

# Projektowanie obiektowe 2024L

## Etap 5 – Aktualizacja danych

### Wstęp

W tym etapie do programu trzeba będzie dodać mechanizm aktualizujący stan wczytanych wcześniej obiektów.

### Źródło danych

Źródło danych symulujące źródło sieciowe, wykorzystywane w poprzednich etapach, zostało wzbogacone o mechanizm zgłaszający zmiany w danych obiektów.

Interfejs źródła zawiera teraz 3 dodatkowe zdarzenia:

- 1) OnIDUpdate – aktualizujące ID obiektu,
- 2) OnPositionUpdate – aktualizujące dane dotyczące pozycji obiektu,
- 3) OnContactInfoUpdate – aktualizujące dane kontaktowe zapisane w obiekcie.

Zdarzenia te mogą być zgłaszane w dowolnym momencie w trakcie działania źródła danych. Należy stworzyć system umożliwiający łatwą obsługę nowych rodzajów zdarzeń aktualizujących obiekty.

### OnIDUpdate

Zdarzenie zdefiniowane jest w następujący sposób:

```
public event IDUpdate? OnIDUpdate;
public delegate void IDUpdate(object sender, IDUpdateArgs args);
public class IDUpdateArgs
{
    public UInt64 ObjectID { get; init; }
    public UInt64 NewObjectID { get; init; }
}
```

ObjectID – obecne ID obiektu

NewObjectID – nowe ID obiektu

### OnPositionUpdate

Zdarzenie zdefiniowane jest w następujący sposób:

```
public event PositionUpdate? OnPositionUpdate;
public class PositionUpdateArgs
{
    public UInt64 ObjectID { get; init; }
    public Single Longitude { get; init; }
    public Single Latitude { get; init; }
}
```

```
public Single AMSL { get; init; }  
}
```

ObjectID – obecne ID obiektu,

Longitude – nowa wysokość geograficzna,

Latitude – nowa szerokość geograficzna,

AMSL – nowa wysokość nad poziomem morza

Po aktualizacji pozycji dane na mapie powinny zostać zaktualizowane, jeśli modyfikacji uległ lot lub samolot, należy przesunąć jego obecną pozycję na mapie do nowej wartości, a następnie interpolować do miejsca docelowego. (Podpowiedź: można użyć tu dekoratora)

## OnContactInfoUpdate

```
public event ContactInfoUpdate? OnContactInfoUpdate;  
public delegate void ContactInfoUpdate(object sender, ContactInfoUpdateArgs args);  
public class ContactInfoUpdateArgs  
{  
    public UInt64 ObjectID { get; init; }  
    public string PhoneNumber { get; init; }  
    public string EmailAddress { get; init; }  
}
```

ObjectID – obecne ID obiektu,

PhoneNumber – nowy numer telefonu,

EmailAddress – nowy email.

## Integracja z aplikacją

Zaktualizowane źródło danych ma współdziałać razem z danymi wczytywanymi z pliku .ftr, aplikacja powinna najpierw wczytać wszystkie dane z pliku .ftr, a następnie uruchomić symulator źródła sieciowego z podanym przykładowym plikiem w formacie .ftre i obsługiwać zgłaszane zdarzenia (także te z nowymi danymi). Wszystkie zmiany dokonywane na istniejących już danych powinny być logowane do pliku tekstowego. W logach powinna znajdować się informacja o stanie sprzed modyfikacji, jak i po. Logi z każdego dnia powinny być logowane do oddzielnych plików. Powinna istnieć możliwość odróżnienia różnych uruchomień aplikacji tego samego dnia.

Uwaga!

Źródło danych przyjmuje teraz plik w formacie .ftre.

Po wykonaniu modyfikacji dane dalej powinny być poprawne.

Jeśli modyfikacje byłyby niemożliwe do wykonania lub niepoprawne należy zapisać w logach odpowiednią informację.

# Termin wykonania

2 tygodnie

Wszystkie pliki źródłowe muszą zostać wgrane na repozytorium git do 23.04.2024 23:59.

Projekt należy zaprezentować prowadzącemu w czasie zajęć 24.04.2024.