

Programowanie Obiektowe – Projekt Symulacja Ewolucji kolonii bakterii

Nazwa Symulacji: Ewolucja kolonii bakterii

Skład grupy: Wojciech Grzybowski

Lider grupy: Wojciech Grzybowski

Język programowania: Java

Opis projektu:

Symulacja przedstawia rozwój dwóch różnych kolonii bakterii. W symulacji są tworzone bakterie typu A i B, a następnie stworzone bakterie są dodawane do mapy dwuwymiarowej (2D). Wyróżniamy dwie podgrupy bakterii: małe oraz duże (doroste). Dodatkowo na mapie pojawia się wylosowana liczba budulca (zależna od wybranego ziarna). Budulec jest odpowiedzialny za wzrost bakterii, po „wchłonięciu” przez bakterię odpowiedniej liczby budulca (równej jeden) mała bakteria przeistacza się w dużą bakterię, która z kolei powtarza poprzednią czynność, aż będzie gotowa do podziału komórkowego. Duża bakteria gotowa do podziału (po wykorzystaniu jednego budulca) dzieli się na dwie małe bakterie. W symulacji obowiązuje ruch na tury - bakteria danego typu wykonuje ruch wyłącznie wtedy gdy jest jej kolej. Ruch bakterii – małe bakterie przemieszczają się o jedną pozycję, a duże o dwie pozycje. Dodatkowo bakterie poruszają się w dowolnym kierunku (tzn. lewo, prawo, góra, dół oraz po przekątnej analogicznie). Możliwa jest również bezczynność bakterii – w tym przypadku bakteria pozostaje na swojej pozycji podczas trwania rundy (tury) bakterii tego typu. Symulacja kończy się po zadanych przez użytkownika czasie. Na koniec program pokazuje podsumowanie symulacji – liczba małych, dużych bakterii danego typu, liczbę wszystkich bakterii danego typu (tj. suma bakterii małych i dużych) czas trwania symulacji całej symulacji.

Główne Funkcje Symulacji:

- 1) Tworzenie mapy:
 - a) Wielkość mapy (w kształcie kwadratu)
- 2) Dodawanie bakterii typu A i B do mapy:
- 3) Rozmieszczanie bakterii na mapie w sposób pseudo-losowy
- 4) Ustawianie warunków dla symulacji tj.:
 - a) Czas symulacji
 - b) Liczność bakterii typu A i B
 - c) Długość i szerokość mapy
- 5) Ruch bakterii po mapie (każda z kolonii bakterii stara się rozprzestrzenić na jak największą skalę):
 - a) Przesuwanie się małej bakterii o jedną pozycję w dowolnym wylosowanym kierunku
 - b) Przesuwanie się dużej bakterii o dwie pozycje w dowolnym wylosowanym kierunku
- 6) Pojawianie się budulca (powodującego wzrost bakterii) na mapie:
 - a) Ustawienie ilości pojawiającego się budulca, po przez wylosowanie liczby pseudo-losowej

- 7) Wzrost bakterii oraz podział komórkowy (jedna bakteria dzieli się na dwie małe):
 - a) Wzrost bakterii „wchłoniętego” budulca po której następuje wzrost bakterii małej do bakterii dużej
 - b) Podział bakterii następuje w wyniku wchłonięcia budulca przez dużą (dorosłą) bakterię. Po wchłonięciu budulca bakteria rozdziela się na dwie małe bakterie danego rodzaju
- 8) Menu:
 - a) Wyjście z programu
 - b) Ustawienie symulacji
 - c) Włączenie symulacji
- 9) Pokazanie końcowych wyników symulacji:
 - a) Ilość małych i dużych bakterii danego typu przed rozpoczęciem symulacji oraz po zakończeniu symulacji
 - b) Ilość wszystkich bakterii danego typu (suma małych i dużych bakterii)
 - c) Czas trwania symulacji
 - d) Zapis wyników do pliku (txt) w folderze Simulation (nazwa pliku: Simulation_Results.txt)

