | | | | |
| setNextMaintenanceByDate (date: DateTime) | MethodResult | O | Задание времени проведения следующего планового обслуживания робота по времени | |
| setNextMaintenanceByOpTime (operationTime: UInt32) | MethodResult | O | Задание времени проведения следующего планового обслуживания робота по наработке | |
| Примечания<br>1 Параметр reason представляет собой набор причин выполнения технического обслуживания. Данный набор может содержать в себе указание компонента робота и кодов ошибок данного компонента.<br>2 Параметр performedAction представляет собой набор действий, который применен при обслуживании робота с указанием компонента робота действий над ним. | | | | |

Класс Maintenance представлен в таблице 32.

Таблица 32 — Класс Maintenance

| Описание: Класс параметров обслуживания робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| lastMaintenanceByDate | DateTime | O | 1 | Время последнего обслуживания робота |
| lastMaintenanceByOpTime | UInt32 | O | 1 | Наработка робота при проведении последнего обслуживания |
| nextMaintenanceByDate | DateTime | O | 1 | Плановое время следующего обслуживания робота |
| nextMaintenanceByOpTime | UInt32 | O | 1 | Плановая наработка робота для проведения следующего обслуживания |
| operationTime | UInt32 | O | 1 | Наработка робота, с |
| maintenanceRecord | MaintenanceRecord | O | N | Журнал технического обслуживания |
| Методы: | | | | |
| resetOperationTime() | MethodResult | O | Установка текущей наработки робота в 0 с | |
| createMaintenanceRecord(type: MaintenanceType, status: MaintenanceStatus, description: String, plannedTime: DateTime, startTime: DateTime, endTime: DateTime, maintenanceOperationTime: UInt32, nextMaintenanceByOpTime: UInt32, nextMaintenanceByDate: DateTime, reportedBy: String, performedAction: KeyValuePair [0..*]) | MethodResult | O | Создание записи о техническом обслуживании в журнале | |
| updateMaintenanceRecord(record Id: Guid, type: MaintenanceType, status: MaintenanceStatus, description: String, plannedTime: DateTime, startTime: DateTime, endTime: DateTime, maintenanceOperationTime: UInt32, nextMaintenanceByOpTime: UInt32, nextMaintenanceByDate: DateTime, reportedBy: String, performedAction: KeyValuePair [0..*]) | MethodResult | O | Редактирование записи о техническом обслуживании в журнале | |
| deleteMaintenanceRecord (recordId: Guid) | MethodResult | O | Удаление записи о техническом обслуживании в журнале | |
| getMaintenanceRecords (offset: UInt32, limit: UInt16) | MethodResult | O | Получение последних записей о техническом обслуживании | |
| getHistoricalMaintenanceRecords (start: DateTime, end: DateTime) | MethodResult | O | Получение записей о техническом обслуживании за определенный промежуток времени | |
| clearMaintenanceRecords (older: Duration) | MethodResult | O | Очистка журнала технического обслуживания | |

Для хранения актуальной информации о текущем состоянии компонентов и робота в целом используют объект класса State, представленный в таблице 33.

Таблица 33 — Класс State

| Описание: Класс текущего состояния робота и его компонентов | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| state | RobotState | M | 1 | Текущее состояние робота |

### 4.5.4 Класс для системы безопасности

Класс SafetyState является фундаментальным компонентом системы безопасности робота. Данный класс предназначен для хранения и обработки информации о текущем состоянии безопасности робота. Информация о текущем состоянии системы безопасности робота определена в классе SafetyState, представленном на рисунке 6 и в таблице 34.

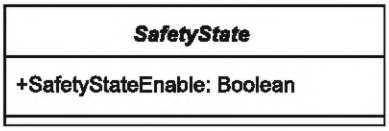


Рисунок 6 — Диаграмма классов для класса SafetyState

Основные характеристики класса:
- является базовым классом для всех типов роботов;
- содержит информацию о состоянии безопасности;
- реализует единый интерфейс для работы с параметрами безопасности.

ОИМ не предъявляет требования к содержанию класса SafetyState в силу наличия большого количества типов роботов со своей спецификой работы. ОИМ обозначает требование к наличию такого объекта в рамках класса Robot для всех типов роботов, но конкретное содержание класса определяется спецификой каждого типа робота путем расширения данного класса через механизм наследования.

Таблица 34 — Класс SafetyState

| Описание: Класс системы безопасности робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| safetyStateEnable | Boolean | M | 1 | Текущее состояние робота |

### 4.5.5 Класс для событий робота

Информация о текущих событиях работы робота определена в классе MessageEvent, представленном на рисунке 7 и в таблице 35. Класс MessageEvent представляет последнее событие, сгенерированное роботом, и историю событий, хранящихся непосредственно на роботе. Структура класса включает в себя атрибуты для хранения данных о событиях, методы для работы с событиями, механизмы записи и чтения истории событий.

