Костюченко Марк Михайлович , взял тему “Реализуйте класс, который описывает IP.

Цель проекта

Реализовать класс , который описывает IP , хранение IP адреса , действителен ли этот IP класс , возможность предоставить IP адрес

Общее описание

Разработать класс IP для использования в проекте на Unity, который будет представлять собой объект IP-адреса. Класс должен поддерживать базовые операции и проверки, связанные с IP-адресами, и быть интегрированным с компонентами Unity.

Цели и задачи

1. Хранение IP-адреса в формате IPv4.

2. Проведение валидации введенного IP-адреса.

3. Предоставление методов для получения информации и работы с IP-адресом.

Требования

1. Атрибуты класса:

   -dress  строка, представляющая IP-адрес в формате "x.x.x.x", где x - число от 0 до 255.

2. Методы класса:

   - IP - конструктор, который принимает строку с IP-адресом и инициализирует объект. Должна быть реализована валидация адреса.

   - bool - метод, возвращающий true, если IP-адрес валиден, и false в противном случае.

   - List<int> - метод, возвращающий список из четырех октетов IP-адреса.

   - override string ToString - метод, возвращающий строковое представление IP-адреса.

3. Валидация IP-адреса:

   - Проверять, что адрес состоит из четырех октетов.

   - Проверять, что каждый октет является целым числом от 0 до 255.

   - Если адрес не валиден, выбрасывать исключение

4. Интеграция с Unity:

   - Класс должен быть написан в виде MonoBehaviour (если это необходимо для использования в Unity).

   - Для демонстрации функциональности, следует создать простой UI (например, InputField для ввода IP-адреса и Button для проверки) в Unity.

5. Тестирование класса:

   - Провести тестирование функциональности с различными входными данными.

   - Написать тесты, если используется система юнит-тестирования, такая как NUnit.

Технические детали

- Язык программирования: C#.

- Версия Unity: 6000.0.36f1 и выше.

- Класс должен быть совместим с Unity.

Сроки выполнения

- Разработка класса: 1 неделя.

- Интеграция с UI: 3 дня.

- Итоговое тестирование: 4 дня.

Дополнительные требования

- Код должен быть хорошо документирован и сопровождаться комментариями.

- Соблюдать принципы хорошего программирования и оформления кода.