



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Зав. кафедрой

«ПОВТиАС»

В.В. Долгов

« 09 » 06 2025 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)


Тема: «ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ
РАЗВИТИЯ ЭПИДЕМИОЛОГИЧЕСКОЙ СИТУАЦИИ»

Направление подготовки 02.03.03 Математическое обеспечение и администрирование
информационных систем

Направленность Математическое обеспечение и администрирование информационных систем

Обозначение ВКР 02.03.03.350000.000 Группа ВМО41

Обучающийся

 06.06.25
подпись, дата

Э.Ю. Волкова

Руководитель ВКР

 07.06.25
подпись, дата

доцент Т.А. Медведева


Консультанты по разделам:

Безопасность и экологичность

 07.06.25
подпись, дата

доцент О.В. Денисов

Нормоконтроль

 10.06.25
подпись, дата

О.А. Попова

Ростов-на-Дону

2025



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Зав. кафедрой

«ПОВТиАС»

В.В. Долгов

« 06 »

05

2025 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

Тема: «ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ
РАЗВИТИЯ ЭПИДЕМИОЛОГИЧЕСКОЙ СИТУАЦИИ»

Обучающийся Волкова Эмилия Юрьевна

Обозначение ВКР 02.03.03.350000.000

Группа ВМО41

Тема утверждена приказом по ДГТУ от « 28 » апреля 2025 г. № 2156-ЛС-О

Срок представления ВКР к защите « 24 » июня 2025 г.

Исходные данные для выполнения выпускной квалификационной работы:

Задание на выпускную квалификационную работу, Ален, Л.Дж.С. Введение в математическую биологию / Л.Дж.С. Ален. – Москва: Мир, 2009. – 320 с., Воронин, А.А. Математические модели распространения эпидемий / А.А. Воронин. – Москва: МГУ, 2019. – 208 с., Гудман Б. Python и анализ данных / Б. Гудман. – Москва: Диалектика, 2022. – 251 с. – ISBN 978-5-8459-2531-2.

Содержание выпускной квалификационной работы

Введение:

Представлены исторические сведения о развитии математических моделей для прогнозирования развития эпидемиологической ситуации, приведено обоснование актуальности темы.

Наименование и краткое содержание разделов:

1. Теоретический обзор. В данном разделе описывается предметная область, рассматриваются существующие математические модели для прогнозирования развития эпидемиологической ситуации. Формулируется постановка цели и задач.
2. Алгоритмическое конструирование. В данном разделе приводится описание алгоритмического конструирования разрабатываемого программного средства. Представлены схемы реализованных алгоритмов.
3. Программное конструирование. В данном разделе обосновывается выбор инструментов для разработки программного средства. Проектируются основные классы и методы, необходимые для успешной реализации программы. Приводится UML-диаграмма классов программного средства.
4. Тестирование программного средства. Приведена инструкция по использованию программного средства, а также протестирована его работоспособность для реализованных математических моделей с различными входными данными. Проанализированы результаты исследования, на их основании сделаны соответствующие выводы об использовании приведенных в данной работе моделей для прогнозирования развития эпидемиологических ситуаций для отдельных заболеваний.
5. Безопасность и экологичность при разработке программного средства. Разработан комплекс мероприятий по организации рационального освещения в рабочем помещении. Выполнен расчет системы искусственного освещения.

Заключение:

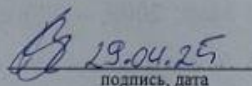
Разработанная программа предоставляет возможность проведения численных экспериментов с использованием различных параметров модели, анализа распространения инфекции и оценки эффективности мер по ее контролю. Созданное программное средство может быть использовано для научных исследований, принятия решений в сфере управления при возникновении реальной эпидемической угрозы, а также в учебных целях при изучении предметов, связанных с математическим моделированием и вычислительной математикой.

Руководитель ВКР


подпись, дата

доцент Т.А. Медведева

Задание принял к исполнению


подпись, дата

Э.Ю. Волкова

Аннотация

Отчет включает: страниц – 96; рисунков – 52; таблиц – 9; источников – 16; приложений – 3.

Цель работы – применение различных математических моделей для прогнозирования развития эпидемиологических ситуаций.

Проведен анализ предметной области и реализованы классическая модель SIR и её модификации. Разработано программное средство для математического моделирования эпидемиологических процессов. Сконструированы основные алгоритмы для реализации десктоп-приложения, включая численные методы решения СДУ. Представлены описания классов и методов в табличном и графическом виде. Проведено тестирование созданного программного средства с целью оценки его соответствия заявленным функциональным требованиям.

Abstract

The report includes: pages – 96; drawings – 52; tables – 9; sources – 16; appendices – 3.

The purpose of the work is to apply various mathematical models to predict the development of the epidemiological situation.

The analysis of the subject area is carried out and the classical SIR model and its modifications are implemented. A software tool for mathematical modeling of epidemiological processes has been developed. The basic algorithms for the implementation of a desktop application, including numerical methods for solving SDU, are constructed. Descriptions of classes and methods are presented in tabular and graphical form. The created software was tested in order to assess its compliance with the stated functional requirements.

Содержание

Введение.....	6
1 Математическое моделирование инфекционных заболеваний.....	8
1.1 Типы эпидемиологических моделей	8
1.2 Детерминистские популяционные модели.....	10
1.3 Численные методы решения СДУ	18
1.4 Существующие аналоги	21
1.5 Постановка задачи	25
1.6 Выводы по главе.....	26
2 Алгоритмическое конструирование программного средства	27
2.1 Структура программного средства	27
2.2 Загрузка данных из файла csv.....	28
2.3 Экспорт результатов моделирования.....	28
2.4 Алгоритмы численного решения СДУ	29
2.5 Выводы по главе.....	31
3 Программное конструирование.....	32
3.1 Выбор языка программирования и среды разработки	32
3.2 Используемые библиотеки.....	33
3.3 Основные классы программного средства	35
3.4 Структура входных и выходных данных	38
3.5 Выводы по главе.....	39
4 Тестирование программного средства.....	40
4.1 Описание контрольного тестирования	40
4.2 Инструкция по использованию программного средства	45
4.3 Описание загрузки данных из файла	48

					02.03.03.350000.000 ПЗ		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.	Волкова Э.Ю.	06.06.25	Программная реализация		Лист	Лист	Листов
Провер.	Медведева Т.А.	07.06.25	математического			4	96
			моделирования развития		ДГТУ Кафедра «ПОВТ и АС»		
Н.контр	Попова О.А.	10.06.25	эпидемиологической ситуации				
Утвердил	Долгов В.В.	09.06.25					

4.4 Демонстрация математических моделей	52
4.5 Анализ результатов моделирования	64
4.6 Выводы по главе	65
5 Безопасность и экологичность при разработке программного средства	66
5.1 Оценка опасности вредных факторов на рабочем месте разработчика ...	66
5.2 Расчет освещения на рабочем месте разработчика	69
5.3 Экологическая обстановка в районе предприятия	74
5.5 Выводы по главе	77
Заключение	78
Перечень используемых информационных источников	79
Приложение А Техническое задание на программное средство	81
Приложение Б Исходный код программного средства	86
Приложение В Протокол предзащиты	96

Введение

В последние десятилетия мир сталкивался с масштабными эпидемиологическими угрозами, такими как пандемия COVID-19 и вспышки лихорадки Эбола. В 2025 году Россия, как и многие другие страны, столкнулась с резким ростом заболеваемости корью – инфекцией, способной вызывать тяжелые осложнения и требующей своевременных мер контроля. Подобные ситуации ставят перед органами здравоохранения сложную задачу: необходимо не только прогнозировать развитие эпидемии, но и оценивать эффективность возможных вмешательств, таких как вакцинация, карантинные меры или ограничение массовых мероприятий. Математическое моделирование эпидемий позволяет решать эти задачи, предоставляя инструменты для анализа динамики распространения инфекции и сравнения различных стратегий борьбы с ней.

Математическое моделирование, или преобразование реальных процессов, явлений в математические соотношения, – это математическая концепция, которая имеет множество применений. Она применяется во многих областях науки таких как физика, экономика, социология и медицина. Данная работа рассматривает математическое моделирование в области медицинских наук в качестве инструмента для прогнозирования развития эпидемиологической ситуации.

Математические методы впервые были применены к изучению заболеваний в 1760 году Даниэлем Бернулли, который использовал их для оценки эффективности различных методов вакцинации от оспы [1].

В 1840 году Уильям Фарр успешно описал данные по смертности от оспы в Англии и Уэльсе за период с 1837 по 1839 год, используя нормальное распределение. Этот метод был развит Джоном Браунли, который в своей статье «Статистический подход к иммунной защите: теория эпидемий» (1906) сравнил ряды эпидемиологических данных на основе распределения Пирсона.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Хамер и Росс, используя математическое описание распространения заболеваний, смогли решить задачи по выяснению механизмов регулярного повторения эпидемии кори и установлению связи между количеством комаров и возникновением малярии. Их работы, наряду с исследованиями Росса и Хадсона, Сопера, а также Кермака и Маккендрика (1927), стали основой для дальнейших исследований в области математического моделирования эпидемий [2].

В этих работах впервые был применен «закон действующих масс», согласно которому количество вновь инфицированных прямо пропорционально произведению числа восприимчивых и инфицированных особей. Модель Кермака и Маккендрика положила начало широкому использованию детерминированных SIR-моделей («Susceptible – Infected – Recovered»), в которых с помощью систем дифференциальных или разностных уравнений описывается динамика групп восприимчивых, инфицированных и выздоровевших.

Рассматриваемые в данной работе модели могут спрогнозировать, как прогрессируют инфекционные заболевания. Модели используют базовые допущения или собранные статистические данные для нахождения параметров для различных инфекционных заболеваний, используя их для расчета эффекта различных вмешательств, таких как ввод карантина.

Программная реализация рассматриваемых математических моделей позволяет провести численный анализ различных сценариев, влияние карантинных мер и других факторов. В данной работе рассматриваются и реализуются основные детерминированные модели развития эпидемий: SI, SIR, SIRS, SEIR, SIQR, MSEIR и M-модель. Эти модели отличаются уровнем детализации и применяются в зависимости от характеристик конкретного заболевания и целей анализа.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

1 Математическое моделирование инфекционных заболеваний

В данном разделе описывается понятие математической модели. Рассматриваются конкретные модели для прогнозирования развития эпидемиологической ситуации и численные методы для решения СДУ.

1.1 Типы эпидемиологических моделей

Моделирование эпидемических процессов всегда предполагает использование упрощающих допущений относительно природы описываемого явления [3]. Ключевым требованием к любой математической модели (будь то аппроксимационная, аналитическая, стохастическая или симуляционная) является адекватное отражение существенных связей в эпидемическом процессе с исключением несущественных и случайных факторов. Чем более полно модель охватывает ключевые взаимосвязи, тем выше ее соответствие реальному развитию конкретного эпидемического процесса.

1.1.1 Стохастические модели

Стохастическая модель – это инструмент для оценки распределения вероятностей потенциальных результатов с учётом случайных изменений одного или нескольких входных параметров с течением времени.

В начале XX века ученые-эпидемиологи пришли к выводу, что для описания распространения инфекционных заболеваний в ограниченных популяциях (например, семьях) применительно к болезням, таким как корь, более целесообразным является вероятностный подход, нежели детерминированный.

В 1926 году Маккендрик сформулировал стохастический вариант модели SIR с непрерывным временем для вывода уравнений, описывающих

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

продолжительность эпидемии, применительно к гриппу и малярии. Однако работа Маккендрика не получила широкого признания.

Значительный вклад в развитие моделей с вероятностным описанием внесла работа Гринвуда 1931 года, посвященная изучению вспышки кори. Также важную роль сыграла модель Рида и Фроста, разработанная ими в период с 1928 по 1930 годы, но опубликованная лишь в 1952 году.

В моделях Гринвуда и Рида-Фроста описание количества инфицированных на каждом временном интервале привело к использованию цепи биномиальных распределений, что обусловило появление термина «цепочечно-биномиальные модели».

1.1.2 Детерминированные модели

Детерминированная модель (противоположна модели стохастической) – математическая модель, параметры и переменные которой зависят друг от друга функционально, т.е. не подчинены случайным колебаниям процесса, в связи с чем характер системы в любое время полностью определяется первоначально выбранными условиями.

При работе с большими группами населения, как в случае с туберкулезом, часто используются детерминированные или компартментальные математические модели. В детерминированной модели люди в популяции распределяются по разным подгруппам или компартментам, каждый из которых представляет собой определённую стадию эпидемии.

Скорости перехода из одного класса в другой математически выражаются как производные, поэтому модель формулируется с помощью дифференциальных уравнений [4]. При построении таких моделей необходимо предполагать, что численность населения в группе дифференцируема по времени и что процесс эпидемии является детерминированным. Другими

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

словами, изменения численности населения в группе можно рассчитать, используя только историю, которая использовалась для разработки модели.

1.2 Детерминистские популяционные модели

В популяционных моделях население представляется как совокупность групп. Формирование этих групп обусловлено различным статусом индивидов относительно рассматриваемого заболевания (например, восприимчивые к заболеванию, инфицированные и выздоровевшие) [5].

Важно отметить, что внутри каждой группы индивиды считаются идентичными по своим характеристикам. Изменения численности групп во времени обусловлены рядом следующих процессов:

- переходы индивидов из одной группы в другую вследствие инфицирования, развития заболевания, выявления и лечения заболевших индивидов;
- пополнение групп за счёт иммиграции и рождения индивидов;
- убыль в результате естественной смертности индивидуумов, гибели из-за болезни и эмиграции в другие регионы.

Среди моделей, используемых для описания динамики распространения инфекционных заболеваний в популяциях, наиболее распространены модели SIR. В этих моделях население подразделяется на три группы:

- S (Susceptible) – восприимчивые к заболеванию;
- I (Infected) – инфицированные;
- R (Recovered/Removed) – переболевшие или удаленные из популяции.

Передача инфекции происходит от инфицированных к восприимчивым. Индивиды, переболевшие болезнью, приобретают иммунитет и не могут быть инфицированы повторно.

Математически модели SIR описываются системами дифференциальных уравнений (для непрерывного времени) или разностными уравнениями (для

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

дискретного времени). Эти уравнения отражают изменение численности каждой группы с течением времени [6].

1.2.1 Классическая SIR-модель

SIR-модель – одна из самых базовых и популярных моделей. Она делит популяцию на три группы:

- S (Susceptible) - восприимчивые к заболеванию;
- I (Infected) - инфицированные;
- R (Recovered/Removed) - переболевшие или удалённые из популяции.

Схема модели SIR показана рисунке 1.1.

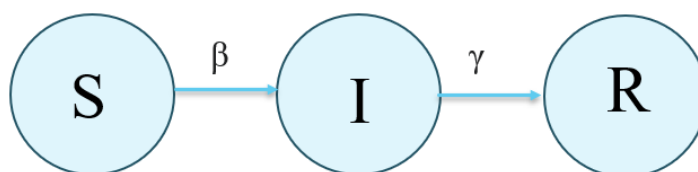


Рисунок 1.1 – SIR-модель

Данная модель предложена в 1927 году Уильямом Огильви Кермаком и Эндерсоном Греем МакКендриком. Она стала классическим фундаментом математической эпидемиологии, на основе которой разработаны и предложены модифицированные модели, рассматривающие дополнительные группы популяции [7].

Модель описывает изменение численности групп во времени с помощью системы дифференциальных уравнений. Например, стандартной SIR-модели соответствует система (1.1).

$$\begin{cases} \frac{dS(t)}{dt} = -\beta * S(t) * I(t), \\ \frac{dI(t)}{dt} = \beta * S(t) * I(t) - \gamma * I(t), \\ \frac{dR(t)}{dt} = \gamma * I(t), \end{cases} \quad (1.1)$$

где β – скорость передачи инфекции;
 γ – скорость выздоровления;
 $S(t), I(t), R(t)$ – неизвестные функции, зависящие от времени.

1.2.2 SI-модель

SI-модель является упрощенной версией классической SIR-модели, где отсутствует стадия выздоровления, то есть все, кто заражается, остаются инфицированными. Схема данной модели показана на рисунке 1.2.

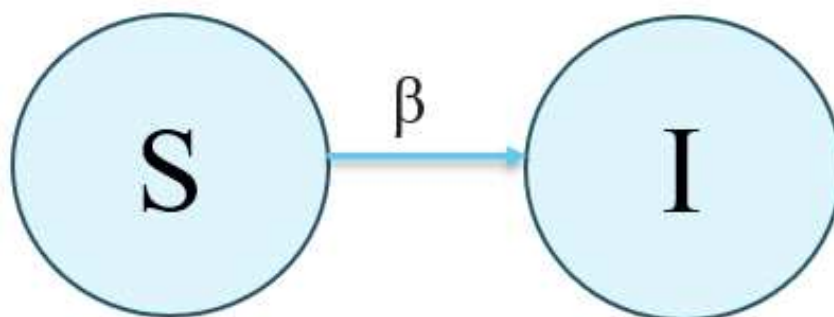


Рисунок 1.2 – SI-модель

Появилась данная модель в начале XX века, основанная на работах У. Х. Хаммерса (1920) и Нильса Кристиана Бергера (1932), как часть общего подхода к эпидемиологическому моделированию [8]. SI-модель может использоваться для описания хронических инфекций, таких как ВИЧ. Данной модели соответствует система (1.2).

$$\begin{cases} \frac{dS(t)}{dt} = -\beta * S(t) * I(t), \\ \frac{dI(t)}{dt} = \beta * S(t) * I(t), \end{cases} \quad (1.2)$$

где β – скорость передачи инфекции;
 γ – скорость выздоровления;
 $S(t), I(t)$ – неизвестные функции, зависящие от времени.

1.2.3 SIRS-модель

SIRS-модель разработана как расширение SIR-модели для инфекций, при которых иммунитет не является постоянным (например, грипп). Схема данной модели показана на рисунке 1.3.

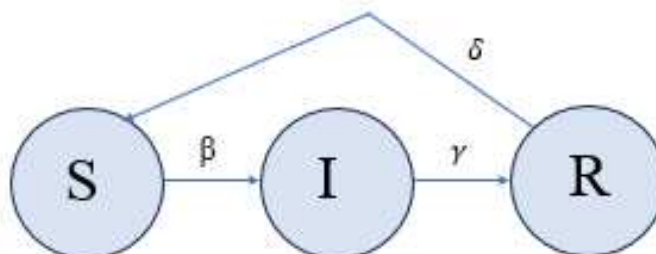


Рисунок 1.3 – SIRS-модель

SIRS-модель применялась активно в 1950–1970-х годах. В этой модели индивиды, выздоровевшие и перешедшие в группу R , могут потерять иммунитет и снова стать восприимчивыми. Данной модели соответствует система (1.3).

$$\begin{cases} \frac{dS(t)}{dt} = -\beta * S(t) * I(t) + \delta * R(t), \\ \frac{dI(t)}{dt} = \beta * S(t) * I(t) - \gamma * I(t), \\ \frac{dR(t)}{dt} = \gamma * I(t) - \delta * R(t), \end{cases} \quad (1.3)$$

где β – скорость передачи инфекции;

γ – скорость выздоровления;

δ – скорость потери иммунитета;

$S(t), I(t), R(t)$ – неизвестные функции, зависящие от времени.

1.2.4 SIQR-модель

SIQR-модель разработана в конце XX – начале XXI века на фоне необходимости учитывать карантинные меры в условиях эпидемий,

включающая дополнительное состояние [9]. Группа населения, находящееся на карантине обозначается символом Q (Quarantined). Схема данной модели представлена на рисунке 1.4.

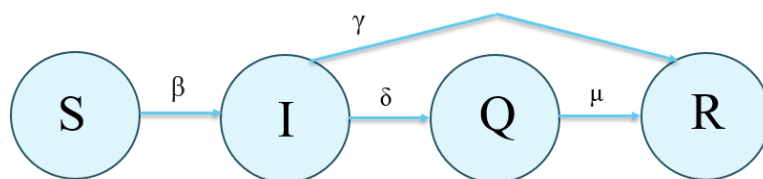


Рисунок 1.4 – SIQR-модель

Наиболее эффективна при моделировании таких инфекций как COVID-19, SARS, туберкулез, холера и другие заболевания, при которых карантин влияет на распространение. Данной модели соответствует система (1.4).

$$\begin{cases} \frac{dS(t)}{dt} = -\beta * S(t) * I(t), \\ \frac{dI(t)}{dt} = \beta * S(t) * I(t) - \gamma * I(t) - \delta * I(t), \\ \frac{dQ(t)}{dt} = \delta * I(t) - \mu * Q(t), \\ \frac{dR(t)}{dt} = \gamma * I(t) + \mu * Q(t), \end{cases} \quad (1.4)$$

где β – скорость передачи инфекции;

γ – скорость выздоровления;

δ – скорость изоляции инфицированных людей (помещения на карантин);

μ – скорость выздоровления людей, находящихся на карантине;

$S(t), I(t), R(t), Q(t)$ – неизвестные функции, зависящие от времени.

1.2.5 SEIR-модель

SEIR-модель включает дополнительное состояние – E (Exposed), или «латентную» фазу, когда человек инфицирован, но еще не заразен. Схема данной модели представлена на рисунке 1.5.

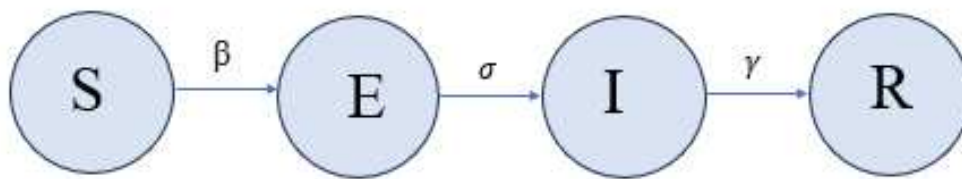


Рисунок 1.5 – SEIR-модель

Модель SEIR полезна для описания инфекций с инкубационным периодом, таких как COVID-19. Предложена в 1930-х–1950-х годах как усовершенствование модели SIR. Активно применялась во второй половине XX века для анализа заболеваний с инкубационным периодом [10]. Данной модели соответствует система (1.5).

$$\begin{cases} \frac{dS(t)}{dt} = -\beta * S(t) * I(t), \\ \frac{dE(t)}{dt} = \beta * S(t) * I(t) - \sigma * E(t), \\ \frac{dI(t)}{dt} = \sigma * E(t) - \gamma * I(t), \\ \frac{dR(t)}{dt} = \gamma * I(t), \end{cases} \quad (1.5)$$

где β – скорость передачи инфекции;

γ – скорость выздоровления;

σ – обратная величина инкубационного периода;

$S(t), I(t), R(t), E(t)$ – неизвестные функция, зависящие от времени.

1.2.6 MSEIR-модель

Модель MSEIR является расширением SEIR-модели, в которой добавляется группа М (Maternal immunity), представляющая новорожденных с пассивным иммунитетом, получивших антитела от матери [10]. Схема данной модели показана на рисунке 1.6.

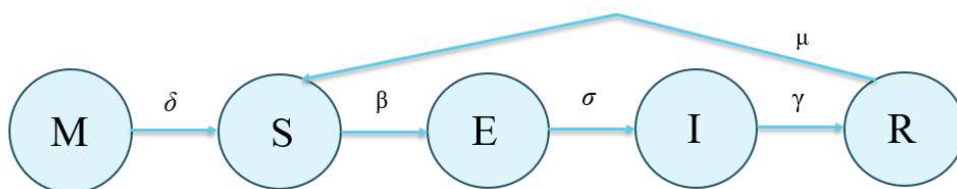


Рисунок 1.6 – MSEIR-модель

MSEIR-модель наиболее эффективна при моделировании таких инфекций как корь, краснуха, коклюш и другие детские заболевания, при которых материнский иммунитет играет важную роль в первые месяцы жизни младенца. Данной модели соответствует система (1.6).

$$\left\{ \begin{array}{l} \frac{dM(t)}{dt} = \mu * N - \delta * M(t) - \mu * M(t), \\ \frac{dS(t)}{dt} = \delta * M(t) - \beta * S(t) * I(t) - \mu * S(t), \\ \frac{dE(t)}{dt} = \beta * S(t) * I(t) - \sigma * E(t) - \mu * E(t), \\ \frac{dI(t)}{dt} = \sigma * E(t) - \gamma * I(t) - \mu * I(t), \\ \frac{dR(t)}{dt} = \gamma * I(t) - \mu * R(t), \end{array} \right. \quad (1.6)$$

где β – скорость передачи инфекции;

γ – скорость выздоровления;

σ – скорость перехода из инкубационного периода в инфекционный;

μ – естественная смертность/рождаемость;

δ – скорость потери материнского иммунитета;

$S(t), I(t), R(t), E(t), M(t)$ – неизвестные функция, зависящие от времени;

N – общее количество населения.

1.2.7 M-модель (Multi-stage)

Для более точного прогнозирования эпидемиологического процесса требуется более гибкий инструмент, учитывающий формы инфекции (легкие,

среднетяжелые, тяжелые) и варианты вмешательства [11]. Таким инструментом является М-модель (Multi-stage), показанная на рисунке 1.7.

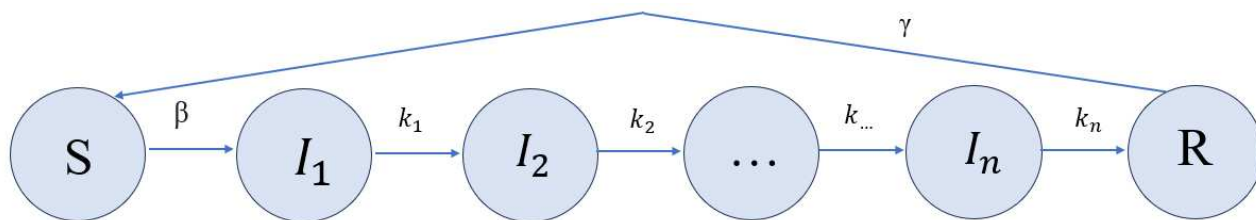


Рисунок 1.7 – М-модель (Multi-stage)

Особенностью данной модели является комплексный учет многостадийности эпидемиологического процесса и соответствующих стратегий вмешательства: карантинных ограничений и вакцинации. Данная модель в своей основе содержит модель SIRS, которая описывается системой (1.7), где группа инфицированных разделяется на несколько стадий.

$$\left\{ \begin{array}{l} \frac{dS(t)}{dt} = -\beta * S(t) * I_1(t), \\ \frac{dI_1(t)}{dt} = \beta * S(t) * I_1(t) - k_1 * I_1(t), \\ \frac{dI_2(t)}{dt} = k_1 * I_1(t) - k_2 * I_2(t), \\ \dots \\ \frac{dI_n(t)}{dt} = k_{n-1} * I_{n-1}(t) - k_n * I_n(t), \\ \frac{dR(t)}{dt} = k_n * I_n(t) - \gamma * R(t), \end{array} \right. \quad (1.7)$$

где β – скорость заражения;

k_i – скорость перехода между стадиями инфекции;

γ – скорость потери иммунитета;

$S(t), I_1(t), I_2(t), \dots, I_n(t), R(t)$ – неизвестные функции, зависящие от времени.

В отличие от традиционных подходов, принципиально важным является динамический характер модели, обеспечивающий возможность корректировки ключевых параметров на разных этапах распространения инфекции, сочетая в себе функции прогнозирования и выработки управленческих решений.

1.3 Численные методы решения СДУ

Представленные в данной работе модели являются системами дифференциальных уравнений, которые как правило не имеют аналитического решения, их необходимо решать численными методами.

Основными инструментами для численного решения являются:

- метод Эйлера: простой в применении, но обладающий ограниченной точностью.
- метод Рунге-Кутты 4-го порядка: более точный и универсальный метод, применимый к широкому спектру задач.

Данные методы позволяют получить приближенное решение, описывающее динамику эпидемии во времени, учитывая различные параметры модели.

1.3.1 Метод Эйлера

Метод Эйлера представляет собой численный метод первого порядка, предназначенный для приближенного решения уравнений и систем обыкновенных дифференциальных уравнений [12]. В его основе лежит идея аппроксимации значения функции в следующий момент времени на основе значения ее производной в текущий момент, как показано в формуле (1.8).

$$y_{n+1} = y_n + h * f(t_n, y_n), \quad (1.8)$$

где $n = 0, 1, \dots, N - 1$, $y(t_0) = y_0$ – начальное условие;

y_n – значение функции в текущий момент времени;

h – шаг временной сетки;

$t_n = n * h$ – текущий момент времени;

$f(t_n, y_n)$ – правая часть дифференциального уравнения $y' = f(t, y)$.

Метод Эйлера, успешно применяемый для решения задачи Коши ОДУ первого порядка (1.8), может быть адаптирован для решения систем

дифференциальных уравнений (СДУ). В отличие от численного решения одного ОДУ, где производная зависит только от текущего состояния системы, в СДУ присутствует зависимость от предыдущих состояний, что усложняет численное моделирование. Пусть дана система m дифференциальных уравнений первого порядка, представленная в формуле (1.9).

$$\begin{cases} \frac{dy_1}{dt} = f_1(t, y_1, y_2, \dots, y_m), \\ \frac{dy_2}{dt} = f_2(t, y_1, y_2, \dots, y_m), \\ \dots \\ \frac{dy_m}{dt} = f_m(t, y_1, y_2, \dots, y_m). \end{cases} \quad (1.9)$$

Тогда метод Эйлера для системы примет вид, представленный в формуле (1.10).

$$y_{n+1,i} = y_{n,i} + h * f_i(t_n, y_{n,1}, y_{n,2}, \dots, y_{n,m}), \quad (1.10)$$

где $i = 1, 2, \dots, m, n = 0, 1, \dots, N - 1, y_i(t_0) = y_{i,0}$ – начальное условие;

$y_{n,i}$ – значение i -й функции на n -м шаге;

h – шаг временной сетки;

f_i – правая часть i -го уравнения системы ДУ;

$t_n = n * h$ – текущий момент времени.

1.3.2 Метод Рунге-Кутта 4-го порядка

Метод Рунге-Кутта 4-го порядка – это распространенный метод численного решения дифференциальных уравнений и их систем, обеспечивающий достаточную точность вычислений [13]. В основе алгоритма лежит вычисление четырех промежуточных значений, используемых для аппроксимации решения дифференциального уравнения представленных в формулах 1.11-1.15.

$$k_1 = f(t_n, y_n), \quad (1.11)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} * k_1\right), \quad (1.12)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} * k_2\right), \quad (1.13)$$

$$k_4 = f(t_n + h, y_n + h * k_3), \quad (1.14)$$

$$y_{n+1} = y_n + \frac{h}{6} f(k_1 + 2 * k_2 + 2 * k_3 + k_4), \quad (1.15)$$

где $n = 0, 1, \dots, N - 1$, $y(t_0) = y_0$ – начальное условие;

y_n – значение функции в текущий момент времени;

h – шаг временной сетки;

$t_n = n * h$ – текущий момент времени;

$f(t_n, y_n)$ – правая часть дифференциального уравнения $y' = f(t, y)$.

Ключевой особенностью применения метода Рунге-Кутты 4-го порядка к системам дифференциальных уравнений является необходимость синхронного вычисления четырех промежуточных коэффициентов ($k_1 - k_4$) для каждого уравнения системы. Это обеспечивает:

- согласованность вычислений всех компонент системы на каждом временном шаге;
- учет взаимосвязей между переменными системы при вычислении промежуточных значений;
- сохранение баланса точности и вычислительной сложности при моделировании сложных систем.

Метод Рунге-Кутты 4-го порядка для систем уравнений вида (1.9) представлен в виде системы (1.16).

$$\begin{cases} k_{1i} = f_i(t_n, y_{n,1}, y_{n,2}, \dots, y_{n,m}), \\ k_{2i} = f_i\left(t_n + \frac{h}{2}, y_{n,1} + \frac{h}{2} * k_{11}, \dots, y_{n,m} + \frac{h}{2} * k_{1m}\right), \\ k_{3i} = f_i\left(t_n + \frac{h}{2}, y_{n,1} + \frac{h}{2} * k_{21}, \dots, y_{n,m} + \frac{h}{2} * k_{2m}\right), \\ k_{4i} = f_i(t_n + h, y_{n,1} + h * k_{31}, \dots, y_{n,m} + h * k_{3m}), \\ y_{n+1,i} = y_{n,i} + \frac{h}{6} (k_{1i} + 2 * k_{2i} + 2 * k_{3i} + k_{4i}), \end{cases} \quad (1.16)$$

где $i = 1, 2, \dots, m, n = 0, 1, \dots, N - 1, y_i(t_0) = y_{i,0}$ – начальные условия;
 f_i – правая часть i -го уравнения системы ДУ;
 $t_n = n * h$ – текущий момент времени;
 h – шаг временной сетки;
 $y_{n,i}$ – значение i -й функции на n -м шаге.

1.4 Существующие аналоги

Существует несколько инструментов для моделирования эпидемиологических ситуаций, каждый из которых обладает уникальными особенностями. Одним из таких инструментов является библиотека «Covasim» (COVID-19 Agent-based Simulator), разработанная на языке Python и доступная на GitHub. Этот инструмент был создан в 2020 году в Институте моделирования заболеваний (IDM, США) для анализа распространения COVID-19 с использованием агентного подхода. Библиотека позволяет анализировать динамику эпидемии, тестировать различные варианты мер противодействия (карантин, социальная дистанция) и оценивать эффективность данных мер. «Covasim» требует навыков программирования, так как не имеет графического интерфейса. Пример результата работы данного инструмента показан на рисунке 1.8.

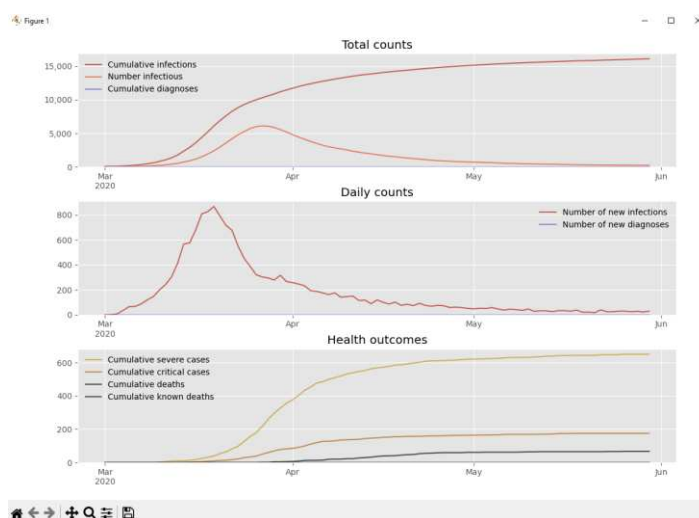


Рисунок 1.8 – Результат работы «Covasim»

Другим примером является платформа «AnyLogic», позволяющая создавать сложные модели бизнес-процессов, логистических систем, моделировать динамику эпидемий и многого другого. Рассматриваемый инструмент уникален тем, что объединяет несколько подходов:

- агентное моделирование (ABM – Agent-Based Modeling), где каждый объект (агент) ведёт себя автономно;
- дискретно-событийное моделирование (DES – Discrete Event Simulation), где система рассматривается как последовательность событий;
- системная динамика (System Dynamics), где происходит моделирование сложных систем с обратными связями (например, экономика, экология);
- гибридные модели, позволяющие комбинировать методы (например, агентная модель и системная динамика).

Разработанная в 2000 году компанией The AnyLogic Company (Россия/США), платформа обладает визуальным конструктором, но для сложных задач может потребовать написания кода на Java. Несмотря на наличие бесплатной учебной версии, основная версия платформы распространяется по проприетарной лицензии. На рисунке 1.9 показана демо-модель из области здравоохранения «Распространение ВИЧ и использование шприцов».