Przykładowa symulacja - zakończona

```
[0, 0, *, *, 0, 0, 0, 0, 0, 0, *, *, *, 0, *, 0, 0, 0, 0, *, *, *, 0, *, 0, *, 0, *, 0, 0]
[0, *, *, 0, 0, a, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *, 0, 0, 0, 0, *, 0, 0, 0, 0]
[*, *, 0, 0, 0, 0, 0, 0, 0, *, 0, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, *, 0, 0, 0, 0, 0, 0]
[0, *, *, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *, 0, 0, 0, 0, b]
[0, *, *, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *, 0, 0, 0, 0, 0]
[0, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *, *, 0, 0, *, *, 0, 0, 0, 0, *, 0, 0]
[0, 0, 0, 0, 0, 0, 0, *, 0, *, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *, b]
[*, 0, 0, 0, *, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, *, 0]
[*, 0, 0, 0, 0, *, 0, 0, 0, 0, *, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, a]
[*, 0, 0, 0, 0, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, a]
[0, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[*, *, *, *, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[*, 0, *, *, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[*, 0, 0, 0, 0, *, 0, 0, A, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, A, a]
[*, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[0, *, *, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[*, *, 0, *, *, 0, a, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *, b]
[0, 0, 0, a, 0, 0, a, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, *, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[*, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[0, *, 0, 0, a, *, 0, *, A, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[0, 0, 0, A, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, *, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[*, 0, *, *, 0, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, b]
[0, *, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[*, *, *, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, b]
[*, 0, *, *, 0, 0, 0, 0, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, b]
[*, *, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[0, 0, *, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, 0]
[*, 0, *, 0, 0, 0, 0, a, 0, 0, 0, 0, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[0, 0, *, *, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, a]
[0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Mapa Symulacyjna po zakończeniu symulacji

Simulation data:

Width (x) of 2D map: 35
Lenght (y) of 2D map: 35

Before simulation:

- 1)Number of adult Bacteria type A: 0
- 2)Number of adult Bacteria type B: 0
- 3)Number of child Bacteria type a: 18
- 4)Number of child Bacteria type b: 17
- 5)Number of Bacteria type A: 18
- 6)Number of Bacteria type B: 17

After simulation:

- 1)Number of adult Bacteria type A: 5
- 2)Number of adult Bacteria type B: 13
- 3)Number of child Bacteria type a: 42
- 4)Number of child Bacteria type b: 42
- 5)Number of Bacteria type A: 47
- 6)Number of Bacteria type B: 55
- 7)Total time of simulation or number of maximum iterations: 1 seconds

Wyniki Symulacji

Legenda:

- Zielonym kolorem zostały oznaczone bakterie typu B
- Duża litera np. (A) oznacza dużą bakterię typu A gotową do podziału
- Mała litera np.(b) oznacza małą bakterię typu B (niegotową do podziału)
- Symbol np.(*) oznacza budulec

Diagram klas UML

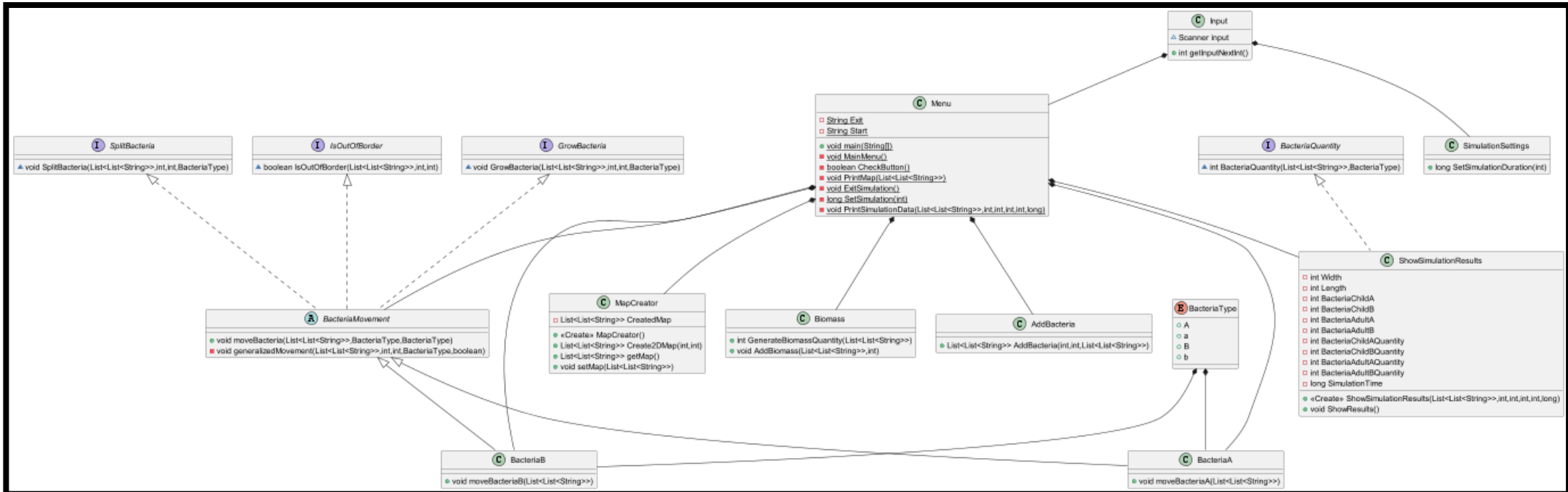


Diagram obiektów UML:

