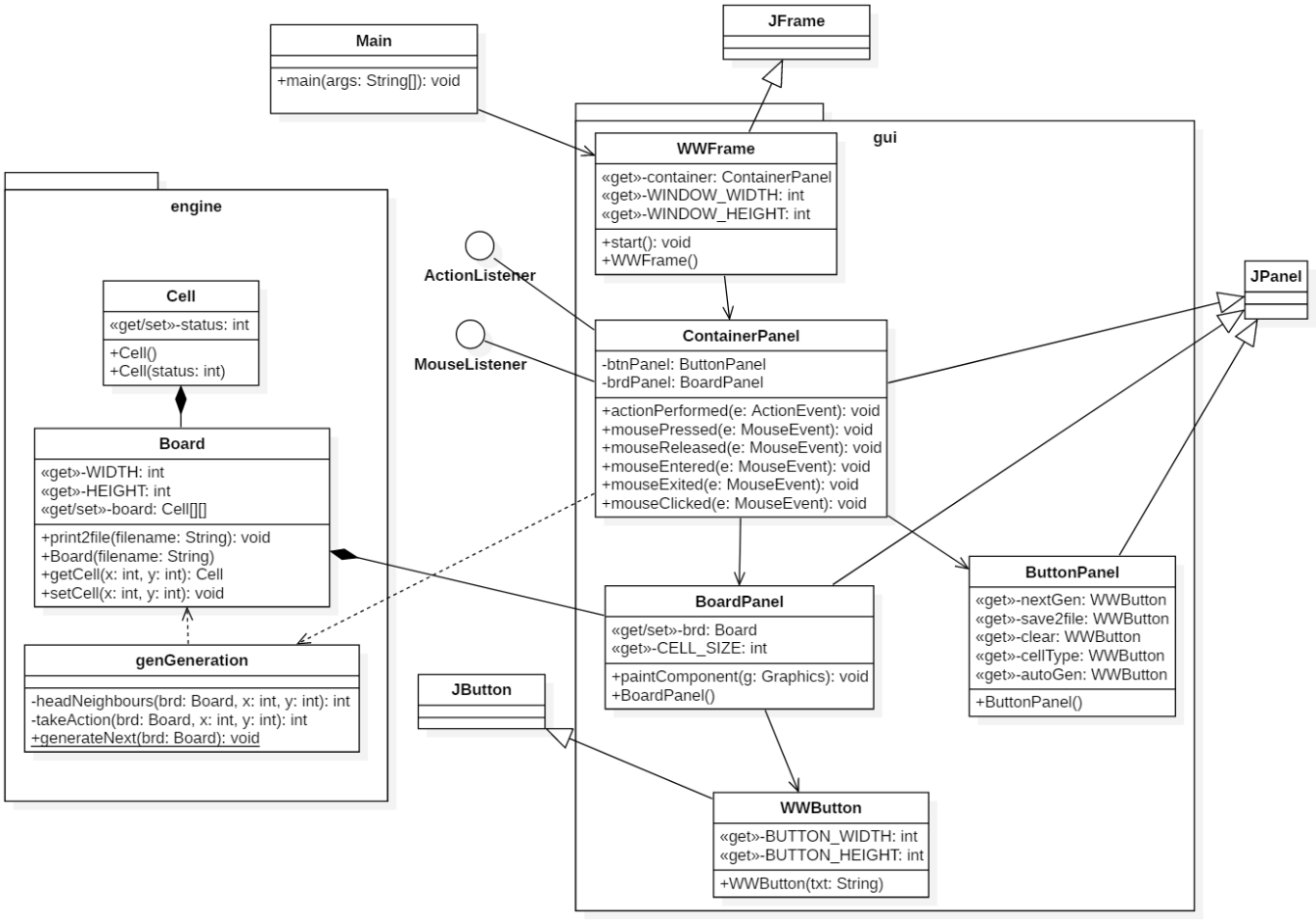
Specyfikacja implementacyjna programu *WireWorld*

Krzysztof Maciejewski
Hubert Kunikowski

9 czerwca 2019

Spis treści

1 Diagram klas



Program składa się z 9 klas.

Klasy z pakietu *engine*, odpowiadające za generowanie i przechowywanie tablicy, to:

- *Cell* - przechowuje status komórki.
- *Board* - przechowuje tablicę komórek (obiektów *Cell*).
- *genGenerator* - odpowiada za generowanie kolejnych generacji.

Klasy z pakietu *gui*, odpowiadające za wyświetlanie i działanie okienka, to:

- *WWFrame* - dziedziczy po *JFrame*, odpowiada za wyświetlanie okna.
- *ContainerPanel* - dziedziczy po JPanel, przechowuje zawartość okna.
- *ButtonPanel* - dziedziczy po JPanel, odpowiada za wyświetlanie i działanie przycisków.
- *BoardPanel* - dziedziczy po JPanel, odpowiada za wyświetlanie i działanie planszy.

- *WWButton* - dziedziczy po JButton, określa właściwości przycisków.

Klasa *Main* odpowiada za pobranie opcjonalnego argumentu wywołania oraz uruchomienie całego programu.

2 Opis klas

1. **Klasa Cell**

   Klasa ta zawiera jeden atrybut typu int określający stan komórki:
   1 - pusta,
   2 - głowa elektronu,
   3 - ogon elektronu,
   4 - przewodnik.

2. **Klasa Board**

   Klasa ta zawiera dwie stałe określające wysokość i szerokość tablicy oraz atrybut przechowujący planszę, czyli dwuwymiarową tablicą instancji klasy *Cell*.

