

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата

09.03.04 - Программная инженерия

Отчет по верификации программного обеспечения

API ДЛЯ МОДУЛЬНОЙ СИСТЕМЫ СБОРА И АНАЛИЗА  
ДААННЫХ

Выполнил:

студент 3 курса группы 22307

Н. А. Федорец

Руководитель:

К. А. Кулаков, к.ф.-м.н., доцент

# Содержание

<b>1</b>	<b>Объект тестирования</b>	<b>3</b>
1.1	Описание программы . . . . .	3
1.2	Архитектура программы . . . . .	3
<b>2</b>	<b>Стратегия тестирования</b>	<b>4</b>
2.1	Структура объекта тестирования . . . . .	4
2.2	Стратегия аттестационного тестирования . . . . .	4
2.3	Стратегия блочного тестирования . . . . .	4
2.4	Стратегия интеграционного тестирования . . . . .	4
<b>3</b>	<b>План тестирования</b>	<b>5</b>
3.1	Аттестационные тесты . . . . .	5
3.2	Блочные тесты . . . . .	19
3.3	Интеграционные тесты . . . . .	21
<b>4</b>	<b>Журнал тестирование</b>	<b>22</b>
<b>5</b>	<b>Пример тестов</b>	<b>24</b>
5.1	Аттестационные . . . . .	24
5.2	Модульные . . . . .	24
5.3	Интеграционные . . . . .	25

# 1 Объект тестирования

## 1.1 Описание программы

Применяется в автоматизированных системах цифровой диагностики, прогностной аналитики и пост-анализа на основе данных многопараметрического мониторинга объектов промышленного производства и транспорта.