Практическое применение: отслеживание действий робота в реальном времени, анализ истории работы для выявления нештатных ситуаций, формирование отчетов о работе робота, диагностика неисправностей на основе анализа событий.

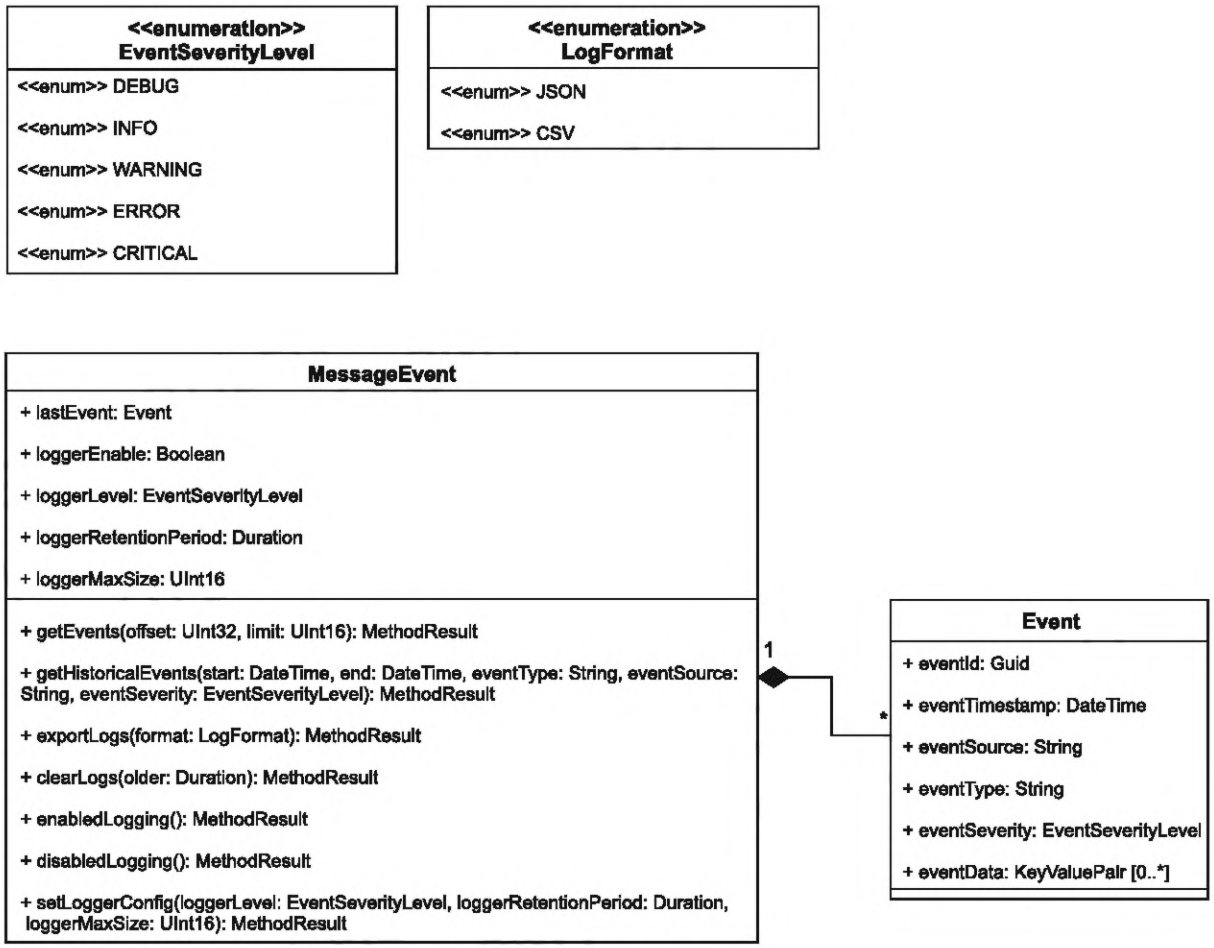


Рисунок 7 — Диаграмма классов для класса MessageEvent

Класс MessageEvent представлен в таблице 35.

