

Projekt PO

Maciej Padula, Aleksandra Piątek

Kwiecień 2022

Spis treści

1	Wstęp	2
2	Tematy projektu	2
2.1	Symulacja ludzkiego kodu genetycznego	2
2.2	Symulacja sawanny	2
3	Wybrany temat	2
3.1	Diagram klas	3
3.2	Diagram obiektów	4
4	Dokumentacja	4

1 Wstęp

Na zajęcia laboratoryjne z kursu Programowanie Obiektowe zostaliśmy poproszeni o wymyślenie dwóch tematów symulacji jakiegokolwiek wycinka rzeczywistości oraz o ich opisanie. Jeden z tematów zostanie wybrany przez prowadzącego, a następnie przez nas wdrożony.

Projektem będzie symulacja napisana w języku Java z wykorzystaniem narzędzia do automatycznej kompilacji - gradle. Etapy projektu będą umieszczane na koncie github: Maciej Padula

Skład grupy:

- Maciej Padula, 263919
- Aleksandra Piątek, 264004

2 Tematy projektu

Postanowiliśmy zaproponować dwa następujące tematy projektu:

2.1 Symulacja ludzkiego kodu genetycznego

Symulacja polegać będzie na wygenerowaniu losowym listy ludzi których ilość zostanie wybrana przez użytkownika. Celem symulacji będzie obserwacja zmiany uproszczonego kodu genetycznego po zadanej ilości epok (jedna epoka oznaczać będzie jeden rok). Uproszczony kod genetyczny zawierać będzie informacje o płci, kolorze włosów i oczu oraz bezpłodności. Każda informacja genetyczna będzie posiadała dwa statusy, tzn. nieuszkodzony i uszkodzony. Generowanie nowego kodu genetycznego będzie odbywało się losowo w oparciu o kod genetyczny rodziców (50% na otrzymanie genu matki i 50% ojca). Do uszkodzenia danego genu dochodzi losowo (5%), jeżeli jeden z rodziców posiada uszkodzony gen, lub jeśli ich geny pokrywają się w co najmniej 80%. Jeżeli uszkodzony jest gen płci osoba staje się z automatu bezpłodna. Każda osoba będzie posiadała informacje o swoich rodzicach, listę dzieci oraz informacje o partnerze (zakładamy tylko jednego partnera danej osoby). Po zakończeniu symulacji do pliku JSON zostanie zapisana lista kodów genetycznych ludzi, którzy żyli w ostatniej jej epoce.

2.2 Symulacja sawanny

Na sawannie (tablica dwuwymiarowa) losowo wygenerowane zostają różne zwierzęta. Zwierzęta będą dzieliły się na trzy typy, roślinożerca, drapieżnik oraz padlinożerca. Zwierzęta będą różniły się gatunkami (np. lew, słoń, zebra, hiena). Podczas każdej epoki zwierzę przemieszcza się w losowym kierunku o ilość pól która różni się w zależności od gatunku. Jeżeli na jednym polu spotykają się dwa zwierzęta dochodzi do zdarzenia:

- Jeżeli są to dwa drapieżniki to rozpoczynają walkę. Atakują się nawzajem i zadają ilość obrażeń w zależności od gatunku. Walczą dopóki jeden z nich nie umrze.
- Jeżeli na jednym polu pojawia się roślinożerca i drapieżnik, roślinożerca ma 20% szans na ucieczkę, jeżeli się mu to uda to przesuwa się o 4 pola w losowym kierunku. W innym wypadku drapieżnik zjada swoją ofiarę i regeneruje swoje zdrowie o ilość zdrowia ofiary.
- Jeżeli na jednym polu spotykają się dwaj roślinożercy, padlinożercy lub jeden padlinożerca i jeden roślinożerca, to zwierzę wchodzące idzie na inne losowe pole.
- Jeżeli na polu spotykają się padlinożerca i drapieżnik, to padlinożerca stara się uciec, ma na to 20% szans. Jeżeli mu to się uda to przemieszcza się o 4 losowe pola. W innym przypadku zostaje zabity przez drapieżnika, jednakże drapieżnik nie zjada go i nie regeneruje się.