## 1.2 Архитектура программы

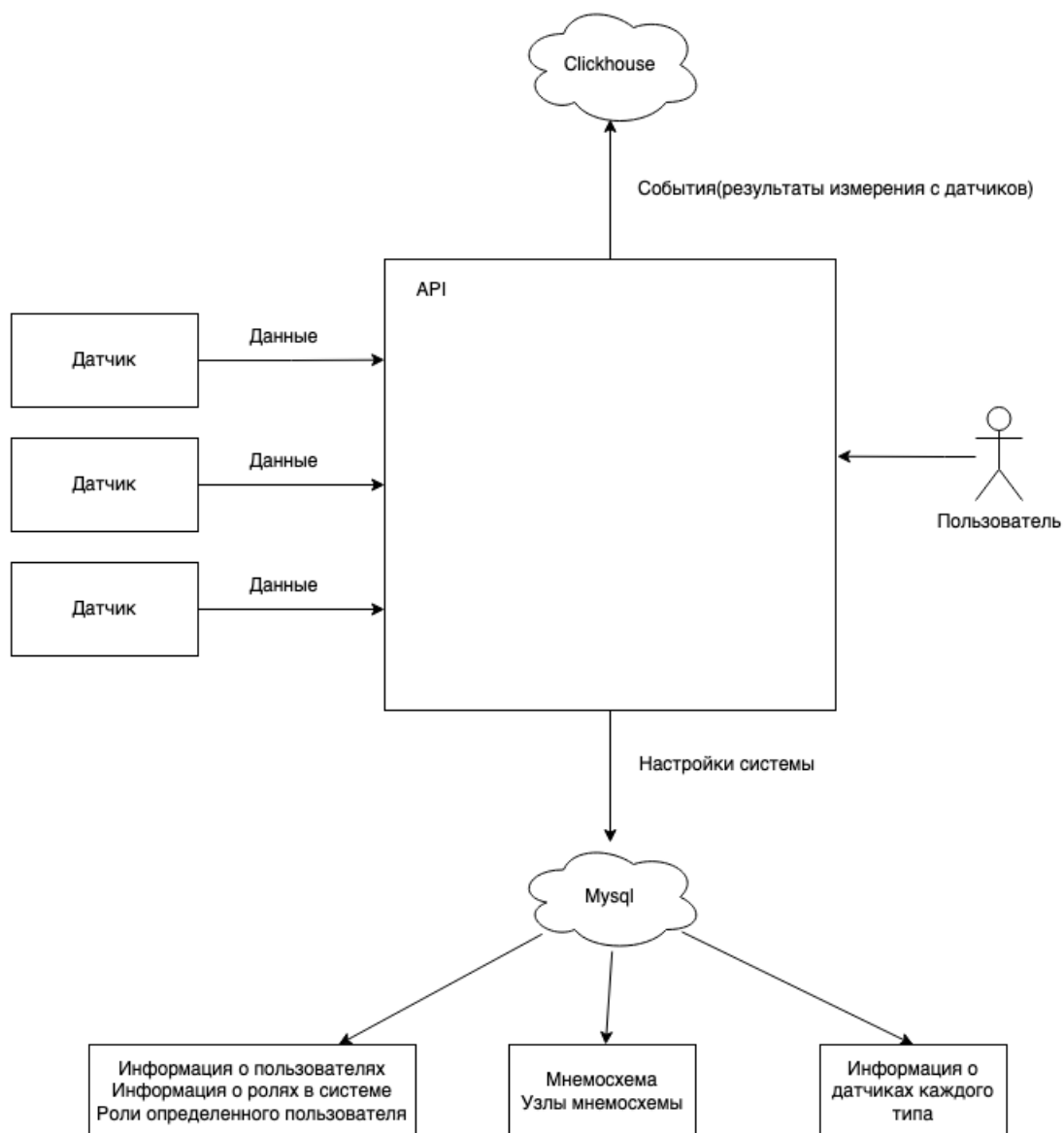


Рис. 1: Архитектура программа

## 2 Стратегия тестирования

### 2.1 Структура объекта тестирования

В качестве объекта тестирования был выбран программный модуль определения статуса – heater monitor modules

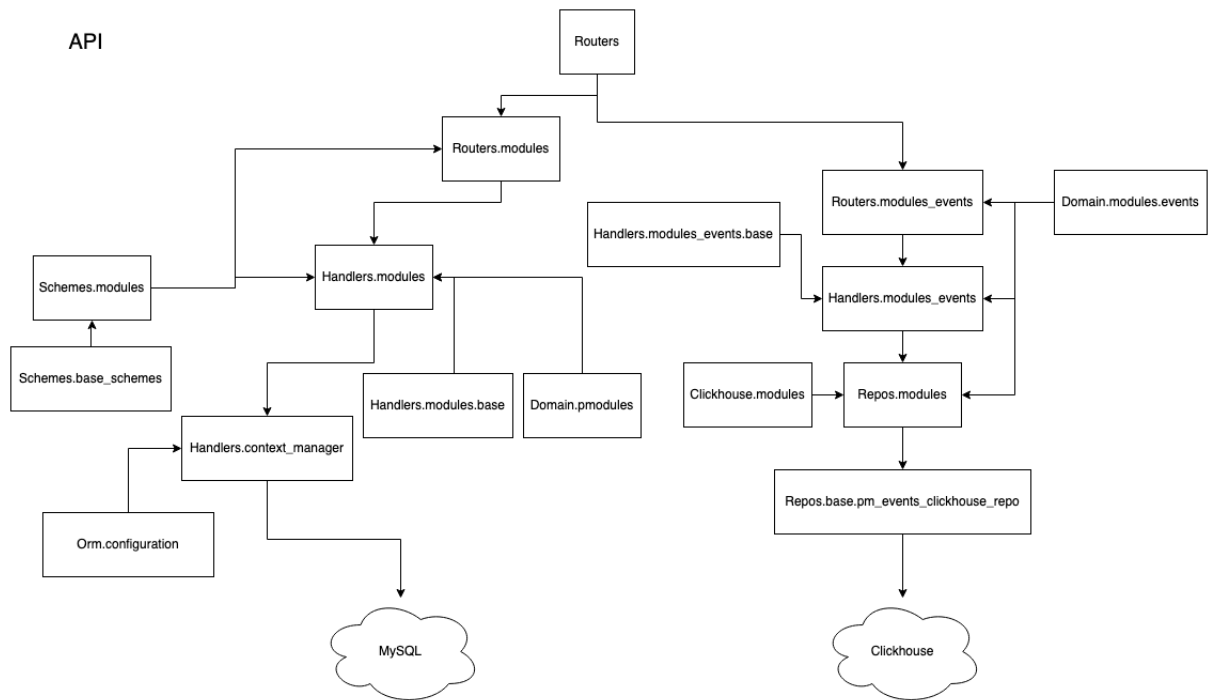


Рис. 2: Архитектура модуля

### 2.2 Стратегия аттестационного тестирования

Нужно вручную провести тестирование http-запросов модуля после запуска системы через интерфейс Fast API.

### 2.3 Стратегия блочного тестирования

Провести тестирование функций `update()`, расположенная в `app/handlers/base.py` и функций `update_by_guid()`, расположенная в `app/handlers/modules/base.py`.

### 2.4 Стратегия интеграционного тестирования

Проверить поведение системы при коллизии `guid`.

## 3 План тестирования

### 3.1 Аттестационные тесты

#### Тест А1

**Тип теста:** Положительный

**Описание теста:** Проверка работы GET запроса для получения программных модулей определения статусов

**Начальное состояние:** Список программных модулей не пуст

**Входные данные:** Нет

**Ожидаемый результат:** Список программных модулей определения статусов

#### Тест А2

**Тип теста:** Положительный

**Описание теста:** Проверка работы POST запроса для добавления программного модуля определения статусов в соответствии с заданными значениями

**Начальное состояние:** Любое

**Входные данные:**

```
{
  "name": "string",
  "description": "string",
  "node_id": 0,
  "error_code": 0,
  "states": "string"
}
```

**Ожидаемый результат:**

```
{
  "id": 1,
  "name": "string",
  "description": "string",
  "guid": "4ea03b03-4815-46c6-be10-f6908a006640",
  "status": "",
}
```

```
"process_status": 2,  
"process_status_description": "",  
"node_id": 0,  
"alias": "5.4",  
"x": null,  
"y": null,  
"error_code": 0,  
"states": "string"  
}
```

### Тест А3

**Тип теста:** Негативный

**Описание теста:** Проверка работы POST запроса для добавления программного модуля определения статусов соответствия с неправильными заданными значениями.

**Начальное состояние:** Любое

**Входные данные:**

```
{}
```

**Ожидаемый результат:**

```
{  
    "detail": "string"  
}
```

### Тест А4

**Тип теста:** Положительный

**Описание теста:** Проверка работы GET запроса для получения программного модуля определения статусов по id

**Начальное состояние:** Список программных модулей не пуст

**Входные данные:** id

**Ожидаемый результат:**

```
{  
    "id": 2,  
    "name": "string",
```

```
"description": "string",
"guid": "b2165ea7-87b2-4ca9-947b-d3b0e803bb84",
"status": "",
"process_status": 2,
"process_status_description": "",
"node_id": null,
"alias": "2",
"x": null,
"y": null,
"error_code": 0,
"states": "string"
}
```

## Тест А5

**Тип теста:** Негативный

**Описание теста:** Проверка работы GET запроса для получения программного модуля определения статусов по id при условии отсутствия модуля с заданным id

**Начальное состояние:** Отсутствует модуль с заданным id

**Входные данные:** id

**Ожидаемый результат:**

```
{
  "detail": "Element id = @id not found"
}
```

## Тест А6

**Тип теста:** Положительный

**Описание теста:** Проверка работы PUT запроса для изменения параметров программного модуля определения статусов по id в соответствии с заданными значениями

**Начальное состояние:**

```
{
  "id": 0,
  "name": "string",
}
```

```
"description": "string",
"guid": "a3758b0d-bb61-45f3-8a5a-38d813fec431",
"status": "string",
"process_status": 1,
"process_status_description": "",
"node_id": 0,
"alias": "5.0",
"x": 0,
"y": 0,
"error_code": 0,
"states": "string"
}
```

**Входные данные:** id,

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "process_status": 1,
  "status": "string",
  "node_id": 0,
  "x": 1,
  "y": 0,
  "error_code": 0,
  "states": "string"
}
```

**Ожидаемый результат:**

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "guid": "a3758b0d-bb61-45f3-8a5a-38d813fec431",
  "status": "string",
  "process_status": 1,
```



```
"process_status_description": "",
"node_id": 0,
"alias": "5.0",
"x": 1,
"y": 0,
"error_code": 0,
"states": "string"
}
```

## Тест А7

**Тип теста:** Негативный

**Описание теста:** Проверка работы PUT запроса для изменения параметров программного модуля определения статусов по id при условии отсутствия модуля с заданным id

**Начальное состояние:** Отсутствует модуль с заданным id

**Входные данные:** id,

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "process_status": 1,
  "status": "string",
  "node_id": 0,
  "x": 0,
  "y": 0,
  "error_code": 0,
  "states": "string"
}
```

**Ожидаемый результат:**

```
{
  "detail": "Element id = @id not found"
}
```

## Тест А8

**Тип теста:** Положительный

**Описание теста:** Проверка работы DELETE запроса для удаления программного модуля определения статусов по id

**Начальное состояние:** Список программных модулей не пуст

**Входные данные:** id

**Ожидаемый результат:** 0

## Тест А9

**Тип теста:** Негативный

**Описание теста:** Проверка работы DELETE запроса для удаления программного модуля определения статусов по id при условии отсутствия модуля с заданным id

**Начальное состояние:** Список программных модулей не пуст

**Входные данные:** id

**Ожидаемый результат:**

```
{  
  "detail": "Element id = @id not found"  
}
```

## Тест А10

**Тип теста:** Положительный

**Описание теста:** Проверка работы GET запроса для получения программного модуля определения статусов по guid

**Начальное состояние:** Список программных модулей не пуст

**Входные данные:** guid

**Ожидаемый результат:**

```
{  
  "process_status": 2,  
  "id": 2,  
  "guid": "b2165ea7-87b2-4ca9-947b-d3b0e803bb84",
```

```
"node_id": null ,
"y": null ,
"states": "string",
"description": "string",
"name": "string",
"status": "",
"x": null ,
"error_code": 0,
"alias": "2"
}
```

### Тест A11

**Тип теста:** Негативный

**Описание теста:** Проверка работы GET запроса для получения программного модуля определения статусов по guid при условии отсутствия модуля с заданным guid

**Начальное состояние:** Отсутствует модуль с заданным guid

**Входные данные:** guid

**Ожидаемый результат:**

```
{
  "detail": "Element guid = @guid not found"
}
```

### Тест A12

**Тип теста:** Положительный

**Описание теста:** Проверка работы PUT запроса для изменения параметров программного модуля определения статусов по guid в соответствии с заданными значениями

**Начальное состояние:**

```
{
  "id": 2,
  "name": "string",
  "description": "string",
  "guid": "b2165ea7-87b2-4ca9-947b-d3b0e803bb84",
}
```

```
"status": "",
"process_status": 2,
"process_status_description": "",
"node_id": null,
"alias": "2",
"x": null,
"y": null,
"error_code": 0,
"states": "string"
}
```

**Входные данные:** guid,

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "process_status": 1,
  "status": "string",
  "node_id": 0,
  "x": 0,
  "y": 0,
  "error_code": 0,
  "states": "string"
}
```

**Ожидаемый результат:**

```
{
  "process_status": 1,
  "id": 0,
  "guid": "b2165ea7-87b2-4ca9-947b-d3b0e803bb84",
  "node_id": 0,
  "y": 0,
  "states": "string",
  "description": "string",
  "name": "string",
}
```

```
"status": "string",
"x": 0,
"error_code": 0,
"alias": "5.0"
}
```

### Тест A13

**Тип теста:** Негативный

**Описание теста:** Проверка работы PUT запроса для изменения параметров программного модуля определения статусов по guid при условии отсутствия модуля с заданным guid

**Начальное состояние:** Отсутствует модуль с заданным guid

**Входные данные:** guid,

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "process_status": 1,
  "status": "string",
  "node_id": 0,
  "x": 0,
  "y": 0,
  "error_code": 0,
  "states": "string"
}
```

**Ожидаемый результат:**

```
{
  "detail": "Element guid = @guid not found"
}
```

### Тест A14

**Тип теста:** Положительный

**Описание теста:** Проверка работы PUT запроса для очистки таблиц события программ-

ного модуля определения статусов по uuid

**Начальное состояние:** Список таблиц не пуст

**Входные данные:** uuid **Ожидаемый результат:** Successful Response

### Тест A15

**Тип теста:** Негативный

**Описание теста:** Проверка работы PUT запроса для очистки таблиц события программного модуля определения статусов по uuid при условии отсутствия таблицы с заданным uuid

**Начальное состояние:** Отсутствует таблица с заданным uuid

**Входные данные:** uuid **Ожидаемый результат:**

```
{  
  "detail": "Element guid = @guid not found"  
}
```

### Тест A16

**Тип теста:** Положительный

**Описание теста:** Проверка работы POST запроса для добавления таблиц события программного модуля определения статусов по uuid

**Начальное состояние:** Список таблиц не пуст

**Входные данные:** uuid

**Ожидаемый результат:** Successful Response

### Тест A17

**Тип теста:** Негативный

**Описание теста:** Проверка работы POST запроса для добавления таблиц события программного модуля определения статусов по uuid при условии отсутствия таблицы с заданным uuid

**Начальное состояние:** Отсутствует таблица с заданным uuid

**Входные данные:** uuid

**Ожидаемый результат:**

```
{
```

```
"detail": "Element guid = @guid not found"
}
```

### Тест A18

**Тип теста:** Положительный

**Описание теста:** Проверка работы DELETE запроса для удаления таблиц события программного модуля определения статусов по uuid

**Начальное состояние:** Список таблиц не пуст

**Входные данные:** uuid

**Ожидаемый результат:** Successful Response

### Тест A19

**Тип теста:** Негативный

**Описание теста:** Проверка работы DELETE запроса для удаления таблиц события программного модуля определения статусов по uuid при условии отсутствия таблицы с заданным uuid

**Начальное состояние:** Отсутствует таблица с заданным uuid

**Входные данные:** uuid

**Ожидаемый результат:**

```
{
  "detail": "Element guid = @guid not found"
}
```

### Тест A20

**Тип теста:** Положительный

**Описание теста:** Проверка работы GET запроса для получения события value\_changed таблицы программного модуля определения статусов по uuid в соответствии с заданными значениями

**Начальное состояние:** Список таблиц не пуст

**Входные данные:** uuid, start\_date, end\_date, skip, limit, filters, order\_by

**Ожидаемый результат:** Successful Response

## Тест A21

**Тип теста:** Негативный

**Описание теста:** Проверка работы GET запроса для получения события value\_changed таблицы программного модуля определения статусов по uuid в соответствии с заданными значениями при условии отсутствия таблицы с заданным uuid

**Начальное состояние:** Отсутствует таблица с заданным uuid

**Входные данные:** uuid, start\_date, end\_date, skip, limit, filters, order\_by

**Ожидаемый результат:**

```
{
  "detail": "Element guid = @guid not found"
}
```

## Тест A22

**Тип теста:** Положительный

**Описание теста:** Проверка работы POST запроса для добавления события value\_changed таблицы программного модуля определения статусов по uuid в соответствии с заданными значениями

**Начальное состояние:** Список таблиц не пуст

**Входные данные:** uuid

```
{
  "timestamp": 0,
  "level": 0,
  "value": 0,
  "unstatus": 0
}
```

**Ожидаемый результат:** Successful Response

## Тест A23

**Тип теста:** Негативный

**Описание теста:** Проверка работы POST запроса для добавления события value\_changed таблицы программного модуля определения статусов по uuid в соответствии с заданными



значениями при условии отсутствия таблицы с заданным uuid

**Начальное состояние:** Отсутствует таблица с заданным uuid

**Входные данные:** uuid

```
{  
  "timestamp": 0,  
  "level": 0,  
  "value": 0,  
  "unstatus": 0  
}
```

**Ожидаемый результат:**

```
{  
  "detail": "Element guid = @guid not found"  
}
```

#### Тест A24

**Тип теста:** Положительный

**Описание теста:** Проверка работы DELETE запроса для удаления события value\_changed таблицы программного модуля определения статусов по uuid в соответствии с заданными значениями

**Начальное состояние:** Список таблиц не пуст

**Входные данные:** uuid **Ожидаемый результат:** Successful Response

#### Тест A25

**Тип теста:** Негативный

**Описание теста:** Проверка работы DELETE запроса для удаления события value\_changed таблицы программного модуля определения статусов по uuid в соответствии с заданными значениями при условии отсутствия таблицы с заданным uuid

**Начальное состояние:** Отсутствует таблица с заданным uuid

**Входные данные:** uuid **Ожидаемый результат:**

```
{  
  "detail": "Element guid = @guid not found"  
}
```

```
}
```

## Тест A26

**Тип теста:** Положительный

**Описание теста:** Проверка работы GET получить типы событий программного модуля определения статусов

**Начальное состояние:** Любое

**Входные данные:** Нет

**Ожидаемый результат:**

```
[  
  {  
    "type": "value_changed",  
    "description": "",  
    "fields": [  
      {  
        "alias": "level",  
        "clhType": "Int8",  
        "description": "",  
        "phys": "-",  
        "type": 1,  
        "unit": "-",  
        "xAxisDescription": "-",  
        "yAxisDescription": "-"  
      },  
      {  
        "alias": "value",  
        "clhType": "Int8",  
        "description": "ID",  
        "phys": "-",  
        "type": 1,  
        "unit": "-",  
        "xAxisDescription": "-",  
        "yAxisDescription": "-"  
      }  
    ]  
  }  
]
```

```

    },
    {
      "alias": "unstatus",
      "clhType": "Int8",
      "description": "(0-3)",
      "phys": "-",
      "type": 1,
      "unit": "-",
      "xAxisDescription": "-",
      "yAxisDescription": "-"
    }
  ]
}
]

```

## 3.2 Блочные тесты

### Тест Б1

**Тип теста:** Положительный

**Описание теста:** Провести тестирование функций update()

**Начальное состояние:** Любое

**Входные данные:** id, scheme

**Ожидаемый результат:**

model == scheme

### Тест Б2

**Тип теста:** Положительный

**Описание теста:** Провести тестирование функций update\_by\_guid()

**Начальное состояние:** Любое

**Входные данные:** guid, scheme

**Ожидаемый результат:**

model == scheme

### Тест Б3

**Тип теста:** Положительный

**Описание теста:** Провести тестирование функций get()

**Начальное состояние:** Любое

**Входные данные:** id

**Ожидаемый результат:**

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "guid": "b2165ea7-87b2-4ca9-947b-d3b0e803bb84",
  "status": "string",
  'states': 'string',
  "process_status": 1,
  "process_status": "[U+FFFD][U+FFFD][U+FFFD][U+FFFD][U+FFFD][U+FFFD][U+FFFD][U+FFFD]",
  "node_id": 0,
  "alias": "5.0",
  "x": 0.0,
  "y": 0.0,
  "error_code": 0
}
```

### Тест Б4

**Тип теста:** Положительный

**Описание теста:** Провести тестирование функций get\_by\_guid()

**Начальное состояние:** Любое

**Входные данные:** guid

**Ожидаемый результат:**

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "guid": "b2165ea7-87b2-4ca9-947b-d3b0e803bb84",
```

```
"status": "string",
'states': 'string',
"process_status": 1,
"process_status": [U+FFFD] [U+FFFD] [U+FFFD] [U+FFFD] [U+FFFD] [U+FFFD] [U+FFFD]
"node_id": 0,
"alias": "5.0",
"x": 0.0,
"y": 0.0,
"error_code": 0
}
```

### 3.3 Интеграционные тесты

#### Тест И1

**Тип теста:** Негативный

**Описание теста:** Исследование поведения программы при коллизии guid двух программных модулей

**Начальное состояние:** Два программных модуля с одинаковым guid

**Входные данные:** guid

**Ожидаемый результат:** Сообщение об ошибке о коллизии guid.

## 4 Журнал тестирование

Номер теста	Результат	Примечание
A1	Пройден	Ручное тестирование
A2	Пройден	Ручное тестирование
A3	Пройден	Ручное тестирование
A4	Пройден	Ручное тестирование
A5	Пройден	Ручное тестирование
A6	Пройден	Ручное тестирование
A7	Пройден	Ручное тестирование
A8	Пройден	Ручное тестирование
A9	Пройден	Ручное тестирование
A10	Пройден	Ручное тестирование
A11	Пройден	Ручное тестирование
A12	Пройден	Ручное тестирование
A13	Пройден	Ручное тестирование
A14	Не пройден	Нет ответа от Fast API
A15	Не пройден	Нет ответа от Fast API

Номер теста	Результат	Примечание
A16	Не пройден	Нет ответа от Fast API
A17	Не пройден	Нет ответа от Fast API
A18	Не пройден	Нет ответа от Fast API
A19	Не пройден	Нет ответа от Fast API
A20	Не пройден	Нет ответа от Fast API
A21	Не пройден	Нет ответа от Fast API
A22	Не пройден	Нет ответа от Fast API
A23	Не пройден	Нет ответа от Fast API
A24	Пройден	Ручное тестирование
Б1	Пройден	Успешно
Б2	Пройден	Успешно
Б3	Пройден	Успешно
Б4	Пройден	Успешно
И1	Не пройден	Коллизия guid никак не обрабатывается

## 5 Пример тестов

### 5.1 Аттестационные

```
def test_valid_id():
    response = client.get("/heater-monitor-modules/0")
    assert response.status_code == 200
    assert response.json() == {"id": 0,
                               "name": "string",
                               "description": "string",
                               "guid": "b2165ea7-87b2-4ca9-947b-d3b0e803bb84",
                               "status": "string",
                               'states': 'string',
                               "process_status": 1,
                               "process_status_description": "Активно",
                               "node_id": 0,
                               "alias": "5.0",
                               "x": 0.0,
                               "y": 0.0,
                               "error_code": 0}
```

Рис. 3: Аттестационный тест

### 5.2 Модульные

```
def test_get_id(mock_my_model):
    my_model, scheme = mock_my_model
    engine = create_engine(config.get_settings().mysql_uri, pool_pre_ping=True)
    session = sessionmaker(autocommit=False, autoflush=True, bind=engine)
    uow = HeaterMonitorModuleUnitOfWork(session)
    handler = HeaterMonitorModulesHandler(uow)
    model = asyncio.run(handler.get(0))
    assert model.id == my_model.id
    assert model.guid == my_model.guid
    assert model.x == my_model.x
    assert model.y == my_model.y
    assert model.name == my_model.name
```

Рис. 4: Модульный тест



### 5.3 Интеграционные

```
def test_guid_collision():
    engine = create_engine(config.get_settings().mysql_uri, pool_pre_ping=True)
    session = sessionmaker(autocommit=False, autoflush=True, bind=engine)
    uow = HeaterMonitorModuleUnitOfWork(session)
    handler = HeaterMonitorModulesHandler(uow)
    model = asyncio.run(handler.get_by_guid('b2165ea7-87b2-4ca9-947b-d3b0e803bb84'))
    assert model == ValidationHTTPException
```

Рис. 5: Интеграционный тест