Таблица 35 — Класс MessageEvent

| Описание: Класс для работы с событиями и логами робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| lastEvent | Event | M | 1 | Последнее событие робота |
| loggerEnable | Boolean | O | 1 | Флаг, указывающий активность записи данных на диск |
| loggerLevel | EventSeverityLevel | O | 1 | Минимальный уровень события, которые записывают |
| loggerRetentionPeriod | Duration | O | 1 | Срок хранения исторических данных (см. ГОСТ Р 7.0.64) |
| loggerMaxSize | UInt32 | O | 1 | Максимальный размер данных на диске в КБайт |

*Окончание таблицы 35*

| Методы: | | | |
|---|---|---|---|
| getEvents(offset: UInt32, limit: UInt16) | MethodResult | O | Получение последних событий из памяти |
| getHistoricalEvents (start: DateTime, end: DateTime, eventType: String, eventSource: String, eventSeverity: EventSeverityLevel) | MethodResult | O | Получение исторических событий с указанием диапазона временем и фильтрацией по типу, источнику и уровню события |
| exportLogs(offset: UInt32, limit: UInt16, format: LogFormat) | MethodResult | O | Экспорт хранящихся логов в заданный формат в заданном диапазоне номеров записей |
| clearLogs(older: Duration) | MethodResult | O | Очистка старых записей. Аргумент older указывает на длительность, данные старше указанной длительности от момента вызова метода должны быть удалены |
| enableLogging() | MethodResult | O | Активация системы логирования |
| disableLogging() | MethodResult | O | Деактивация системы логирования |
| setLoggerConfig(loggerLevel: EventSeverityLevel, loggerRetentionPeriod: Duration, loggerMaxSize: UInt32) | MethodResult | O | Изменение параметров логирования. logger MaxSize указывается в КБайт |

Перечисление EventSeverityLevel представлено в таблице 36.

Таблица 36 — Перечисление EventSeverityLevel

| Описание: Уровни событий | |
|---|---|
| Атрибуты: | |
| DEBUG | Техническая информация |
| INFO | Информационные сообщения |
| WARNING | Предупреждения |
| ERROR | События, требующие вмешательства |
| CRITICAL | Аварийные ситуации |

Перечисление LogFormat представлено в таблице 37.

Таблица 37 — Перечисление LogFormat

| Описание: Формат данных выгрузки логов | |
|---|---|
| Атрибуты: | |
| JSON | Данные представлены в формате JSON |
| CSV | Данные представлены в формате CSV |

Класс Event представлен в таблице 38.

Таблица 38 — Класс Event

| Описание: Формат события робота | | | | |
|---|---|---|---|---|
| Выведен из: Неприменимо | | | | |
| Атрибуты: | | | | |
| eventId | Guid | M | 1 | Уникальный идентификатор события |
| eventTimestamp | DateTime | M | 1 | Время возникновения события |
| eventSource | String | M | 1 | Источник возникновения события |
| eventType | String | M | 1 | Тип события |
| eventSeverity | EventSeverityLevel | M | 1 | Уровень события |
| eventData | KeyValuePair | O | N | Данные события |

**Приложение А**
**(обязательное)**

## Правила присвоения имен

### А.1 Общие положения

Правила присвоения имен выведены на основе руководств и принципов, описанных в ГОСТ Р ИСО/МЭК 11179-5. Данные правила были приспособлены для конкретного применения в информационной модели. Они разработаны для обеспечения единообразного и адекватного именования как базовых, так и агрегированных информационных объектов.

Базовый информационный объект используют для представления информации и построения информационных моделей, а его примерами являются класс объекта, атрибут класса объекта, метод объекта.

### А.2 Правила присвоения имен

Применяют следующие правила присвоения имен:

- правило 1. Имя элемента должно быть уникальным в логической группировке данных, к которой принадлежит данный элемент;
- правило 2. Имя класса допускается начинать с прописной буквы. Имя члена класса рекомендуется начинать со строчной буквы;
- правило 3. Имя класса не должно содержать последовательных дублирующих слов;
- правило 4. Рекомендуется, чтобы имя элемента содержало менее 10 символов при максимальной длине 256 символов. При необходимости сокращение имени элемента может быть определено и использовано;
- правило 5. В имени элемента допускается использовать буквенно-цифровые символы, если они не определены особо;
- правило 6. Имя элемента должно быть представлено в единственном числе, если его смысл не предполагает множественного числа (например, goods).

## Приложение Б
## (справочное)

## Формат представления класса

Классы определены в настоящем стандарте с использованием табличного формата, представленного в таблице Б.1. В таблице Б.1 буквы «М» и «О» обозначают «обязательный (mandatory)» и «факультативный (optional)» соответственно.

Таблица Б.1 — Табличный формат для представления класса

| | | | | |
|---|---|---|---|---|
| Описание: <Описание класса> | | | | |
| Выведен из: <Имя порождающего класса> или <неприменимо> | | | | |
| Атрибуты | | | | |
| <Имя атрибута> | <Тип атрибута> | Обязательный (М) или факультативный (О) | Число объектов | Описание |
| | | | | |
| Методы | | | | |
| <Имя метода с типами и именами параметров> | <Тип возвращаемого значения> | Обязательный или факультативный | Описание | |
| | | | | |

Атрибутами перечисляемого типа данных являются перечисляемые значения или константы. Пример представления класса MyClass приведен в таблице Б.2.