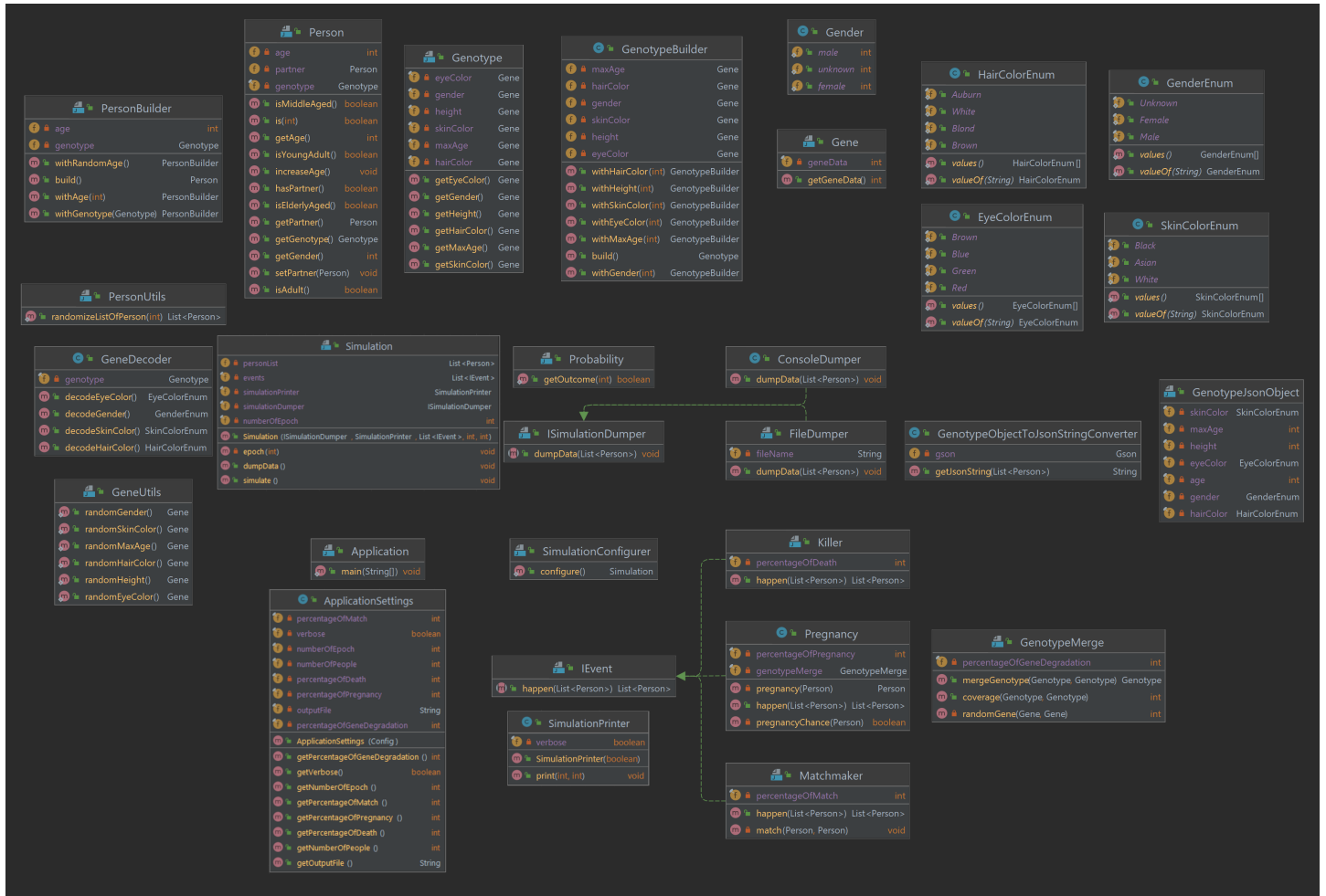
Użytkownik będzie mógł podać ilość każdego typu zwierząt. Ich gatunki zostaną wylosowane. Wynikiem symulacji będzie lista(zapisana w pliku JSON) zwierząt, które pozostały przy życiu.

3 Wybrany temat

Jako temat naszego projektu postanowiliśmy wybrać temat nr 1 tzn. symulacja ludzkiego kodu genetycznego. Tak jak powyżej napisaliśmy podczas każdej epoki symulacji ludzie, którzy na początku programu zostaną losowo wygenerowani łączą się w pary. Po dobraniu się w pary dochodzi do tworzenia potomstwa, które może powstać jedynie w przypadku par o różnych płciach. Podczas każdej epoki każdy człowiek ma zadaną przez użytkownika szansę w % na śmierć z przyczyn losowych. W drugim przypadku osoba umiera jeżeli jej wiek jest większy niż maksymalny wiek zapisany w jego kodzie genetycznym.

