

PROGRAMOWANIE OBIEKTOWE

PROJEKT WTOREK 17:05

Virus

Autor:

Uladzislau TRYBUKHOUSKI (lider)

Yaroslav PEREPILKA

Anton HOMAN

Prowadzący:

Mgr. inż Tobiasz PUŚLECKI

Spis treści

1	Opis i cele symulacji.....	3
1.1	Wstęp.....	3
1.2	Opis programu symulacji.....	3
2	Wybór technologii.....	4
2.1	FXML.....	4
2.2	CSS.....	4
3	Opis klas oraz ich działanie i przeznaczenie.....	5
3.1	Class Population.....	5
3.2	Class Virus.....	6
3.2.1	Class RespiratoryVirus.....	6
3.2.2	Class FoodborneVirus.....	6
3.2.3	Class ContactVirus.....	7
3.3	Class Map.....	7
3.4	Class MainController.....	7
3.5	Class StartApplication.....	9
3.6	Class StartController.....	9
3.7	Class VirusChoiceController.....	9
3.8	Class VirusSetupController.....	9
3.9	Class setupContactVirusController.....	10
3.10	Class setupFoodborneVirusController.....	11
3.11	Class setupRespiratoryVirusController.....	11
3.12	Class Country.....	12
4	Diagramy klas i obiektów.....	14
4.1	Diagram obiektów.....	14
4.2	Diagram klas.....	15

Opis i cele symulacji

Wstęp

Celem projektu jest symulująca rozprzestrzenianie się i rozwój wirusa w środowisku wirtualnym. Użytkownicy mogą dostosować różne parametry wirusa i środowiska, aby zbadać jego wpływ na populację i skuteczność różnych strategii kontroli.

Opis programu symulacji

1) Wybór wirusa

Na początku symulacji użytkownik ma możliwość wyboru jednego z trzech typów wirusa:

- Oddechowy
- Kontaktowy
- Pokarmowy

Każdy typ wirusa ma swoje cechy, w tym drogi przenoszenia. Użytkownik może dostosować parametry wirusa, takie jak okres inkubacji, odporność na temperaturę i listę objawów (kaszel, bóle głowy, pocenie się, itp.). Te objawy wpływają na kluczowe cechy wirusa:

- Zakaźność
- Śmiertelność
- Odporność na środowisko
- Odporność na leki

2) Konfiguracja osady

Po skonfigurowaniu wirusa użytkownik przechodzi do konfiguracji osady. Osady w symulacji są przedstawione jako heksagony, z których każdy może pomieścić do 1000 osób. Każda osada ma następujące parametry:

- Poziom medycyny
- Odporność każdego mieszkańca na wirusa

Osady mogą sąsiadować ze sobą, jednak czasami są oddzielone przeszkodami oznaczonymi szarymi heksagonami (niezamieszkałymi). Wszystkie osady są również podzielone na grupy "termiczne":

- Niebieskie – najzimniejsze
- Pomarańczowe – najcieplejsze

3) Etap symulacji

rozprzestrzenianie się wirusa zależy od następujących czynników:

- Zakaźność wirusa
- Czynniki środowiskowe
- Poziom medycyny w osadzie
- Poziom odporności osady na wirusa
- Śmiertelność wirusa

Gdy wirus dostanie się do człowieka, rozpoczyna się okres inkubacji. W tym czasie zarażony może zakażać innych zdrowych ludzi. Po zakończeniu okresu inkubacji człowiek ma dwa możliwe scenariusze:

1. Umiera.
2. Pozostaje zakażonym
Ludzkość stara się przetrwać, dla tego zamykają granice państw.
3. Zakończenie symulacji
Symulacja trwa do momentu, gdy cała ludzkość albo wymrze, albo wyzdrowieje.

Wybór technologii

FXML

W naszym projekcie do tworzenia interfejsu użytkownika wykorzystujemy technologię FXML. FXML to język znaczników w formacie XML, opracowany do opisu interfejsów graficznych na platformie JavaFX.

Jak wykorzystujemy FXML w naszym projekcie:

- **Definiowanie interfejsu:** Wszystkie główne elementy interfejsu (np. przyciski, pola tekstowe, menu) są opisane w plikach FXML. To pozwala na efektywniejszą współpracę między projektantem interfejsu a programistą.

- **Integracja z Java:** Pliki FXML są powiązane z odpowiednimi kontrolerami napisanymi w języku Java, które obsługują zdarzenia i współpracują z logiką aplikacji. Jest to realizowane za pomocą adnotacji i mechanizmów wiązania danych JavaFX.

CSS

Nasz projekt wykorzystuje CSS do stylizacji interfejsu użytkownika, co zapewnia elastyczność, modułowość i spójność designu.

Jak wykorzystujemy CSS w naszym projekcie:

- Za pomocą CSS zapewniamy jednolitość designu w całej aplikacji. Definiując style w jednym miejscu, możemy zastosować je do wielu elementów interfejsu.

- To upraszcza wprowadzanie zmian w designie: wystarczy zmienić styl w jednym miejscu, aby zmiany odzwierciedliły się w całej aplikacji.

