

Specyfikacja funkcjonalna projektu grupowego  
z przedmiotu *Algorytmy i struktury danych*

Krzysztof Kowalski, Szymon Motłoch, Aleksander Wodnicki

17 grudnia 2020

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
1.1	Cel dokumentu . . . . .	3
1.2	Cel projektu . . . . .	3
1.3	Opis symulacji . . . . .	3
<b>2</b>	<b>Opis działania programu</b>	<b>4</b>
2.1	Opis graficznego interfejsu . . . . .	4
2.2	Dane wejściowe . . . . .	5
2.2.1	Plik z informacjami o mapie . . . . .	6
2.2.2	Plik z informacjami o pacjentach . . . . .	7
2.3	Uruchomienie programu . . . . .	8
2.4	Scenariusz działania programu . . . . .	8
2.5	Komunikaty błędów . . . . .	9
<b>3</b>	<b>Testowanie</b>	<b>10</b>

# 1 Wstęp

## 1.1 Cel dokumentu

Niniejszy dokument zawiera informacje na temat tworzonej aplikacji w ramach projektu grupowego z przedmiotu *Algorytmy i struktury danych*. Opisuje on zasady działania programu, tj. sposób uruchomienia, przyjmowane dane, sposób działania, a także dane wyjściowe otrzymywane po jego zakończeniu. Co więcej, opisuje on sposób testowania programu oraz interpretację komunikatów o błędach. W skrócie – użytkownik, po zapoznaniu się z tym dokumentem, będzie w stanie bez problemu uruchomić omawiany program i posługiwać się nim.

## 1.2 Cel projektu

Celem projektu jest utworzenie aplikacji z interfejsem graficznym, która będzie przeprowadzała symulacje działania pracy pogotowia ratunkowego w czasach pandemii. Na bazie informacji znajdujących się w dostarczonych przez użytkownika plikach, zostanie utworzona mapa, a na niej umiejscowione osoby oczekujące na pomoc ze strony pogotowia. Na bazie informacji o tym, gdzie znajdują się szpitale oraz jakie istnieją drogi, pogotowie będzie przemierzać kraj w poszukiwaniu najbliższego szpitala, który nie został jeszcze odwiedzony przez karetkę. Celem tych podróży będzie znalezienie wolnego miejsca w szpitalu dla pacjenta.

## 1.3 Opis symulacji

W tej sekcji znajduje się szczegółowy opis teoretycznych założeń projektu, które definiują jak symulacja będzie przebiegać.

Naszą mapą jest najmniejszy zbiór wypukły zawierający w sobie wszystkie obiekty z pierwszego pliku. Po wygenerowaniu takiej mapy, umieszczamy na niej pacjentów według współrzędnych z drugiego pliku. Pacjenci będą rozmieszczeni w takiej kolejności, w jakiej znajdowali się w pliku. Jeśli pacjent został dodany pojedynczo poprzez podanie współrzędnych w oddzielnym formularzu, to będzie obsługiwany jako ostatni.

W symulacji nie występuje ograniczenie ilości karetek. Dla każdego pacjenta, który pojawi się na terenie kraju, zostanie od razu rozdysponowana do niego karetka. W żaden sposób nie zajmujemy się pacjentami poza granicami naszego kraju. Ponieważ pacjent może mieć współrzędne niezajdujące się na zdefiniowanych drogach, karetka, która po niego przybędzie pokona drogę

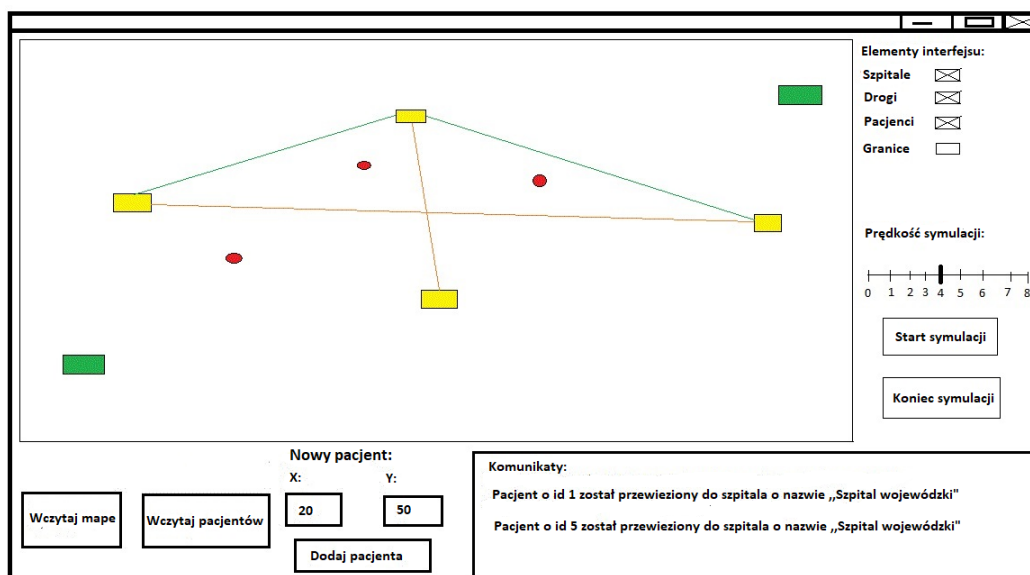
jadąc po linii prostej do najbliższego szpitala. Jeśli w szpitalu będzie znajdować się miejsce dla pacjenta, zostanie on do niego przyjęty i liczba wolnych łóżek znajdujących się w szpitalu zostanie pomniejszona. W przeciwnym razie, jeśli nie będzie miejsc w szpitalu, karetka będzie jeździć po kolejnych szpitalach i szukać wolnego miejsca dla pacjenta. Karetka nie będzie brała pod uwagę ilości wolnych miejsc w szpitalach, tylko odległość między nimi i to, czy zostały już wcześniej odwiedzone. Jeśli karetka odwiedzi wszystkie szpitale i nie znajdzie wolnego miejsca dla pacjenta, będzie stała pod ostatnim z odwiedzonych szpitali czekając aż zwolni się miejsce w tym szpitalu.

## 2 Opis działania programu

W tej sekcji zostaną opisane zagadnienia związane z działaniem programu, tj. sposoby i scenariusze uruchomienia, dane wejściowe, graficzny interfejs symulacji, jak również komunikaty błędów.

### 2.1 Opis graficznego interfejsu

Tak wygląda poglądowy projekt graficznego interfejsu aplikacji. Poniżej opisane zostały wszystkie jego istotne elementy.



Rysunek 1: Projekt graficznego interfejsu gry

- **Wczytaj mapę** – przycisk służący do wczytania pliku z danymi o szpitalach, obiektach i drogach;
- **Wczytaj pacjentów** – przycisk służący do wczytania pliku ze współrzędnymi pacjentów;
- **Nowy pacjent** – pole, w którym możemy dodać współrzędne nowego pacjenta bez konieczności wczytywania pliku;
- **Dodaj pacjenta** – przycisk służący do dodania powyższego pacjenta, którego współrzędne wprowadziliśmy w odpowiednich komórkach do tego przeznaczonych;
- **Komunikaty** – miejsce, w którym użytkownik będzie otrzymywał informacje o kolejnych etapach symulacji;
- **Prędkość symulacji** – suwak służący do określenia prędkości, z jaką symulacja będzie przebiegać;
- **Start symulacji** – przycisk służący do rozpoczęcia symulacji;
- **Koniec symulacji** – przycisk służący do zatrzymania symulacji przed jej zakończeniem;
- **Elementy interfejsu** – miejsce, w którym użytkownik może odznaczyć obiekty według swoich preferencji. Skutkuje to ich zniknięciem z mapy i może poprawić czytelności symulacji.

## 2.2 Dane wejściowe

Danymi wejściowymi, które powinien zapewnić użytkownik, aby możliwe było rozpoczęcie symulacji, są dwa pliki w formacie `.txt`. W obu plikach poszczególne sekcje powinny zostać rozdzielone odpowiednimi nagłówkami. Linia pliku, w której znajduje się nagłówek rozpoczyna się znakiem `#`. Nagłówki powinny wyglądać identycznie jak nagłówki, które będą przedstawione w przykładach w dalszej części dokumentu. Ponadto, wartości w liniach powinny być podzielone na części separatorami `|`.

### 2.2.1 Plik z informacjami o mapie

W pierwszym pliku wejściowym użytkownik będzie mógł zdefiniować elementy takie jak szpitale, obiekty oraz drogi. Na podstawie tego pliku będzie wygenerowana mapa, na terenie której przeprowadzona zostanie symulacja.

1. **Szpitale** – będą posiadać współrzędne na mapie oraz informacje o tym, ile jest łóżek w szpitalu, a także ile z nich jest wolnych. Będą one celem karetek pogotowia, które będą poszukiwać szpitali z wolnymi łózkami. Po przewiezieniu pacjenta do szpitala liczba wolnych łóżek zmaleje.

- id jest liczbą całkowitą nieujemną (każde id oryginalne);
- nazwa może zawierać dowolne znaki oprócz separatora | (znaki białe są ignorowane);
- współrzędne x i y są liczbami całkowitymi (mogą być ujemne);
- liczba łóżek i liczba wolnych łóżek to liczby całkowite nieujemne mniejsze od 100 000. Liczba wolnych łóżek nie może być większa niż liczba łóżek.

2. **Obiekty** – służyć będą wyznaczeniu dodatkowych punktów ograniczających granice kraju. Granica kraju będzie wyznaczana na podstawie najbardziej wypukłych elementów mapy (szpitali i obiektów). Pogotowie będzie przewozić wyłącznie osoby znajdujące się w granicach kraju.

- id jest liczbą całkowitą nieujemną (każde id oryginalne);
- nazwa może zawierać dowolne znaki oprócz separatora | (znaki białe są ignorowane);
- współrzędne x i y są liczbami całkowitymi (mogą być ujemne).

3. **Drogi** – będą określały połączenia występujące między danymi szpitalami. Nie wszystkie szpitale muszą być ze sobą bezpośrednio połączone. Dla każdej drogi będzie podana informacja o tym, ile zajmuje jej przebycie. W miejscach, gdzie drogi będą się przecinać, będą występować skrzyżowania. Będzie to poszerzało możliwości poruszania się karetek, umożliwiając im skręcenie w inną drogę na skrzyżowaniu.

- id jest liczbą całkowitą nieujemną (każde id oryginalne);
- identyfikatory szpitali muszą występować w sekcji **Szpitale**. Każde połączenie musi być oryginalne;
- odległość jest liczbą całkowitą nieujemną mniejszą od 1 000 000.

Przykładowa zawartość tego pliku wygląda następująco:

```
# Szpitale (id | nazwa | wsp. x | wsp. y | Liczba łóżek |  
Liczba wolnych łóżek)  
1 | Pierwszy Szpital im. Prezesa RP | 120 | 130 | 99 | 0  
2 | Drugi Szpital im. Naczelnika RP | 10 | 140 | 70 | 1  
3 | Trzeci Szpital im. Króla RP | 140 | 10 | 996 | 0  
# Obiekty (id | nazwa | wsp. x | wsp. y)  
1 | Pomnik Wikipedii | -1 | 50  
2 | Pomnik Fryderyka Chopina | 110 | 55  
# Drogi (id | id_szpitala | id_szpitala | odległość)  
1 | 1 | 2 | 700  
2 | 1 | 3 | 550
```

Przy czym nagłówek **Szpitale** znajduje się w pliku w jednej linii.

### 2.2.2 Plik z informacjami o pacjentach

W drugim pliku znajdują się informacje na temat pacjentów. Wskazana zostaje lokalizacja, do której powinna przybyć po nich karetka. Lokalizacja ta może być dowolnym miejscem i nie musi znajdować się na drodze. Jeśli lokalizacja ta wykracza poza ustalone granice kraju, to taki pacjent nie jest brany pod uwagę.

W tym pliku powinna znaleźć się tylko jedna sekcja, które powinna spełniać poniższe warunki:

#### 1. Pacjenci:

- id jest liczbą całkowitą nieujemną (każde id oryginalne);
- współrzędne x i y są liczbami całkowitymi (mogą być ujemne).

Przykładowa zawartość tego pliku wygląda następująco:

```
# Pacjenci (id | wsp. x | wsp.y)  
1 | 20 | 20  
2 | 99 | 105  
3 | 23 | 40
```

## 2.3 Uruchomienie programu

Program może zostać uruchomiony na dwa sposoby, zależne od preferencji użytkownika:

1. w stosowanym przez użytkownika środowisku programistycznym (np. NetBeans);
2. z konsoli używając komendy:

```
java -jar Pandemic.jar
```

Program nie wymaga podawania żadnych argumentów wywołania, ponieważ wszystkie operacje odbywać się będą z poziomu graficznego interfejsu.

## 2.4 Scenariusz działania programu

1. Wczytanie pliku zawierającego dane szpitali, obiektów i dróg poprzez wciśnięcie przycisku *Wczytaj mapę*.
2. Sprawdzenie poprawności otrzymanych danych w podanym pliku.
3. Wygenerowanie i pokazanie użytkownikowi mapy przedstawiającej zdefiniowane szpitale, obiekty, drogi oraz obszar nimi ograniczony.
4. Przekazanie do programu danych pacjentów, dla których zostanie stworzona symulacja może się odbyć na dwa sposoby:
  - wczytanie pliku zawierającego identyfikatory i współrzędne pacjentów poprzez wciśnięcie przycisku *Wczytaj pacjentów*;
  - podanie współrzędnych pacjenta w polu *Nowy pacjent* (pole to pozostaje aktywne przez cały czas trwania symulacji).
5. Sprawdzenie poprawności danych otrzymanych danych pacjentów.
6. Jeśli wszystkie dane zostały zaakceptowane, odblokowuje się przycisk *Start symulacji* i wciskając go, rozpoczynamy symulację.
7. Symulacja kończy swoje działanie, gdy wszyscy pacjenci zostali obsłużeni lub gdy przyciśniemy przycisk *Koniec symulacji*.
8. Po zakończeniu symulacji można zamknąć program lub załadować nowe dane i przeprowadzić kolejną symulację.



## 2.5 Komunikaty błędów

Komunikaty błędów w tym programie mogą wystąpić w przypadku wprowadzenia danych niezgodnych z wymaganiami określonymi w sekcji *Dane wejściowe*. Jeśli podczas wczytywania plików pojawi się jakiś błąd, to użytkownik zostanie powiadomiony o nim na specjalnym okienku służącym do wyświetlania komunikatów dla użytkownika. Komunikat błędu będzie posiadał dokładną informację o tym jaki błąd został popełniony oraz o tym, w której linii pliku wystąpił. Jeśli pojawi się jakikolwiek błąd, to żaden z elementów mapy występujących w pliku, lub analogicznie żaden pacjent, nie zostanie wczytany. Można wyróżnić następujące komunikaty:

- jeśli program nie znajdzie podanej ścieżki lub pliku z danymi wejściowymi, to zostaniemy o tym poinformowani. Dostaniemy również informację, gdzie powinien on się znaleźć;
- jeśli podany plik z danymi wejściowymi będzie z innym rozszerzeniem niż wymagany (`.txt`), to dostaniemy o tym informację oraz o tym, jakie powinno być rozszerzenie;
- jeśli w znalezionym pliku będzie nieprawidłowy format nagłówek lub jakiś nagłówek nie zostanie znaleziony, to dostaniemy informację o tym, razem z numerem linii, w której wystąpił błąd;
- jeśli w pliku identyfikatory, liczby łóżek i odległości będą opisane innymi znakami niż liczby całkowite nieujemne lub współrzędne nie będą liczbami całkowitymi, to dostaniemy stosowny komunikat wraz z numerem linii, w której się znajduje błąd;
- jeśli w pliku w poszczególnych sekcjach będą się powtarzać identyfikatory, to zostaniemy o tym poinformowani razem z numerem linii;
- jeśli w pliku nazwy szpitali i obiektów będą zawierać znak separatora (`|`), będzie to skutkować niepoprawnym formatem pliku i stosownym komunikatem zawierającym numer linii, w której zaszła nieprawidłowość.

Przykładowy komunikat, który otrzyma użytkownik w przypadku znalezienia dwóch identyfikatorów szpitali o wartości równej 3 będzie wyglądać następująco:

*Identyfikator '3' się powtarza. Identyfikatory muszą być oryginalne. Linia: 8.*

### 3 Testowanie

Testowanie będzie odbywało się poprzez wykorzystanie testów jednostkowych za pomocą narzędzi `JUnit 5` oraz `AssertJ`. W związku z tym, powstanie klasa testowa, w której zostaną przetestowane poszczególne metody, zaczynając od prostych funkcji, a kończąc na tych bardziej rozbudowanych.

### References

- [1] ISOD — Wirtualny Dziekanat:  
`isod.ee.pw.edu.pl`