3.1 Diagram klas

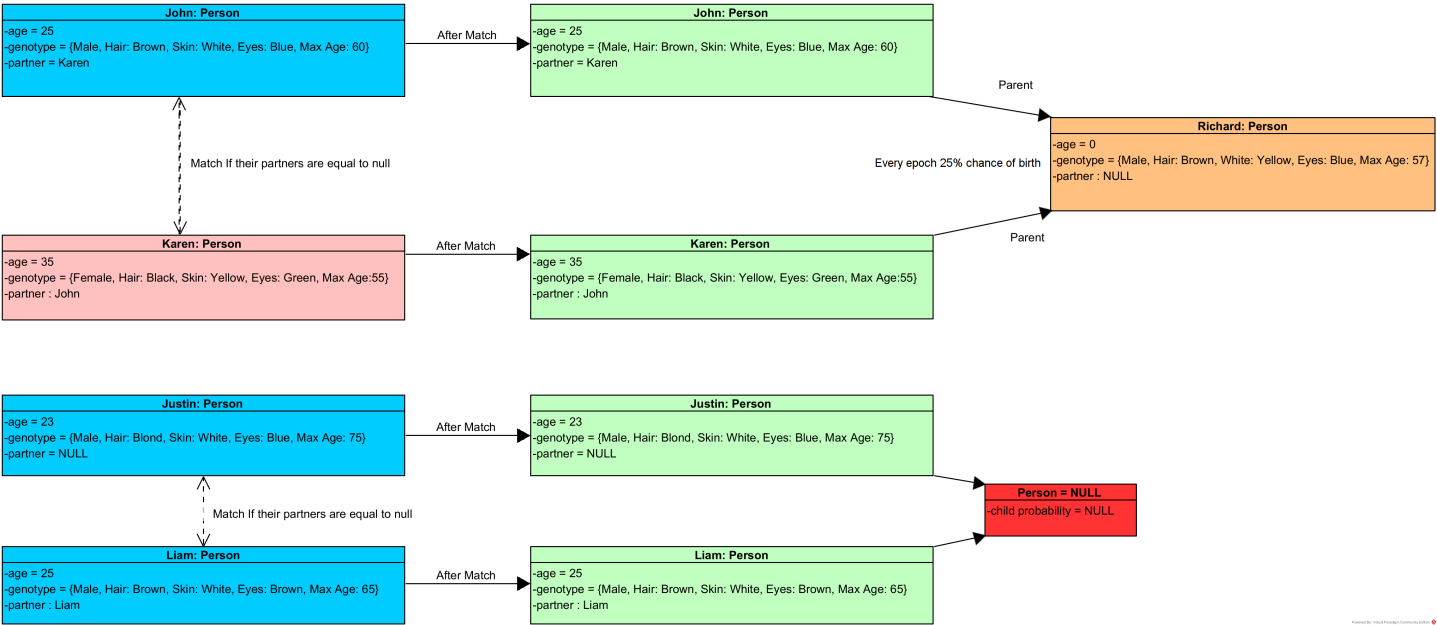
Symulacja będzie składać się z głównej klasy `Simulation`. Będzie ona przechowywała pola klas `Matchmaker`, `Pregnancy`, `Probability` oraz `Killer`. Zdarzenia będą losowe. Procentowa szansa na wykonanie się danego zdarzenia jest modyfikowalna przy pomocy zmiany odpowiednich wartości w pliku `application.json`. Obiekty tych klas będą odpowiadać za interakcje między agentami symulacji (obiekty klasy `Person`). Każda osoba posiada swój genotyp pod postacią obiektu klasy `Genotype`. Każdy genotyp posiada informacje o genach (Klasa `Gene`). Oto diagram klas wraz z zależnościami:



Rysunek 1: Diagram klas

3.2 Diagram obiektów

Agentami symulacji będą osoby losowo wygenerowane podczas uruchamiania symulacji. Podczas każdej epoki każda osoba która nie posiada partnera będzie go szukać. Obiekt klasy matchmaker będzie dokonywać połączeń. Jeżeli uzna, że połączenie dwóch osób jest prawidłowe to ustawi ich pola partner naprzemiennie. Od następnej epoki powstaje możliwość urodzenia dziecka. O urodzeniu nowego dziecka decyduje obiekt klasy Pregnancy. Poniżej przykładowy diagram obiektów w systemie:



Rysunek 2: Diagram obiektów

4 Dokumentacja

Link do dokumentacji kodu na stronie github.io: <https://maciejpadula.github.io/ProjektPO>