- CSS pozwala oddzielić wizualne formatowanie od logiki i struktury interfejsu. To upraszcza utrzymanie kodu i sprawia, że jest on bardziej czytelny.

- Projektanci mogą pracować nad stylizacją interfejsu, nie ingerując w główny kod aplikacji.

Opis klas oraz ich działanie i przeznaczenie

Class Population

1) Opis classa

Class Population to klasa ludzkości w jednym kraju.

2) Zmienne

“worldPopulation”: Całkowita liczba osób na świecie.

“worldInfected”: Całkowita liczba zarażonych osób na świecie.

“worldCorpse”: Całkowita liczba martwych ludzi na świecie.

“country”: nazwa kraju.

“populationDensity”: Gęstość zaludnienia kraju, obliczana na podstawie całkowitej liczby ludności i powierzchni kraju.

“population”: Całkowita liczba ludności w kraju.

“infected”: Liczba zarażonych osób w kraju.

“stepSick”: Liczba nowych infekcji na krok symulacji.

“stepCorpse”: Liczba nowych zgonów na krok symulacji.

“corpse”: Całkowita liczba zgonów w kraju.

“stability”: Poziom stabilności w kraju wpływający na rozprzestrzenianie się wirusa.

“averageTemperature”: Średnia temperatura w danym kraju może wpływać na przeżywalność wirusa.

“borders”: Parametr wskazujący zamkniętość granic kraju.

“medicalLevel”: Poziom opieki medycznej w kraju wpływający na kontrolę wirusa

“COUNTRY_AREA”: Stała reprezentująca obszar kraju.

“random”: do generowania liczb losowych.

3) Metody klasowe

Population(String country, double populationDensity, int population, double stability, double averageTemperature, boolean borders, double medicalLevel): Podstawowy konstruktor, który inicjalizuje wszystkie zmienne instancji i aktualizuje statystyki świata.

Population(String country): Alternatywny konstruktor, który inicjalizuje tylko nazwę kraju.

simulateInfectionStep(Virus virus): Metoda obliczania jednoetapowej symulacji infekcji. Aktualizuje populację, gęstość zaludnienia, poziom medycyny i prawdopodobieństwo transmisji.

medicalDevelopment(): Metoda modelowania postępów lub ulepszeń medycznych.

getInfections(): Zwraca liczbę infekcji.

setInfected(int infected): Ustawia liczbę infekcji i aktualizuje krok nowych infekcji.

getStepSick(): Zwraca liczbę nowych infekcji w jednym kroku.

setAverageTemperature(double averageTemperature): Ustawia średnią temperaturę w kraju.

getPopulation(): Zwraca aktualną liczbę ludności w kraju.

getInfected(): Zwraca aktualną liczbę zakażonych w kraju.

getCorpse(): Zwraca bieżącą liczbę zgonów w kraju.

setBorders(boolean borders): Ustawia stan granic kraju (otwarte lub zamknięte).

getWorldPopulation(): Zwraca aktualną liczbę ludności na świecie.

getWorldInfected(): Zwraca aktualną liczbę zainfekowanych osób na świecie.

getWorldCorpse(): Zwraca aktualną liczbę martwych osób na świecie.

getCountryName(): Zwraca nazwę kraju.

redactWorldPopulation(boolean isCountry): Metoda edycji światowej populacji, zwiększa lub zmniejsza światową populację w zależności od stanu kraju.

Class Virus

1)Opis classa

Class Virus to matczyzna klasa wirusów dla wirusów :RespiratoryVirus , FoodborneVirus , ContactVirus .

2)Zmienne

“type”: Rodzaj wirusa.

“incubationPeriod”:Okres inkubacji wirusa, czas między zakażeniem a pojawieniem się objawów.

“infectionProbability”: Prawdopodobieństwo zarażenia się wirusem.

“mortalityRate”: Śmiertelność wirusowa.

“mutation”: zmienna typu boolean, wskazująca, czy nastąpiła mutacja wirusa.

“mutationSpeed”: zmienna całkowita, reprezentująca prędkość mutacji wirusa.

3) Metody klasowe

Virus(String type, double incubationPeriod, double infectionProbability, double mortalityRate):): Konstruktor inicjujący wirusa z danym okresem inkubacji, prawdopodobieństwem infekcji i współczynnikiem śmiertelności.

setCharacteristics(double incubationPeriod, double infectionProbability, double mortalityRate): Metoda dostrajania właściwości wirusa, takich jak okres inkubacji, prawdopodobieństwo infekcji i wskaźnik śmiertelności.

getTransmissionRoute(): Abstrakcyjna metoda uzyskiwania ścieżki transmisji wirusa do zaimplementowania w klasach podrzędnych.

mutation(): Abstrakcyjna metoda modelowania mutacji wirusa do zaimplementowania w klasach potomnych.

getInfectionProbability(): Abstrakcyjna metoda uzyskiwania prawdopodobieństwa infekcji, która musi zostać zaimplementowana w klasach potomnych.

Class RespiratoryVirus

1)Opis classa

Class RespiratoryVirus to klasa wirusów typu oddechowego.

2) Metody klasowe

RespiratoryVirus(double incubationPeriod, double infectionProbability, double mortalityRate): Konstruktor inicjujący wirusa oddechowego z danym okresem inkubacji, prawdopodobieństwem infekcji i współczynnikiem śmiertelności.

getTransmissionRoute(): Zwraca ciąg znaków "Airborne" wskazujący, że wirus jest przenoszony drogą powietrzną.

mutation(): Metoda modelowania mutacji wirusa oddechowego.

getInfectionProbability(): Odwraca prawdopodobieństwo zarażenia się wirusem układu oddechowego.

Class FoodborneVirus

1)Opis classa

Class FoodborneVirus to wirusa klasa przenoszonych przez żywność.

2) Metody klasowe

FoodborneVirus(double incubationPeriod, double infectionProbability, double mortalityRate): Konstruktor, który wywołuje konstruktor nadrzędny klasy Virus i ustawia typ wirusa na „Foodborne” oraz przyjmuje wartości dla okresu inkubacji, prawdopodobieństwa infekcji i śmiertelności.

getTransmissionRoute(): wskazuje, że wirus jest przenoszony drogą pokarmową.

mutation(): Metoda modelowania mutacji wirusów przenoszonych przez żywność.

getInfectionProbability() Zmniejsza prawdopodobieństwo zarażenia się wirusem przenoszonym przez żywność.

Class ContactVirus

1) Opis classa

Class FoodborneVirus to klasa wirusa przenoszonych przez kontakt.

2) Metody klasowe

ContactVirus(double incubationPeriod, double infectionProbability, double mortalityRate): Konstruktor, który wywołuje konstruktor nadrzędny klasy Virus i ustawia typ wirusa na „Contact” oraz przyjmuje wartości dla okresu inkubacji, prawdopodobieństwa infekcji i śmiertelności.

getTransmissionRoute(): wskazuje, że wirus jest przenoszony przez kontakt.

mutation(): Metoda modelowania mutacji w wirusach przenoszonych drogą kontaktową.

getInfectionProbability(): Zmniejsza prawdopodobieństwo zarażenia się wirusem kontaktowym.

Class Map

1) Opis classa

Class Map to klasa wykonująca zadanie mapy świata .

2) Zmienne

“world”: Instancja klasy Country.

“mainController”: Instancja klasy MainController.

“countries”: dwuwymiarowa tablica list krajów.

“quantityCountries”: Liczba krajów na mapie.

“date”: Zmienna do śledzenia bieżącej daty lub czasu symulacji.

“Width”: Szerokość mapy.

“Height”: Wysokość mapy.

“Radius”: Promień sześciokąta używanego do reprezentowania krajów.

“numCols”: Liczba kolumn na mapie.

“numRows”: Liczba linii na mapie.

“pickedCountry”: Instancja klasy Country reprezentująca wybrany kraj.

3) Metody klasowe

Map(int Width, int Height, int Radius): Konstruktor, który inicjalizuje podstawowe parametry mapy i wywołuje metodę mapCreate

mapCreate(double hexRadius, double mapWidth, double mapHeight) Tworzy mapę z sześciokątnymi krajami.

newInfected(Country infectedCountry) Rozprzestrzenia infekcję na sąsiednie kraje.

setMouseEvent(String type, boolean isEvent) Ustawia lub usuwa obsługę zdarzeń myszy dla krajów.

setPickedCountry(Country country) Ustawia wybrany kraj i aktualizuje jego wybór.

getColor(int row) Zwraca kolor kraju w zależności od jego wiersza na mapie.

createHexagon(double x, double y, double radius) Tworzy sześciokąt o podanych współrzędnych i promieniu.

getCountries() Zwraca listę wszystkich krajów na mapie.

getWorld() Zwraca instancję klasy Country reprezentującą cały świat.

Class MainController

1) Opis classa

Class MainController to jest kontroler dla pliku Main FXML

2) Zmienne:

“PaneWidth”: Szerokość panelu, w którym wyświetlana jest mapa. Wartość domyślna: 750.

“paneHeight”: Wysokość panelu, w którym wyświetlana jest mapa. Wartość domyślna: 410.

“instance”: Zmienna statyczna zawierająca pojedynczą instancję klasy MainController.

“scheduler”: Harmonogram zadań do wykonywania metod w określonych odstępach czasu.

“circle_cnt”: Liczba okręgów używanych do wyświetlania zainfekowanych obszarów.

“map”: Obiekt klasy Map reprezentujący mapę z rozmieszczeniem krajów.

“virus”: Obiekt klasy RespiratoryVirus reprezentujący wirusa.

“Map”: Tablica obiektów Polygon reprezentujących mapę.

“stage”: Obiekt Stage reprezentujący bieżące okno aplikacji.

“scene”: Obiekt Scene reprezentujący bieżącą scenę aplikacji.

“mutationCnt”: Licznik mutacji wirusa.

"ProgressCnt": Licznik postępu symulacji
 "previousWorldInfected": Przechowują poprzednie wartości zainfekowanych i ludzi na świecie.
 "PreviousWorldCorpse": Przechowują poprzednie wartości zmarłych ludzi na świecie.
 "root": Obiekt Parent reprezentujący węzeł główny bieżącej sceny.

3) Komponenty FXML:

"world": Label do wyświetlania informacji o świecie.
 "startButton": Button uruchamiający symulację.
 "label": Label dla różnych tekstów.
 "myPane": Pane aby wyświetlić mapę i zainfekowane obszary.
 "field": Rectangle reprezentujący obszar mapy.
 "healthy": Label aby pokazać liczbę zdrowych osób.
 "sick": Label aby wyświetlić liczbę chorych osób.
 "dead": Label aby wyświetlić liczbę zmarłych osób.
 "healthyCountries": Label aby wyświetlić liczbę zdrowych krajów.
 "infectedCountries": Label aby wyświetlić liczbę zainfekowanych krajów.
 "communicationLabel": Label do wyświetlania komunikatów dla użytkownika.
 "healthyBar": ProgressBar aby pokazać postęp zdrowych ludzi.
 "infectedBar": ProgressBar do wyświetlania postępów zainfekowanych osób.
 "corpseBar": ProgressBar do wyświetlania postępów zmarłych osób.
 "incubation": Label aby pokazać okres inkubacji wirusa.
 "country": Label aby wyświetlić nazwę kraju.
 "pause": ToggleButton aby wstrzymać symulację.
 "start1": ToggleButton dla powolnego rozpoczęcia symulacji.
 "start2": ToggleButton aby szybko rozpocząć symulację.
 "symptoms": VBox do wystąpienia objawów wirusa.

3) Metody klasowe

MainController(): Konstruktor klasy, który ustawia bieżącą instancję klasy na statyczną zmienną instance.
 getInstance(): Zwraca bieżącą instancję klasy MainController.
 setEvents(boolean isRedact): Ustawia zdarzenia myszy dla mapy. Jeśli isRedact ma wartość true, umożliwia edycję mapy, w przeciwnym razie umożliwia wybór krajów.
 setStatistics(): Aktualizuje statystyki na ekranie, w tym liczbę zdrowych, chorych i martwych osób, a także zdrowych i zainfekowanych krajów.
 setPopulationStatistics(): Ustawia tekst statystyk populacji.
 checkProgress(): Sprawdza postęp symulacji i aktualizuje odpowiednie liczniki.
 home(): Powraca do ekranu głównego aplikacji, ładując nową scenę.
 setVirus(ContactVirus virus): Ustawia wirusa na typ ContactVirus.
 setFirstInfectedCountry: Ustawia pierwszy zainfekowany kraj i zwraca true, jeśli ustawienie było pomyślne, w przeciwnym razie false.
 newDay: Zwiększa licznik dni symulacji.
 startTimerOnce: Uruchamia timer symulacji z pojedynczą prędkością.
 startTimerTwice: Uruchamia timer symulacji z podwójną prędkością.
 stopTimerFunc: Zatrzymuje timer symulacji.
 stopTimer: Zatrzymuje timer, jeśli jest uruchomiony.
 generateRandomPointInPolygon(Polygon polygon): Generuje losowy punkt wewnątrz wielokąta.
 showInfected(Country country): Wyświetla zainfekowane osoby na mapie dla danego kraju.
 showCorpse(Country country): Wyświetla zmarłe osoby na mapie dla danego kraju.
 redactEnd(): Kończy fazę redakcji i przechodzi do etapu symulacji.
 initialize(): Inicjalizuje kontroler, ustawiając początkowe wartości i konfiguracje.
 setVirus(FoodborneVirus virus): Ustawia wirusa na typ FoodborneVirus.
 repeatSimulation(): Resetuje symulację i ponownie uruchamia wyświetlanie mapy.
 nextStep(): Wykonuje kolejny krok symulacji, aktualizując stany krajów i wyświetlając nowe zainfekowane obszary.
 setVirus(RespiratoryVirus virus): Ustawia bieżącego wirusa.
 ShowMap(): Tworzy i wyświetla mapę.
 setFirstInfectedCountry(): Ustawia pierwszy zainfekowany kraj i rozpoczyna symulację.
 setSymptoms(): Ustawia i wyświetla objawy wirusa.
 startTimer(int period): Uruchamia timer w celu wykonania następnego kroku symulacji w określonym odstępie czasu.

Class StartApplication

1)Opis classa

Class StartApplication to jest klasa dla rozpoczęcia symulacji

2) Metody klasowe

stage: Argument metody, obiekt stage reprezentujący główne okno aplikacji.

Parent root: Obiekt klasy Parent reprezentujący główny węzeł sceny. Tutaj jest on ładowany z pliku FXML.

Scene scene: Obiekt klasy Scene, który reprezentuje scenę w JavaFX. Jest on tworzony na podstawie węzła głównego.