Таблица Б.2 — Пример представления класса MyClass

| | | | | |
|---|---|---|---|---|
| Описание: Это пример. | | | | |
| Выведен из: ОИМ | | | | |
| Атрибуты | | | | |
| <Имя атрибута> | <Тип атрибута> | Обязательный или факультативный | Число объектов | Описание |
| attribute1 | String | M | 1 | Данный атрибут использован для примера |
| attribute2 | Double | M | 1 | Данный атрибут использован для примера |
| attribute3 | UInt16 | 0 | N | Данный атрибут имеет тип списка из N объектов |
| Методы | | | | |
| <Имя метода с типами и именами параметров> | <Тип возвращаемого значения> | Обязательный или факультативный | Описание | |
| method1() | MethodResult | M | Описание метода method1 | |
| method2 (paraName1: UInt16, paraName2: String) | MethodResult | M | Описание метода method2 с параметрами | |
| Примечание — Типы параметров методов отделены двоеточием «:». | | | | |

### Формат представления перечисления

Перечисления определены в настоящем стандарте с использованием табличного формата, представленного в таблице Б.3.

Таблица Б.3 — Табличный формат для представления класса

| Описание: Это пример. | |
|---|---|
| Атрибуты: | |
| <Имя атрибута> | Описание |
| | |

Пример представления перечисления MyEnumeration приведен в таблице Б.4.

Таблица Б.4 — Пример представления перечисления MyEnumeration

| Описание: <Описание перечисления> | |
|---|---|
| Атрибуты: | |
| ATTRIBUTE1 | Данный атрибут использован для примера |
| | |

## Приложение В
## (справочное)

## Рекомендации по применению общей информационной модели

ОИМ роботов, установленная в настоящем стандарте, является единым понятийным и структурным основанием для конкретных ИМ, разрабатываемых для различных типов и классов роботов. ОИМ является абстрактной моделью, не предназначенной для непосредственной реализации. Пользователям необходимо использовать конкретные ИМ для определенных типов роботов.

ОИМ — это родительская структура для всех последующих ИМ роботов (промышленного, сервисного и т. д.). Все конкретные ИМ должны непосредственно наследовать классы, атрибуты, методы и логику ОИМ, добавляя свои специализированные атрибуты, методы и связи — без изменения единого основного ядра и парадигмы построения модели.

ОИМ применяют для:

- формирования требований и разработки ИМ конкретных типов роботов;
- стандартизации обмена данными между роботами различных изготовителей;
- стандартизации обмена данными между роботами и смежными ИС и АСУ;
- обеспечения восходящей совместимости при эволюции стандартов.

ОИМ неприменима для прямой реализации и подлежит конкретизации на уровне дочерних моделей, ОИМ не описывает специфику обмена сообщениями, форматов данных, API и сетевых протоколов, а также не содержит требований к реализации бизнес-логики и безопасности.

Поскольку ОИМ — формальная основа для дочерних расширенных моделей, информацию в ней применяют на этапах, указанных в таблице В.1

Таблица В.1 — Этапы применения ОИМ

| Этап/класс ОИМ | Проектирование | Разработка | Эксплуатация | Техобслуживание |
|---|---|---|---|---|
| Feature | M | M | — | — |
| Diagnostic | M | M | — | — |
| SafetyState | M | M | — | — |
| MessageEvent | M | M | — | — |

На этапе «Проектирование» осуществляется применение ОИМ как перечня обязательных и опциональных классов для построения модели любого типа робота, разработка архитектуры будущих ИМ.

На этапе «Разработка» осуществляется пошаговая конкретизация (наследование, расширение, детализация) классов и атрибутов в соответствии с требованиями целевого применения, формирование дочерних информационных моделей.

На этапах «Эксплуатация» и «Техобслуживание» конечный пользователь взаимодействует с дочерней ИМ (при условии непосредственной эксплуатации и техобслуживания), описывающей конкретный тип роботов.

УДК 621.865.8:007.52:006.354 ОКС 25.040.30

Ключевые слова: роботы, робототехнические устройства, общая информационная модель

Редактор *Е.Ю. Митрофанова*
Технический редактор *И.Е. Черепкова*
Корректоры *Е.Д. Дульнева, М.И. Першина*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 01.12.2025. Подписано в печать 01.12.2025. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 4,18. Уч.-изд. л. 3,47.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru