



**Politechnika  
Śląska**

## **PROJEKT INŻYNIERSKI**

Aplikacja do symulacji rozprzestrzeniania się zarażeń

**Jakub CIOŁEK**

Nr albumu: 295618

**Kierunek:** Informatyka

**Specjalność:** Bazy danych i Inżynieria Systemów

**PROWADZĄCY PRACĘ**

**Dr inż. Ewa Płuciennik**

**KATEDRA Informatyki Stosowanej**

**Wydział Automatyki, Elektroniki i Informatyki**

**Gliwice 2023**



**Tytuł pracy**

Aplikacja do symulacji rozprzestrzeniania się zarażeń

**Streszczenie**

(Streszczenie pracy – odpowiednie pole w systemie APD powinno zawierać kopię tego streszczenia.)

**Słowa kluczowe**

Modelowanie epidemii, Aplikacja edukacyjna, Bezpieczeństwo zdrowotne, Interaktywne narzędzie edukacyjne, Matematyczne modele epidemiologiczne

**Thesis title**

Application for simulating the spread of infections.

**Abstract**

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

**Keywords**

Epidemic modeling, Educational application, Health safety, Interactive educational tool, Mathematical epidemiological models



# Spis treści

1	Wstęp	1
2	Modele symulacji rozprzestrzeniania się zarażeń	3
2.1	Modele bazujące na SIR . . . . .	4
2.2	Modele agentowe . . . . .	5
3	Wymagania i narzędzia	9
4	[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]	11
5	[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]	13
6	Weryfikacja i walidacja	15
7	Podsumowanie i wnioski	17
	Bibliografia	19
	Spis skrótów i symboli	23
	Źródła	25
	Lista dodatkowych plików, uzupełniających tekst pracy	27
	Spis rysunków	29
	Spis tabel	31



# Rozdział 1

## Wstęp

W świecie, który dopiero co doświadczył globalnej pandemii COVID-19, zauważamy potrzebę skutecznych narzędzi zarówno do przewidywania rozprzestrzeniania się infekcji, jak i podnoszenia świadomości społeczeństwa na temat konieczności przestrzegania restrykcji i ochrony zdrowia. Pandemia wywołała potrzebę innowacyjnych rozwiązań, w obszarze przewidywania i zrozumienia dynamiki rozprzestrzeniania się wirusa. W kontekście informatyki, praca skupia się na wykorzystaniu komputerów do opracowania aplikacji, która nie tylko pozwala na modelowanie dynamicznych scenariuszy rozprzestrzeniania się wirusa, ale także stawia na edukację społeczną w zakresie efektywnych praktyk prewencyjnych.

Dziedzina informatyki w znacznym stopniu przyczynia się do rozwiązania problemów związanych z pandemią. Mając do dyspozycji technologię, jesteśmy w stanie opracować zaawansowane algorytmy symulacyjne, które umożliwiają modelowanie złożonych interakcji społecznych i ruchu ludzi w środowisku biurowym. Komputery stają się potężnym narzędziem do analizy danych, identyfikowania wzorców i prognozowania potencjalnych scenariuszy rozprzestrzeniania się infekcji.

Symulacje komputerowe pozwalają nam przewidywać, jak różne warunki środowiskowe i społeczne wpływają na tempo i zasięg rozprzestrzeniania się wirusa. Ponadto, algorytmy sztucznej inteligencji mogą być używane do analizy zachowań społecznych, co pozwala na lepsze zrozumienie, jak ludzie reagują na różne sytuacje i jakie czynniki wpływają na przestrzeganie restrykcji.

Nasza praca w obszarze informatyki nie tylko skupia się na technicznej strukturze aplikacji, ale również na zastosowaniu narzędzi informatycznych w celu zwiększenia świadomości społecznej. Komputery służą jako platforma, na której możemy nie tylko symulować scenariusze, ale także efektywnie komunikować się z użytkownikami spoza środowiska medycznego, edukując ich na temat istoty zachowania się w sposób, który zmniejsza ryzyko zakażenia.

W kolejnych rozdziałach przedstawione zostaną dokładne metody i technologie, jakie wykorzystano do implementacji aplikacji, oraz skoncentrujemy się na roli informatyki w

rozwiązaniu współczesnych wyzwań zdrowotnych.

Rozdział drugi skupiony jest na przeglądzie istniejących modeli symulacji zarażeń, opartym na dogłębnej analizie dostępnej literatury. Ma na celu zidentyfikować różne podejścia i metody, które zostały wykorzystane w modelowaniu rozprzestrzeniania się infekcji.

Rozdział trzeci posłuży do przedstawienia wymagań projektowych i narzędzi, które posłużą do ich realizacji. Analiza potrzeb funkcjonalnych i technicznych pozwoli na wybór odpowiednich technologii i narzędzi programistycznych

W kolejnych dwóch rozdziałach przedstawiono odpowiednio specyfikacja zewnętrzną i wewnętrzną aplikacji. Zdefiniowany zostanie interfejs użytkownika, funkcjonalności dostępne dla użytkowników końcowych oraz scenariusze użycia. Następnie opis architektury, struktury kodu i wszystkich kluczowych elementów wewnętrznych.

Szósty rozdział poświęcony został procesom weryfikacji i walidacji stworzonej aplikacji. Opisuje wykorzystane scenariusze testowe.

Ostatni rozdział to podsumowanie całej pracy, uwzględniające wnioski powstałe z realizacji projektu oraz ewentualne kierunki dalszych rozwoju.



## Rozdział 2

# Modele symulacji rozprzestrzeniania się zarażeń

Aby skutecznie symulować rozprzestrzenianie się zarażeń, konieczne jest w pierwszej kolejności zrozumienie mechanizmów, które kierują postępującą zarazą. Początek naszej pracy powinien poprzedzić dogłębne zbadanie natury patogenu, jego zdolności i ograniczeń wynikających z procesów selekcji naturalnej. Wirus, aby przetrwać, musi zdolnością zarażania przewyższać zdolność zabijania, co sprowadza się do utrzymania współczynnika rozprzestrzeniania większego niż 1. Dodatkowo, uwzględnienie okresu inkubacji jest kluczowe, ponieważ wirus potrzebuje czasu na rozmnożenie się w organizmie nosiciela.

Jednakże, natura patogenu to tylko jeden z elementów, na które należy zwrócić uwagę w kontekście symulacji. Równie istotnym aspektem jest człowiek jako ofiara. Analiza funkcjonowania współczesnego społeczeństwa pomoże nam określić skalę, na jaką może rozprzestrzeniać się zaraza. Zrozumienie tego kontekstu umożliwi nam lepsze odzwierciedlenie rzeczywistości w modelowaniu.

Zbraną wiedzę należy następnie przełożyć na język matematyki i modelować ją komputerowo. W tym procesie istotne jest zidentyfikowanie obszarów, które mogą być uproszczone, oraz tych, które wymagają szczegółowego odwzorowania, aby osiągnąć postawione cele symulacji. W ten sposób, połączenie wiedzy o patogenie i społeczeństwie, przełożone na modele matematyczne, pozwoli nam skutecznie symulować i analizować procesy rozprzestrzeniania się zarażeń.

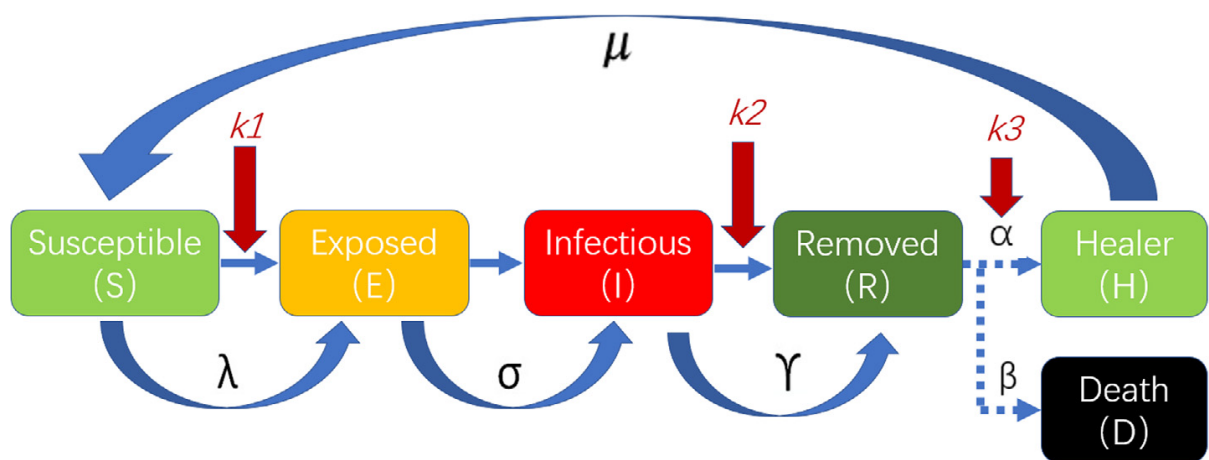
Mimo wcześniejszych prób i starań badaczy nad zjawiskiem rozprzestrzeniania się patogenów, znaczące postępy i wzmożone zainteresowanie tematem pojawiły się dopiero w latach 20. ubiegłego wieku. Świat po I wojnie światowej stanął przed pandemią grypy hiszpanki, która zarażając 1/3 ówczesnej populacji i powodując więcej ofiar niż dopiero co zakończony globalny konflikt zbrojny, spowodowała pilną potrzebę zrozumienia i kontrolowania takich masowych zjawisk. W okresie tym, w odpowiedzi na potrzebę zrozumienia dynamiki pandemii grypy hiszpanki, powstał jeden z pierwszych matematycznych modeli symulacyjnych dotyczących rozprzestrzeniania się chorób zakaźnych, znany jako model

SIR (podatni-zainfekowani-ozdrowieńcy). Model ten, opracowany w tamtych latach, stał się punktem wyjścia dla wielu kolejnych prac nad matematycznym modelowaniem epidemii, ukazując potencjał tego podejścia do zrozumienia i przewidywania rozprzestrzeniania się patogenów w społeczeństwie.

## 2.1 Modele bazujące na SIR

We współczesnych badaniach często rozwija się model SIR tak aby mógł lepiej dokładniej odzwierciedlać rozprzestrzenianie się choroby. Takimi modyfikacjami najczęściej są dalsze podzielenie populacji na grupy czy dodanie dodatkowych czynników wpływających na zarazę. Jednym z takich modeli jest *K-SEIR* opisany w artykule „*K-SEIR-Sim: A simple customized software for simulating the spread of infectious diseases.*” [2]

We wspomnianym artykule zaproponowany model rozszerza oryginalny SIR o dodatkową grupę *E* - *Exposed* (*narażeni*) oraz dodaje czynnik *K*, który określa działania przeciwdziałające zarazie podejmowane przez ludzi. Na podstawie modelu, dodatkowych parametrów oraz danych epidemiologicznych Covid-19 z miasta Wuhan zostały opracowane równania do matematycznego modelowania postępu rozprzestrzeniania się choroby, które autorzy przedstawili w tabeli.



Rysunek 2.1: Schemat działania modelu *K-SEIR*

Teoretyczny model K-SEIR został przekształcony w prosty oprogramowanie, zaimplementowane w języku PYTHON, przy użyciu inżynierii oprogramowania. W szczególności, ukończono zadania związane z projektowaniem interfejsu graficznego użytkownika, logiką sterowania, logiką operacji, kontrolą precyzji, kontrolą prędkości, wizualizacją danych, importem i eksportem danych, dopasowaniem parametrów, wyświetlaniem kluczowych danych oraz innymi konkretnymi treściami.

Tabela 2.1: Opis modelu epidemiologicznego  $K$ -SEIR.

Populacja	Równanie	Parametry
Podatni (S)	$\frac{ds}{dt} = -\frac{\lambda si}{N} + \mu h$	$\lambda$ : średnia dzienna ilość zarażeń $s$ : liczba populacji (S) w czasie $t$ $i$ : liczba populacji (I) w czasie $t$ $\mu$ : średnia dzienna ilość ponownych zarażeń $h$ : liczba populacji (H) w czasie $t$ $N$ : liczba całkowitej populacji w danym regionie
Narażeni (E)	$\frac{de}{dt} = \frac{\lambda si}{N} - \sigma e$	$\sigma$ : wskaźnik zachorowań na dzień $e$ : liczba populacji (E) w czasie $t$
Zarażeni (I)	$\frac{di}{dt} = \sigma e - \gamma i$	$\gamma$ : średni dzienny współczynnik zmniejszania grupy zarażonych pacjentów
Usunięci (R)	$\frac{dr}{dt} = \gamma i$	Suma wyleczonych i zmarłych
Ozdrowieńcy (H)	$h = \alpha r$	$r$ : liczba populacji (R) w czasie $t$ $\alpha$ : średnia dzienna współczynnik zdrowienia
Zmarli (D)	$d = \beta r$	$\beta$ : średnia dzienny współczynnik śmiertelności
	$\alpha + \beta = \gamma$	
	$s_0 + e_0 + i_0 + r_0 = N$ (dla $t = 0$ )	0: czas $t = 0$
	$\lambda_k = (1 - k_1)\lambda$	$K$ : współczynnik interwencji ludzkiej
	$\gamma_k = k_2\gamma$ $\alpha_k = k_3\alpha$	$k_1$ : miara izolacji fizycznej, współczynnik $\lambda$ $k_2$ : zdolność przyjęcia do szpitala, współczynnik $\gamma$ $k_3$ : zdolność leczenia, współczynnik $\alpha$

## 2.2 Modele agentowe

Modele agentowe stanowią komputerowe symulacje, w których agenci, reprezentujący różne jednostki, mogą wejść w interakcję między sobą. Agentami mogą być jednostki takie jak osoby, organizacje, a nawet obszary geograficzne, takie jak województwa czy kraje. Wzajemne oddziaływania między agentami są określone przez zdefiniowane zasady progra-

mowe. Każdy z agentów podejmuje decyzje indywidualnie, co może obejmować zarówno proste wybory, na przykład decyzję o kierunku ruchu, jak i bardziej złożone decyzje, takie jak znalezienie konkretnego innego agenta lub sekwencję zdarzeń. W kontekście symulacji zarażeń, tego typu podejście pozwala na realistyczne uwzględnienie nieprzewidywalnych decyzji, jakie może podjąć jednostka.

Jednym z artykułów naukowych, poruszających temat modelowania agentowego pod tytułem: „*An open-data-driven agent-based model to simulate infectious disease outbreaks*” [1] dostarczy nam cennych informacji na temat implementacji takich rozwiązań.

W artykule możemy przeczytać szczegółowe informacje na temat tego co badacze wzięli po uwagę podczas tworzenia swojego modelu. Był on oparty o dane populacji, umiejscowienia szkół i miejsc pracy a także danych dotyczących szczepień. Rzeczywiste dane są używane do określenia struktury wiekowej i płciowej naszych populacji, wraz z właściwym rozkładem wielkości gospodarstw domowych oraz innymi cechami takimi jak wiek dzieci. Następnie dane opisujące lokalizację szkół oraz miejsc pracy dają możliwość wiernego odwzorowania interakcji pomiędzy ludźmi. Na koniec aby ocenić podatność populacji na rozprzestrzenianie się choroby uwzględniono ilość szczepień (badanie opierało się na badaniu epidemii odry). Następnym krokiem badaczy było wybranie testowanego obszaru i podzielenie go na mniejsze jednostki, w których przebywający ludzie są uznawani za mający kontakt ze sobą. Później należało rozmieścić agentów w odpowiednich domach według danych o populacji. Na koniec należało uwzględnić dodatkowe czynniki między innymi transport. Ostatnim krokiem było matematyczne opisanie szans na zarażeniem, w tym celu autorzy wykorzystali równanie:

$$R_0 = cpd$$

gdzie:

$R_0$  - prawdopodobieństwo infekcji.