FXMLLoader.load(getClass().getResource("start_window.fxml")): To wywołanie ładuje układ interfejsu użytkownika z pliku start_window.fxml.

Class StartController

1)Opis classa

Class StartController to jest kontroler dla pliku Start FXML

2)Zmienne

“stage”: Obiekt klasy stage reprezentujący bieżące okno (stage) aplikacji JavaFX.

“scene”: Obiekt klasy Scene reprezentujący scenę w bieżącym oknie.

“root”: Obiekt klasy Parent reprezentujący główny węzeł hierarchii elementów graficznych załadowanej sceny.

3) Metody klasowe

next(ActionEvent event): Ta metoda jest wywoływana, gdy powiązane zdarzenie (np. naciśnięcie przycisku) jest aktywowane i służy do przejścia do następnego ekranu/sceny. Metoda ładuje nową scenę z pliku FXML i ustawia ją w bieżącym oknie.

Class VirusChoiseController

1)Opis classa

Class VirusChoiseController to jest kontroler dla pliku VirusChoise FXML.

2)Zmienne

“stage”: Obiekt klasy Stage reprezentujący bieżące okno (stage) aplikacji JavaFX.

“scene”: Obiekt klasy Scene, który reprezentuje scenę w bieżącym oknie.

“root”: Obiekt klasy Parent reprezentujący główny węzeł hierarchii elementów graficznych załadowanej sceny.

3) Metody klasowe

RespiratoryVirus(ActionEvent event): Metoda jest wywoływana po wybraniu wirusa przenoszonego drogą powietrzną. Wczytuje i ustawia scenę w celu dostosowania RespiratoryVirus.

ContactVirus(ActionEvent event): Metoda jest wywoływana po wybraniu wirusa kontaktowego. Wczytuje i ustawia scenę dla konfiguracji ContactVirus.

FoodborneVirus(ActionEvent event): Metoda jest wywoływana po wybraniu wirusa przenoszonego przez żywność. Wczytuje i ustawia scenę dla ustawienia FoodborneVirus.

Class VirusSetupController

1)Opis classa

Class VirusSetupController to jest kontroler dla pliku VirusSetup FXML

2) Komponenty FXML:

“resources”: Obiekt ResourceBundle używany do lokalizacji interfejsu.

“location”: Obiekt URL wskazujący lokalizację pliku FXML.

“BronchitisCheckBox”: CheckBox aby wybrać objaw „Zapalenie oskrzeli”.

“ConvulsionsCheckBox”: CheckBox aby wybrać objaw „drgawki”.

“CoughCheckBox”: CheckBox aby wybrać objaw „Kaszel”.

“FeverCheckBox”: CheckBox aby wybrać objaw „Gorączka”.

“HallucinationsCheckBox”: CheckBox aby wybrać objaw „Halucynacje”.

“HeadacheCheckBox”: CheckBox aby wybrać objaw „Ból głowy”.

"PharyngitisCheckBox": CheckBox aby wybrać objaw „Zapalenie gardła”.
 "VomitCheckBox": CheckBox aby wybrać objaw „Wymioty”.
 "DurationOfFeverSlider": Slider w celu ustalenia czasu trwania gorączki.
 "IncubationPeriodSlider": Slider aby ustawić okres inkubacji.
 "TemperatureSlider": Slider do ustawiania temperatury podczas gorączki.
 "StartButton": Przycisk uruchamiający symulację z bieżącymi ustawieniami wirusa.

3) Metody klasowe

initialize(): Metoda inicjalizacji kontrolera, w której ustawiane są początkowe akcje dla elementów interfejsu.

Class setupContactVirusController

1) Opis classa

Class setupContactVirusController to jest kontroler dla pliku setupContactVirus FXML.

2) Zmienne:

"selected Symptoms": Tablica przechowująca wybrane symptomy przez użytkownika.
 "abdominalPainCheckBox": Jeśli użytkownik zaznaczy pole wyboru, to w metodzie getSelectedSymptoms() zostanie dodany symptom "Abdominal Pain" do listy wybranych symptomów.
 "bodilyFluidsCheckBox": To pole wyboru pozwala użytkownikowi wskazać, czy wirus przenosi się poprzez płyny ustrojowe. Jeśli użytkownik zaznaczy to pole, zwiększa to poziom infekcyjności wirusa.
 "cotaminatedObjectsCheckBox": To pole wyboru umożliwia użytkownikowi wskazanie, czy wirus przenosi się poprzez skażone przedmioty. Jeśli użytkownik zaznaczy to pole, zwiększa to poziom infekcyjności wirusa.
 "resistansBar": Pasek postępu, który pokazuje poziom odporności wirusa na różne czynniki.
 "virus": konstruktor.

3) Komponenty FXML:

"abdominalPainCheckBox": CheckBox aby wybrać objaw „Ból brzucha”.
 "antiviralDrugsResistanceCheckBox": CheckBox w celu selekcji pod kątem oporności na leki przeciwwirusowe.
 "bodilyFluidsCheckBox": CheckBox aby wybrać przenoszenie przez płyny ustrojowe.
 "comaCheckBox": CheckBox aby wybrać objaw „Śpiączka”.
 "contaminatedObjectsCheckBox": CheckBox aby wybrać transmisję przez zanieczyszczone obiekty.
 "disinfectantsResistanceCheckBox": CheckBox do wyboru odporności na środki dezynfekujące.
 "feverCheckBox": CheckBox aby wybrać objaw „Gorączka”.
 "nauseaCheckBox": CheckBox aby wybrać objaw „Nudności”.
 "pHchangesResistanceCheckBox": CheckBox dla wyboru odporności na zmiany pH.
 "paralysisCheckBox": CheckBox aby wybrać objaw „Paraliż”.
 "physicalContactCheckBox": CheckBox aby wybrać transmisję poprzez kontakt fizyczny.
 "rushCheckBox": CheckBox aby wybrać objaw „Wysypka”.
 "sepsisCheckBox": CheckBox aby wybrać objaw „Sepsa”.
 "vomitingCheckBox": CheckBox aby wybrać objaw „Wymioty”.
 "incubationSlider": Slider w celu ustalenia okresu inkubacji wirusa.
 "infectivityBar": ProgressBar aby pokazać poziom zakaźności wirusa.
 "mortalityBar": ProgressBar aby wyświetlić wskaźnik śmiertelności wirusa.
 "resistanceBar": ProgressBar aby wyświetlić poziom odporności wirusa.
 "nextButton": Przycisk umożliwiający przejście do następnego ekranu.

3) Metody klasowe

StartSimulation(): Metoda przeprowadzania symulacji.
 initialize(): Metoda inicjalizująca kontroler. Jest automatycznie wywoływana po załadowaniu pliku FXML.
 W tej wersji jest pusta, ale może być używana do inicjalizacji wartości początkowych.
 getVirus(): Zwraca skonfigurowany wirus typu ContactVirus.
 UpdateSimulation(ActionEvent event): Metoda aktualizacji parametrów symulacji.
 GetSelectedSymptoms() Metoda uzyskiwania wybranych objawów.
 CalculateInfectivity(): Metoda obliczania zakaźności.
 CalculateResistance(): Oblicza odporność wirusa na podstawie wybranych odporności. Uwzględnia zaznaczone pola wyboru dotyczące odporności na leki przeciwwirusowe, środki dezynfekcyjne i zmiany pH.
 CalculateMortality(): Metoda obliczania śmiertelności.

Class `setupFoodborneVirusController`

1) Opis classa

Class `setupFoodborneVirusController` to jest kontroler dla pliku `setupFoodborneVirus FXML`.

2) Zmienne:

“selected Symptoms”: Tablica przechowująca wybrane symptomy przez użytkownika.

“virus”: konstruktor.

2) Komponenty FXML:

“selectedSymptoms”: Tablica ciągów znaków do przechowywania wybranych objawów.

“abdominalPainCheckBox”: `CheckBox` aby wybrać objaw „Ból brzucha”.

“antisepticsResistanceCheckBox1”: `CheckBox` do wyboru odporności na środki antyseptyczne.

“contaminatedWaterCheckBox”: `CheckBox` aby wybrać transmisję przez zanieczyszczoną wodę.

“dehydrationCheckBox”: `CheckBox` aby wybrać objaw „Odwodnienie”.

“feverCheckBox”: `CheckBox` aby wybrać objaw „Gorączka”.

“freezingResistanceCheckBox”: `CheckBox` aby wybrać odporność na zamarzanie.

“diarheaCheckBox”: `CheckBox` aby wybrać objaw „Biegunka”.

“improperFoodHandlingCheckBox”: `CheckBox` do selekcji w celu przeniesienia poprzez niewłaściwe obchodzenie się z żywnością.

“nauseaCheckBox”: `CheckBox` aby wybrać objaw „Nudności”.

“pasteurizationResistanceCheckBox”: `CheckBox` do wyboru odporności na pasteryzację.

“salivacheckBox”: `CheckBox` aby wybrać transmisję śliny.

“sepsisCheckBox”: `CheckBox` aby wybrać objaw „Sepsa”.

“stomachAcidResistanceCheckBox”: `CheckBox` aby wybrać odporność na kwas żołądkowy.

“toxicHepatitisCheckBox”: `CheckBox` aby wybrać objaw „Toksyczne zapalenie wątroby”.

“vomitingCheckBox”: `CheckBox` aby wybrać objaw „Wymioty”.

“incubationSlider”: `Slider` ustalenie okresu inkubacji wirusa.

“infectivityBar”: `ProgressBar` aby pokazać poziom zakaźności wirusa.

“mortalityBar”: `ProgressBar` aby wyświetlić wskaźnik śmiertelności wirusa.

“resistanceBar”: `ProgressBar` aby wyświetlić poziom odporności wirusa.

“nextButton”: Przycisk umożliwiający przejście do następnego ekranu.

3) Metody klasowe

`StartSimulation()`: Metoda przeprowadzania symulacji.

