Wytłumaczenie UI:

* Wyświetlana mapa składa się z kwadratów trzech kolorów:
  + Żółty – roślina
  + Zielony – trawa
  + Czerwony – zwierzę
  + Niebieski - dziecko
* Dodatkowo, kolor zwierzęcia zależy od ilości energii, jaką ma – im mniej, tym ciemniejszy kolor zwierzaka: od czerwonego do czarnego

Statystyki symulacji dzień po dniu zapisywane są w pliku simulation\_data.csv.

1. **MapDirection**
   * Enum reprezentujący możliwe kierunki na mapie (NORTH, SOUTH, EAST, WEST), z metodami do konwersji na wektory jednostkowe oraz obliczania nowych kierunków po obróceniu.
2. **MoveValidator**
   * Interfejs służący do walidacji, sprawdzający, czy obiekt może poruszać się na wskazaną pozycję na mapie.
3. **Plant**
   * Klasa reprezentująca roślinę na mapie, posiadająca pozycję i metodę toString(), zwracającą znak rośliny.
4. **Simulation**
   * Klasa zarządzająca symulacją, obejmująca logikę poruszania zwierząt, reprodukcji, konsumpcji energii, wzrostu roślin i zbierania danych do pliku CSV.
5. **SimulationApp**
   * Abstrakcyjna klasa, bazująca na Application, służąca jako punkt wyjścia dla aplikacji symulacji.
6. **SimulationParameters**
   * Klasa zawierająca parametry konfiguracyjne symulacji, takie jak wymiary mapy, energia roślin, liczba zwierząt, itp.
7. **TmpValidatorImplementation**
   * Implementacja interfejsu MoveValidator, która sprawdza, czy pozycja leży w określonym zakresie.
8. **Vector2d**
   * Klasa reprezentująca dwuwymiarowy wektor z współrzędnymi x i y oraz metodami do operacji arytmetycznych, porównań i konwersji.
9. **World**
   * Klasa uruchamiająca aplikację symulacyjną, odpowiedzialna za uruchomienie UI w aplikacji.
10. **WorldElement**

* Interfejs reprezentujący elementy świata, posiadające pozycję na mapie.

1. **WorldMap**

* Interfejs definiujący operacje na mapie, takie jak umieszczanie zwierząt, ruch, rozmnażanie, konsumpcję roślin i zarządzanie przestrzenią.

1. **Animal**

* Klasa reprezentująca zwierzę, zawierająca logikę poruszania się, jedzenia, rozmnażania, tracenia energii oraz przechowywania genomu.

1. **CSVWriter**

* Klasa pomocnicza do zapisywania danych do pliku CSV, z funkcjami otwierania, pisania linii i zamykania pliku.

1. **GenTypeStartGeneration**

* Klasa generująca początkowy typ genomu dla zwierząt, zależny od liczby genów.

1. **Boundary**

* Klasa reprezentująca granice mapy, przechowująca minimalne i maksymalne współrzędne w dwóch wymiarach.

1. **SimulationPresenter**

* Klasa odpowiedzialna za interfejs graficzny symulacji, rysująca mapę i dane związane z przebiegiem symulacji.

Autorzy:

* Maciej Sitny
* Bartłomiej Prus