$c$  - liczba kontaktów na jednostkę czasu.

$p$  - szansa na zarażenie podczas kontaktu.

$d$  - długość trwania infekcji

Poprzez przekształcenie równania otrzymano:

$$p = \frac{R_0}{cd}$$

Dodatkowo w scenariuszach testowych, w których brano pod uwagę szczepienia użyto dodatkowego równania. Jest ono oparte na odporność zbiorowej jednak zmodyfikowane aby uwzględniać skuteczność szczepionki:

$$V_c = \frac{(1 - \frac{1}{R_0})}{V_e}$$

gdzie:

$V_c$  - objęcie szczepieniami populacji

$V_e$  - skuteczność szczepionki.

Badacze przetestowali swój program na wielu miastach w Irlandii, ze względu na dostępne dane co do ich struktury oraz populacji. Dodatkowo model został porównany z rzeczywistymi danymi dotyczącymi epidemii odry w Schull, Irlandia w 2012 roku. Ze względu na losowość w modelu efekty każdej z symulacji było nieco inne.

cyt. „ (...) Średnia liczba zainfekowanych agentów w różnych próbach wyniosła 17, przy maksymalnej liczbie 90 zainfekowanych agentów w jednym przypadku. Dwadzieścia pięć procent prób skończyło się wybuchem, w którym więcej niż 30 agentów zostało zainfekowanych. Wyniki pokazują, że chociaż średnia dla wszystkich prób jest niższa niż liczba zainfekowanych w przypadku wybuchu w Schull, to liczba faktycznie zainfekowanych osób znajduje się w 75. centylu wyników modelu.” (...) [1]

Podsumowując, przewagą modeli agentowych nad modelami takimi jak SEIR jest uwzględnienie decyzji jednostki, przykładowo jeżeli mimo choroby osoba zdecyduje się pójść do szkoły lub pracy, ilość zarażeń wzrośnie lub odwrotnie jeżeli zdecyduje się zostać w domu, ilość zarażeń zmaleje, takie niuanse są wychwytywane dzięki skupieniu się na jednostce. Niestety takie podejście dużo gorzej radzi sobie w miarę zwiększania populacji, dlatego idealnie nadają się do symulacji epidemii w małych miastach ale nie w obrębie całego kraju lub świata.



# Rozdział 3

## Wymagania i narzędzia

- wymagania funkcjonalne i нефункционалне
- przypadki użycia (diagramy UML) – dla prac, w których mają zastosowanie
- opis narzędzi, metod eksperymentalnych, metod modelowania itp.
- metodyka pracy nad projektowaniem i implementacją – dla prac, w których ma to zastosowanie



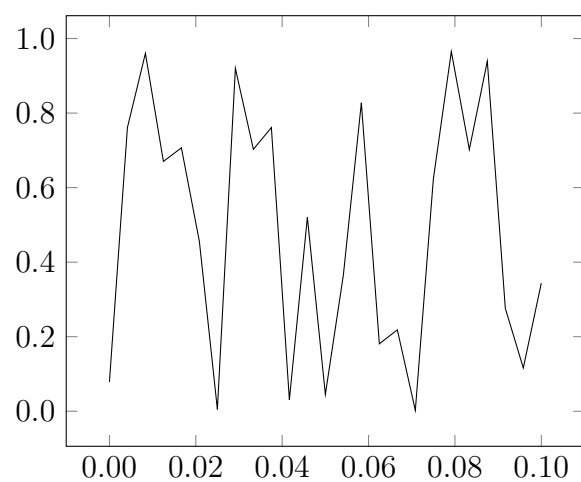


# Rozdział 4

## [Właściwy dla kierunku – np. Specyfikacja zewnętrzna]

Jeśli „Specyfikacja zewnętrzna”:

- wymagania sprzętowe i programowe
- sposób instalacji
- sposób aktywacji
- kategorie użytkowników
- sposób obsługi
- administracja systemem
- kwestie bezpieczeństwa
- przykład działania
- scenariusze korzystania z systemu (ilustrowane zrzutami z ekranu lub generowanymi dokumentami)



Rysunek 4.1: Podpis rysunku po rysunkiem.

## Rozdział 5

# [Właściwy dla kierunku – np. Specyfikacja wewnętrzna]

Jeśli „Specyfikacja wewnętrzna”:

- przedstawienie idei
- architektura systemu
- opis struktur danych (i organizacji baz danych)
- komponenty, moduły, biblioteki, przegląd ważniejszych klas (jeśli występują)
- przegląd ważniejszych algorytmów (jeśli występują)
- szczegóły implementacji wybranych fragmentów, zastosowane wzorce projektowe
- diagramy UML

Krótką wstawka kodu w linii tekstu jest możliwa, np. **int a;** (biblioteka `listings`). Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rys 5.1, a naprawdę długie fragmenty – w załączniku.

---

```
1 class test : public basic
2 {
3     public:
4         test (int a);
5         friend std::ostream operator<<(std::ostream & s,
6                                         const test & t);
7     protected:
8         int _a;
9
10 };
```

---

Rysunek 5.1: Pseudokod w `listings`.

# Rozdział 6

## Weryfikacja i walidacja

- sposób testowania w ramach pracy (np. odniesienie do modelu V)
- organizacja eksperymentów
- przypadki testowe zakres testowania (pełny/niepełny)
- wykryte i usunięte błędy
- opcjonalnie wyniki badań eksperymentalnych

Tabela 6.1: Nagłówek tabeli jest nad tabelą.

$\zeta$	metoda						
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$	
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

# Rozdział 7

## Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna ...)
- problemy napotkane w trakcie pracy





# Bibliografia

- [1] John Kelleher Elizabeth Hunter Brian Mac Namee. „An open-data-driven agent-based model to simulate infectious disease outbreaks”. W: *PLOS ONE* (2018).
- [2] Chaobao Zhang Xiaona Wei Xiangqi Li Hongzhi Wanga Zhiying Miao. „K-SEIR-Sim: A simple customized software for simulating the spread of infectious diseases”. W: *Computational and Structural Biotechnology Journal* 19 (2021), s. 1966–1975.



# Dodatki



# Spis skrótów i symboli

SIR Suceptible, Infected, Removed (podatni-zainfekowani-ozdrowieńcy)

SEIR Suceptible, Exposed, Infected, Removed (podatni-wystawieni-zainfekowani-ozdrowieńcy)

MVC model – widok – kontroler (ang. *model-view-controller*)

$N$  liczebność zbioru danych

$\mu$  stopień przyleżności do zbioru

$\mathbb{E}$  zbiór krawędzi grafu

$\mathcal{L}$  transformata Laplace’a



# Źródła

Jeżeli w pracy konieczne jest umieszczenie długich fragmentów kodu źródłowego, należy je przenieść w to miejsce.

---

```
1 if (_nClusters < 1)
2     throw std::string ("unknown_number_of_clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You should set a maximal number of
        iteration or minimal difference — epsilon.");
5 if (_nIterations > 0 and _epsilon > 0)
6     throw std::string ("Both number of iterations and minimal
        epsilon set — you should set either number of iterations
        or minimal epsilon.");
```

---





# Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- źródła programu,
- dane testowe,
- film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
- itp.



# Spis rysunków

2.1	Schemat działania modelu <i>K-SEIR</i> . . . . .	4
4.1	Podpis rysunku po rysunkiem. . . . .	12
5.1	Pseudokod w <code>listings</code> . . . . .	14



# Spis tabel

2.1	Opis modelu epidemiologicznego $K$ - $SEIR$ . . . . .	5
6.1	Nagłówek tabeli jest nad tabelą. . . . .	16