`UpdateSimulation(ActionEvent event)`: Metoda aktualizacji parametrów symulacji.

`GetSelectedSymptoms()` Metoda uzyskiwania wybranych objawów.

`CalculateInfectivity()`: Metoda obliczania zakaźności.

`CalculateMortality()`: Metoda obliczania śmiertelności.

Class `setupRespiratoryVirusController`

1) Opis classa

Class `setupRespiratoryVirusController` to jest kontroler dla pliku `setupRespiratoryVirus FXML`.

2) Zmienne:

“selected Symptoms”: Tablica przechowująca wybrane symptomy przez użytkownika.

“virus”: konstruktor.

2) Komponenty FXML:

“selectedSymptoms”: Tablica ciągów znaków do przechowywania wybranych objawów.

“feverCheckBox”: `CheckBox` aby wybrać objaw „Gorączka”.

“coughCheckBox”: `CheckBox` aby wybrać objaw „Kaszel”.

“fatigueCheckBox”: `CheckBox` aby wybrać objaw „Zmęczenie”.

“HeadacheCheckBox”: `CheckBox` aby wybrać objaw „Ból głowy”.

“soreThroatCheckBox”: `CheckBox` aby wybrać objaw „Ból gardła”.

“shortnessOfBreathCheckBox”: `CheckBox` aby wybrać objaw „Duszność”.

“pulmonaryAcidosisCheckBox”: `CheckBox` aby wybrać objaw „kwasica płucna”.

“cerebralEdemaCheckBox”: `CheckBox` aby wybrać objaw „Obrzęk mózgu”.

“airborneCheckBox”: `CheckBox` aby wybrać powietrzną drogę transmisji.

“contactCheckBox”: `CheckBox` aby wybrać kontaktową ścieżkę transmisji.

“dropletsCheckBox”: CheckBox aby wybrać transmisję przez krople.
 “antibioticResistanceCheckBox”: CheckBox w celu selekcji odporności na antybiotyki.
 “heatResistanceCheckBox”: CheckBox aby wybrać odporność na wysoką temperaturę.
 “coldResistanceCheckBox”: CheckBox aby wybrać odporność na niskie temperatury.
 “incubationSlider”: Slider ustalenie okresu inkubacji wirusa.
 “infectivityBar”: ProgressBar aby pokazać poziom zakaźności wirusa.
 “mortalityBar”: ProgressBar aby wyświetlić wskaźnik śmiertelności wirusa.
 “resistanceBar”: ProgressBar aby wyświetlić poziom odporności wirusa.
 “nextButton”: Przycisk umożliwiający przejście do następnego ekranu.

3) Metody klasowe

initialize(): Metoda inicjalizacji kontrolera, w której ustawiane są początkowe akcje dla elementów interfejsu.
 getVirus(): Zwraca skonfigurowany wirus typu RespiratoryVirus.
 StartSimulation(): Metoda przeprowadzania symulacji.
 UpdateSimulation(ActionEvent event): Metoda aktualizacji parametrów symulacji.
 CalculateResistance(): Oblicza odporność wirusa na podstawie wybranych odporności. Uwzględnia zaznaczone pola wyboru dotyczące odporności na leki przeciwwirusowe, środki dezynfekcyjne i zmiany pH.
 GetSelectedSymptoms() Metoda uzyskiwania wybranych objawów.
 CalculateInfectivity(): Metoda obliczania zakaźności.
 CalculateMortality(): Metoda obliczania śmiertelności.

Class Country

1) Opis classa

Class Country to klasa jednego kraju

2) Zmienne

“healthyCountries”: Zmienna statyczna śledząca liczbę zdrowych krajów.
 “dateInfected”: Data, kiedy kraj został zainfekowany.
 “infectedCountries”: Zmienna statyczna śledząca liczbę zainfekowanych krajów.
 “isInfected”: Zmienna logiczna wskazująca, czy kraj jest zainfekowany.
 “row”: Numer linii kraju na mapie.
 “col”: Numer kolumny kraju na mapie.
 “population”: Instancja klasy Population reprezentująca populację kraju.
 “eventSetGray”: Obsługa zdarzenia zmiany statusu kraju (szary).
 “eventPick”: Obsługa zdarzeń dla wyboru kraju.
 “color” Kolor kraju.
 “area”: Wielokąt (sześciokąt) reprezentujący kraj na mapie.
 “isCountry”: Zmienna logiczna wskazująca, czy obiekt jest prawdziwym krajem.

3) Metody klasowe

Country(Polygon hexagon, String countryName, double populationDensity, int quantity, double stability, double averageTemperature, boolean borders, double medicalLevel): Konstruktor, który inicjuje kraj z podanymi parametrami.