   Konstruktor klasy *Board* pobiera jako argument nazwę pliku, z którego odczyta wstępną konfigurację komórek. Dane w pliku powinny być w postaci:

   ```
   1 2 2
   2 4 5
   4 3 2
   3 5 6
   ```

   , gdzie pierwsza liczba w wierszu oznacza stan komórki, druga współrzędną x, a trzecia współrzędną y. W przypadku błędnych danych wejściowych (np.: podane współrzędne nie zgadzają się z wymiarami tablicy), program zwróci błąd.

   Klasa ta zawiera również metodę *print2file*, która zapisuje planszę do pliku, w postaci identycznej jak przy pliku wejściowym.

3. **Klasa genGenerator**

   Klasa ta zawiera 3 metody. Pierwsza z nich, *headNeighbours*, pobiera planszę i zlicza ile głów elektronu znajduje się w sąsiedztwie Moore'a komórki o podanych współrzędnych dla danej planszy.

   Powyższa metoda jest wykorzystywana w drugiej metodzie, *nextStatus*. Pobiera ona planszę i zwraca nowy status dla komórki o podanych współrzędnych.

   Trzecia metoda, *generateNext*, jest publiczna i statyczna. Pobiera ona obiekt typu *Board* i, korzystając z funkcji *nextStatus*, przekazuje do obiektu nową planszę.

4. **Klasa WWFrame**

   Klasa ta dziedziczy po klasie *JFrame* z biblioteki Swing. Zawiera 2 stałe określające wymiary okna oraz atrybut typu *ContainerPanel*, który zawiera zawartość okna.

   Klasa posiada konstruktor określający właściwości okna.

5. **Klasa ContainerPanel**

   Klasa *ContainerPanel* dziedziczy po klasie *JPanel*. Służy do przechowywania innych paneli.

   Klasa ta zawiera dwa atrybuty typu *ButtonPanel* oraz *BoardPanel*.

Klasa ta również implementuje interfejsy *ActionListener* oraz *MouseListener*. Określone są działania związane z naciśnięciem konkretnego przycisku lub klinknięciem(zmianą stanu) komórki na planszy.

6. **Klasa ButtonPanel**

   Klasa *ButtonPanel* posiada cztery atrybuty odpowiadające konkretnym przyciskom typu *WWButton*.

   Przycisk "Next generation" wygeneruje kolejną generację.

   Przycisk "Autogenerate" zacznie automatycznie generować kolejne generację z określonym opóźnieniem.

   Przycisk "Save to file" zapisze generację do pliku tekstowego.

   Przycisk "Clear" wyczyści planszę.

   Ostatni przycisk posłuży do wybrania jaką komórkę chcemy wstawić na planszę. Przycisk może znajdować się w trzech stanach: "Connector", "Electron head" oraz "Electron tail".

7. **Klasa BoardPanel**

   Klasa ta dziedziczy po *JPanel* i odpowiada za wyświetlanie planszy. Zawiera atrybuty typu *Board* przechowujący planszę oraz atrybut *cellSize* określający wymiary komórki w pikselach.

   Nadpisana metoda *paintComponent* służy do rysowania planszy. Jest wywoływana za każdym razem kiedy atrybut *Board* jest aktualizowany.

8. **Klasa WWButton**

   Klasa *WWButton* dziedziczy po klasie *JButton* i służy do określenia właściwości wspólnych dla wszystkich przycisków. Pozwala to zaoszczędzić powtarzania kodu. Wszystkie przyciski będące atrybutami *ButtonPanel* są typu *WWButton*.

   Właściwości przycisków są określane przy wywoływaniu konstruktora klasy, który pobiera również argument typu String, określający napis na danym przycisku.

9. **Klasa Main**

   Metoda *main* pobiera jeden, opcjonalny argument, będący nazwą pliku wejściowego z generacją. Jeżeli argument nie zostanie podany, program wczyta domyślny plik tekstowy z generacją z folderu *resources*.

3 Opis przepływu sterowania

1. Pobranie nazwy pliku z generacją wejściową.
2. Utworzenie obiektu *WWFrame*.

   2.1 Utworzenie *ContainerPanel*.

   2.1.1 Utowrzenie i dodanie *ButtonPanel* oraz *BoardPanel* do *ContainerPanel*.

   2.1.1.1 Utorzenie i dodanie *WWButton* do *ButtonPanel*.
3. Utworzenie obiektu typu *Board*.
4. Wywołanie metody *WWFrame.start()* - dodanie *ContainerPanel* do *WWFrame*.
5. Ewentualne zakończnie działania programu poprzez zamknięcie okna.

4 Opis głównych algorytmów

Generowanie kolejnej generacji

Kiedy wciśniemy przycisk "Next Generation", wywołana zostaje metoda *actionPerformed* w obiekcie nasłuchującym *ContainerPanel*. Wewnątrz metody rozpoznywane jest źródło sygnału. Wywoływana jest metoda statyczna klasy *genGenerator*: *generateNext* na rzecz atrybutu *Board* obiektu *BoardPanel*. Kiedy plansza jest zaktualizowana, wywoływana jest metoda *paintComponent* na rzecz obiektu *BoardPanel*.

Zmiana stanu komórki na planszy

Kiedy klikniemy myszką w obrębie planszy, wywoływana jest metoda *mouseClicked* w obiekcie nasłuchującym *ContainerPanel*. Pobierane są współrzędne kliknięcie. Bazując na wymiarach planszy, rozmiarze komórki oraz wymiarach panelu, rozpoznaje się nad którą komórką nastąpiło kliknięcie. Następnie, zależnie od aktualnego stanu tej komórki oraz od aktualnego stanu przycisku służącego do wybrania konkretnej komórki do wstawienia, przypisany zostaje nowy stan. Na koniec wywołana jest metoda *paintComponent* na rzecz obiektu *BoardPanel*.