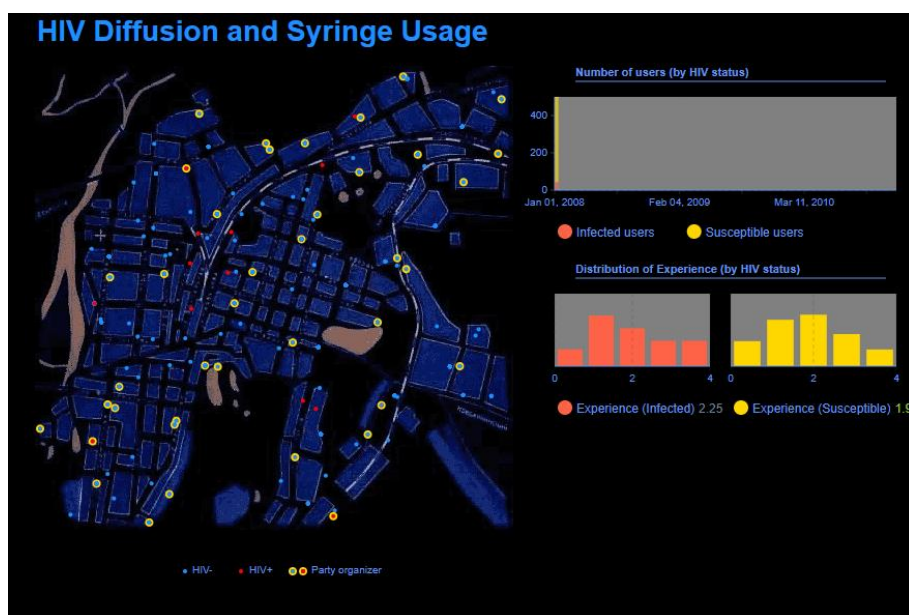


Рисунок 1.9 – Демо-модель AnyLogic

Платформа «GLEAMviz» (Global Epidemic and Mobility Model Visualizer), разработанная в 2010 году Университетом Северо-Восточного Бостона (США) и ISI Foundation (Италия), ориентирована на моделирование глобального распространения инфекций с учетом данных о перемещении людей между городами и странами. «GLEAMviz» позволяет:

- оценивать эффективность карантинных мер и ограничений на поездки;
- исследовать сценарии вакцинаций и других вмешательств;
- визуализировать динамику заражений на карте мира.

Этот инструмент доступен как бесплатное десктоп-приложение. Интерфейс «GLEAMviz» показан на рисунке 1.10.

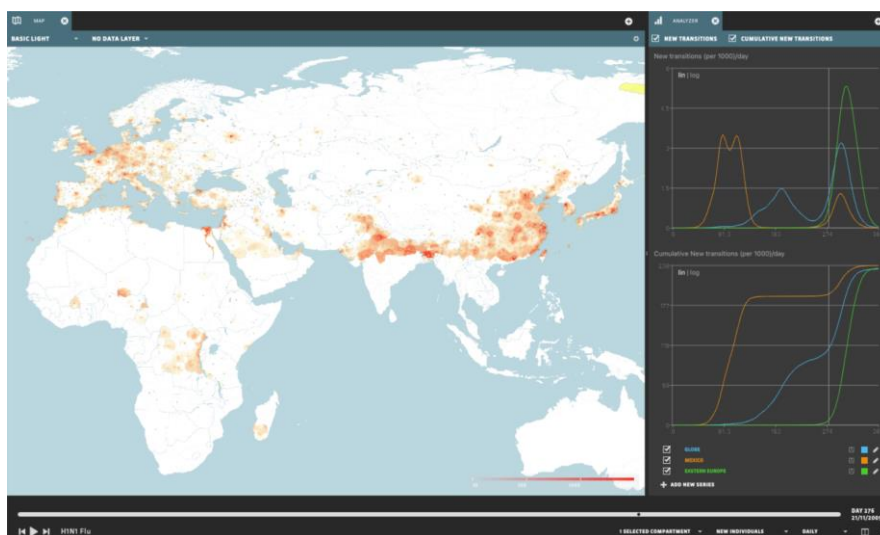


Рисунок 1.10 – Интерфейс «GLEAMviz»

«ЦЭМИ РАН» (Центральный экономико-математический институт Российской академии наук) – ведущее научное учреждение, разрабатывающее математические модели для анализа экономических и социальных процессов, включая распространение инфекционных заболеваний. Институт фокусируется на следующих направлениях:

- экономико-математические модели;
- эпидемиологические модели.

В «ЦЭМИ РАН» используют комбинацию подходов при моделировании распространения инфекций, состоящую из дифференциальных уравнений,

агентных моделей и стохастических методов. Модели института адаптированы к российским данным, основанным на статистике Росстата и Минздрава, но они являются закрытыми и их детали не публикуются, а потому используются только в госорганах.

«FRED» (Framework for Reconstructing Epidemiological Dynamics) – это открытая платформа для агентного моделирования (ABM) распространения инфекционных заболеваний, разработанная в Университете Питтсбурга (США). Платформа позволяет создавать детализированные симуляции с учётом демографии, социальных контактов и географических особенностей. «FRED» используется для:

- прогнозирования вспышек инфекций;
- оценки мер вмешательства;
- исследования социально-экономических последствий эпидемий;
- обучения эпидемиологии и общественному здравоохранению.

Доступ к данному инструменту можно получить через веб-интерфейс, облачную версию или открытый исходный код, написанный на C++. «FRED» поддерживает моделирование только на основе данных по штатам США. Пример моделирования эпидемиологической ситуации показан на рисунке 1.11.



Рисунок 1.11 – Пример моделирования «FRED»

Основываясь на рассмотренных аналогах, можно составить таблицу, состоящую из нескольких критериев. Аналоги оцениваются по наличию или отсутствию каждого критерия, что показано в таблице 1.1.

Таблица 1.1 – Оценка существующих аналогов

Критерий	Наименование аналога				
	Covasim	AnyLogic	GLEAMviz	Модели ЦЭМИ РАН	FRED
Графический интерфейс	-	+	+	-	+
Простота освоения	-	-	+	-	-
Поддержка классических моделей	±	±	-	+	+
Работа с реальными данными	+	+	+	+	+
Визуализация результатов	±	+	+	-	+
Бесплатность	+	±	+	-	+
Агентное моделирование	+	+	+	+	+
Поддержка русского языка	-	-	-	+	-
Экспорт результатов	±	+	+	-	+

1.5 Постановка задачи

Целью данной работы является разработка программной реализации классических и модифицированных моделей математической эпидемиологии с их решением с помощью численных методов.