Country(String countryName): Konstruktor inicjujący kraj o podanej nazwie.

getArea(): Zwraca wielokąt reprezentujący kraj.

setIsCountry(boolean isCountry): Ustawia, czy obiekt jest rzeczywistym krajem i aktualizuje liczbę zdrowych krajów.

getIsCountry(): Zwraca, czy obiekt jest rzeczywistym krajem.

setColour(Paint colour): Ustawia kolor kraju.

getColour(): Zwraca kolor kraju.

setEventSetGray(EventHandler<MouseEvent> eventSetGray): Ustawia obsługę zdarzenia zmiany statusu kraju.

getEventSetGray(): Zwraca program obsługi zdarzeń dla zmiany statusu kraju.

getDateInfected: Zwraca datę, kiedy kraj został zainfekowany.

setZeroValues(): Resetowanie liczników dla zainfekowanych i zdrowych krajów.

setEventPick(EventHandler<MouseEvent> eventPick): Ustawia obsługę zdarzenia wyboru kraju.

getEventPick(): Zwraca obsługę zdarzenia wyboru kraju.

setIsInfected(): Ustawia kraj jako zainfekowany i aktualizuje liczbę zainfekowanych krajów.

getIsInfected(): Zwraca, czy kraj jest zainfekowany.

`getHealthyCountries()`: Zwraca liczbę zdrowych krajów.
`getInfectedCountries()`: Zwraca liczbę zainfekowanych krajów.

Diagramy klas i obiektów

Diagram obiektów

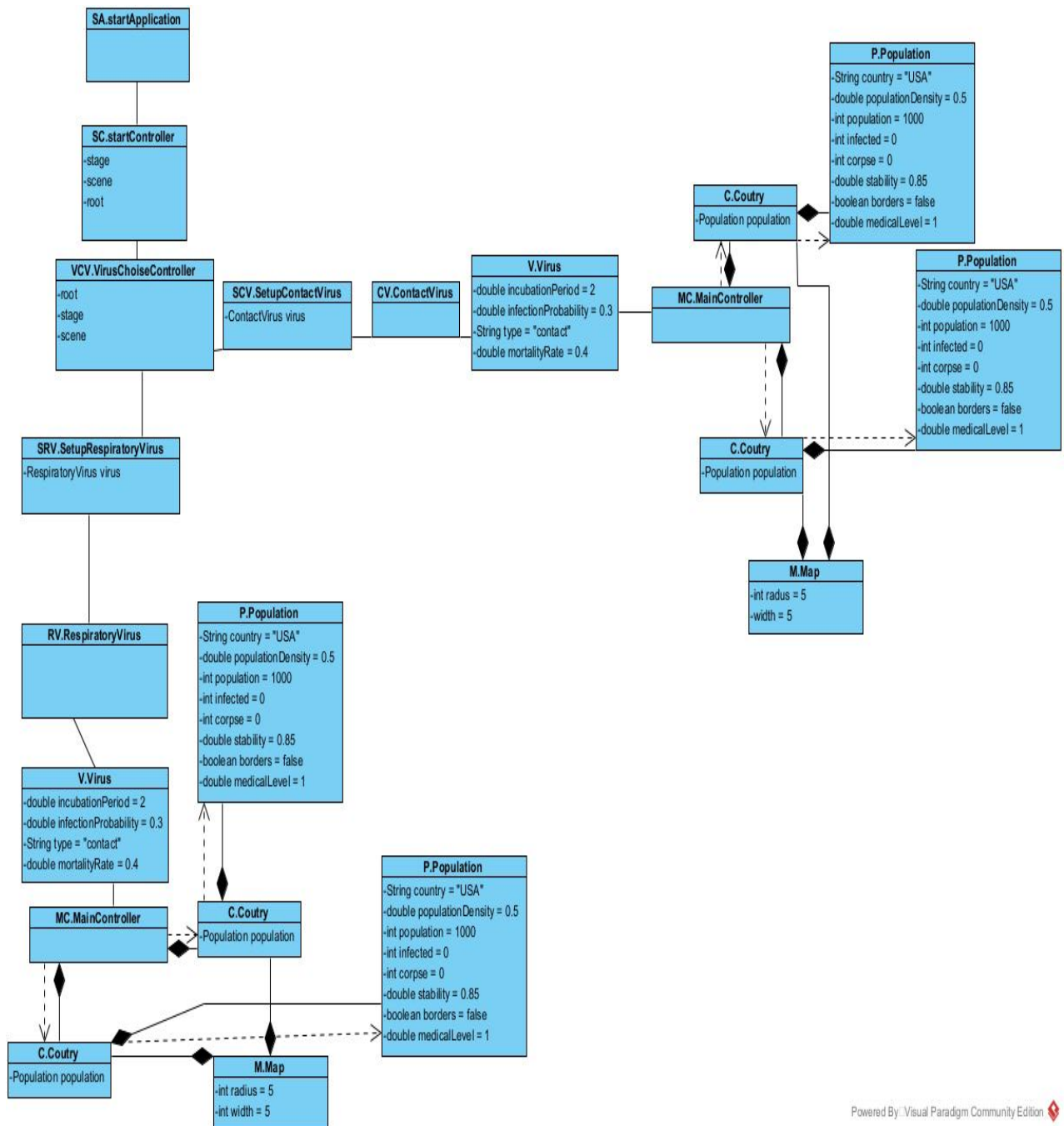


Diagram klas

