

Аналіз роботи напарника (Красноручької Марії)

1. Недоліки проєктування.

- Використання числівника у назві класу («**FirstRoad**») – клієнт, який буде працювати з цим типом, може неправильно зрозуміти його суть;
- Модифікатори доступу типів: майже всі класи можна буде використати лише в межах даної збірки;
- Порушено принцип відкритості\ закритості. При додаванні ще одного світлофору доведеться модифікувати метод *AddTrafficLight()* класу **FirstRoad**;
- В інших аспектах програма годиться для подальшого розширення.

2. Визначити відповідність реалізації програми.

- З умови завдання до програми слід було додати такі сутності: смуга, перехрестя та симулятор (це виконує клас **Contol**);

```
Lights colors and dirs:
northSouth    southNorth    eastWest    westEast
Green+Left    Green+Left    Red+Right   Yellow+Right Yellow+Right
```

- повинен показуватися час;
- Користувач не має змоги міняти час перемикавання на інше світло.

3. Зазначити недоліки реалізації.

- Інтерфейс «IRoad»: для цього краще виділити абстрактний клас;
- Цей інтерфейс має відповідати за свою частину функціональності програми. Порушено принцип єдиного обов'язку - методи *Start()* та *Stop()* не повинні бути у ньому;
- Класи **OneDirTrafficLight** та **TwoDirTrafficLight**. Із назви зрозуміле скорочення, але краще писати повні назви класів та їх членів. А загалом хороший підхід до реалізації цих класів. У програму можна

буде додати ще один світлофор методом успадкування класу **TrafficLight** (який мав би бути абстрактним);

- У методі ***ToString()*** використано конкатенацію у циклі. Це надмірне споживання ресурсів. Для вирішення проблеми треба було використати клас **StringBuilder**;
- `public void RaiseChangeSideCo() - ?;`
- Метод ***Change(object sender, ElapsedEventArgs e)*** класу **TrafficLight** притримується конвенції *Microsoft* щодо подій. *Гарно*.

4. Перевірити валідацію даних.

- Параметри методів класу валідатора – **object** (поверхнева типізація). Це робить програму гнучкою, але в цьому випадку краще напряму визначити ті типи параметрів, які будуть підтримуватися методами валідатора;
- Щодо реалізації цих методів: **.NET** передбачив типи значень та типи посилання у nullable контексті. Можна було цим скористатися без створення цих методів.
- У тілі методу ***isCorrectInterval(object input)*** виконується ***boxing***, цього можна було уникнути.

5. Відмітити складність перебудови попередньої програми.

Порівняно із попередньою версією продукту, зберігся клас **TrafficLight**, але його вміст доповнено. Це свідчить про те, що при нововведеннях, попередня версія не зовсім підходила для розширення функціоналу програми. Щодо складності перебудови: попередня версія не містила велику кількість класів, тому складність перебудови, як на мене, невисока.