```
# JsonOutputParser

import pytest
import re
from pydantic import BaseModel
from agentparse import JsonOutputParser


# Define a sample Pydantic model for testing
class MyModel(BaseModel):
    name: str
    age: int


# Test the initialization of JsonOutputParser
def test_json_output_parser_initialization():
    parser = JsonOutputParser(MyModel)
    assert parser.pydantic_object == MyModel
    assert isinstance(parser.pattern, re.Pattern)


# Test parsing valid JSON from text
def test_parse_valid_json():
    parser = JsonOutputParser(MyModel)
    text = ""
    model = parser.parse(text)
```

```
    assert model.name == "John"
    assert model.age == 42


# Test parsing without JSON code block
def test_parse_without_json_code_block():
    parser = JsonOutputParser(MyModel)
    text = '{"name": "John", "age": 42}'
    model = parser.parse(text)
    assert model.name == "John"
    assert model.age == 42


# Test parsing with no matching pattern
def test_parse_no_match_pattern():
    parser = JsonOutputParser(MyModel)
    text = "This is some random text without JSON."

    with pytest.raises(Exception) as exc_info:
        parser.parse(text)

    assert "Failed to parse MyModel" in str(exc_info.value)


# Test get_format_instructions method
def test_get_format_instructions():
```

```
parser = JsonOutputParser(MyModel)

instructions = parser.get_format_instructions()

assert "JSON Formatting Instructions:" in instructions

assert '"name": {"type": "string"}' in instructions

assert '"age": {"type": "integer"}' in instructions
```