

## 1 Warstwa logiczna aplikacji

Klasa **Baloon** jest reprezentacją pojedynczego balonu, opisanego polami:

- **long number** - specyfikujący kolejny i unikalny numer balonu.
- **BaloonColor color** - opisujący jeden z 16 różnych kolorów (których nie ma w klasie **Color**). Kolor balonu będzie definiowany przez typ wylicznikowy (**enum**) dostarczany klasie jako parametr konstruktora.

Przygotuj klasę **Factory**, która będzie asynchronicznie i nieustannie produkować balony (realizowane przez obiekt klasy **Baloon**) z częstotliwością pomiędzy 100[ms] a 3[s] sekund.

Program będzie mógł utworzyć bliżej nieokreśloną ilość obiektów klasy **Factory** z których każdy, niezależnie będzie produkował balony. Wyprodukowane balony zostaną umieszczone w składzie realizowanym przez klasę **Storage** na podstawie jednej z istniejących kolekcji (dobierz kolekcję adekwatnie do realizowanego problemu). Skład będzie przyjmował dokładnie 99 balonów, a jeżeli okaże się że skład jest pełny to fabryki niemożące składować swojego towaru nie mogą produkować kolejnych balonów.

Przygotuj również klasę **Transporter**, która będzie asynchronicznie transportować balony pobierane ze składu w kolejności od najdłużej składowanego balonu. Transportowany towar będzie pobierany w partiach po 10 sztuk i wysyłany do odbiorców nie częściej niż jeden transport co 10[s]. Załadunek towaru trwa do czasu uzyskania całej partii i nie może zostać zakłócony przez żaden inny transporter. Dostawa towaru trwa tyle samo czasu ile trwał załadunek towaru.

## 2 Warstwa wizualna

Wykorzystaj komponenty **SWING** do stworzenia okienkowej aplikacji, która:

- w pozycji początku linii, będzie miała przewijalny panel zawierający wszystkie utworzone fabryki oraz przycisk pozwalający na dodanie nowej instancji klasy **Factory**;
- w pozycji końca linii, będzie miała przewijalny panel zawierający wszystkie utworzone transportery oraz przycisk pozwalający na dodanie nowej instancji klasy **Transporter**;
- w środkowej pozycji panel przedstawiający stan magazynu realizowanego przez klasę **Storage**;

Wizualizacja pojedynczego panelu reprezentującego fabrykę powinna zawierać:

- licznik balonów wyprodukowanych przez tą fabrykę,
- suwak pozwalający na zmianę czasu produkcji pojedynczego balonu
- przycisk usunięcia danej fabryki

Wizualizacja pojedynczego panelu reprezentującego transporter powinna zawierać:

- informację o statusie transportu (*oczekiwanie* → *zaadunek* → *dostawa* → ...)
- przyciski sterujące pozwalające na zatrzymanie lub uruchomienie komponentu

Gdy z magazynie znajduje się przynajmniej 90 balonów, następuje awaryjne wypuszczenie wszystkich balonów, któremu towarzyszy animacja wzlatujących ku niebu balonów w aplikacji.

Rozwiązanie w postaci dostarczonej aplikacji należy przedstawić do dnia 29.V.2021 na platformie Teams.