Для достижения поставленной цели решены следующие задачи:

1. Проведен теоретический анализ существующих моделей распространения инфекционных заболеваний.
2. Сформулированы математические модели (в виде систем обыкновенных дифференциальных уравнений) для каждой рассматриваемой модели.

3. Реализовано программное решение моделей с использованием численного метода Рунге-Кутты 4-го порядка и метода Эйлера.

4. Разработан пользовательский интерфейс, обеспечивающий ввод параметров и визуализацию результатов моделирования.

5. Проведен сравнительный анализ поведения различных моделей при идентичных начальных условиях.

6. Исследованы влияние ключевых параметров (скорость заражения, инкубационный период, миграция и т.п.) на динамику эпидемии.

Данная работа представляет собой создание инструмента для анализа эпидемических ситуаций и возможности принятия решений в сфере общественного здравоохранения.

1.6 Выводы по главе

В данной главе рассмотрены типы математических моделей, проведен теоретический обзор классической SIR-модели и её модификаций: SI, SIRS, SEIR, SIQR, MSEIR и M-модель. Рассмотрены численные методы для решения СДУ. Найдены и проанализированы существующие аналоги.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

2 Алгоритмическое конструирование программного средства

В данном разделе описано алгоритмическое конструирование разработанного программного средства. Приводятся схемы используемых численных методов для решения СДУ.

2.1 Структура программного средства

В графическом интерфейсе разработанного программного средства можно выбрать одну или несколько моделей, ввести для каждой начальные данные, выбрать численный метод для решения ОДУ и экспортировать результаты моделирования в файл формата .xlsx. Описанная структура программного средства показана на рисунке 2.1.

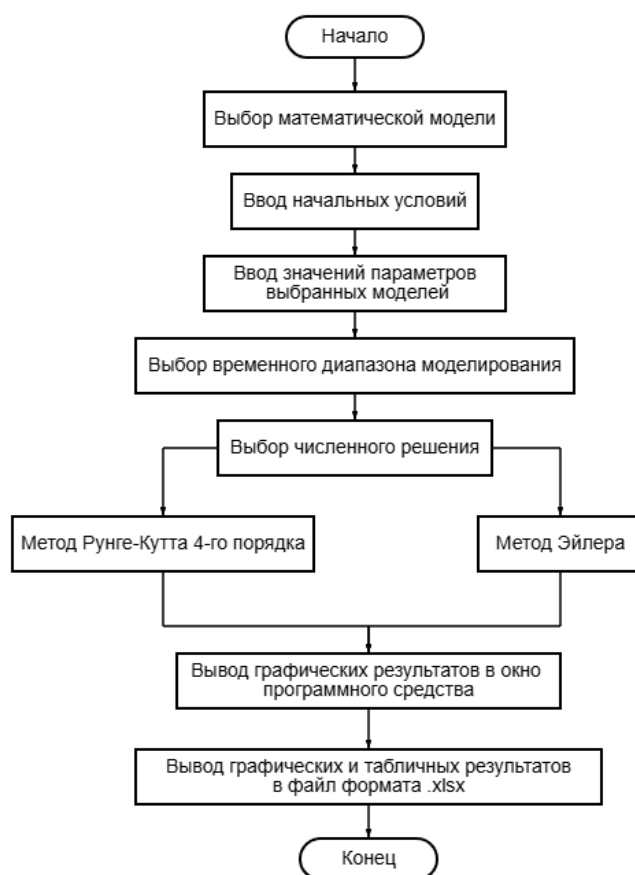


Рисунок 2.1 – Схема программного средства

2.2 Загрузка данных из файла csv

Загружаемый файл должен содержать столбцы Date, Country/Region, Confirm и Recovered. После загрузки данных приложение произведёт нормализацию количества восприимчивых, инфицированных и выздоровевших лиц, приведя сумму начальных значений к единице. На их основе программа вычисляет параметры заболевания. Данный алгоритм показан на рисунке 2.2.

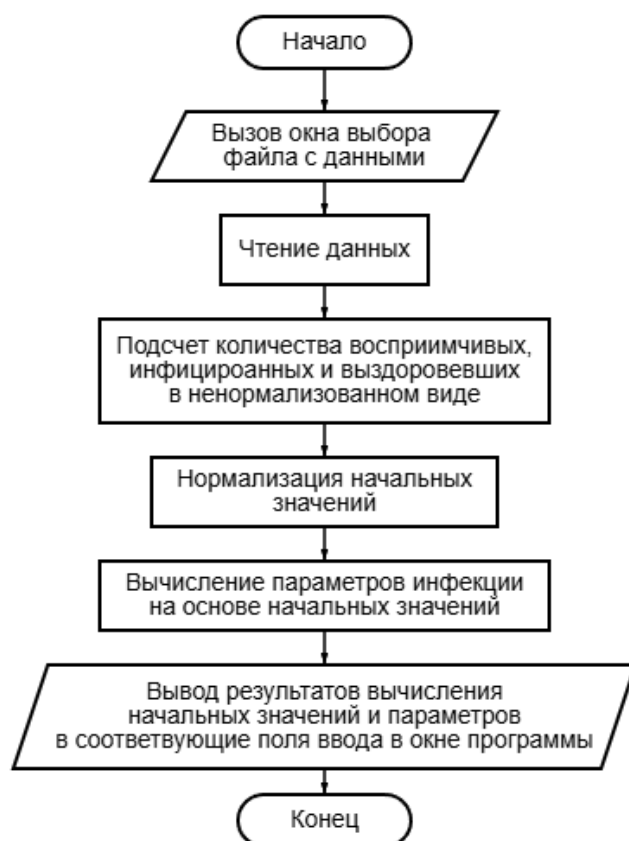


Рисунок 2.2 – Схема алгоритма загрузки данных из файла csv

2.3 Экспорт результатов моделирования

По окончании завершения моделирования в окно программного средства выводятся графики выбранных моделей. После этого пользователь может выгрузить результаты по всем моделям сразу, нажав на кнопку «Экспорт в Excel и ZIP». Алгоритм выгрузки результатов изображен на схеме 2.3.

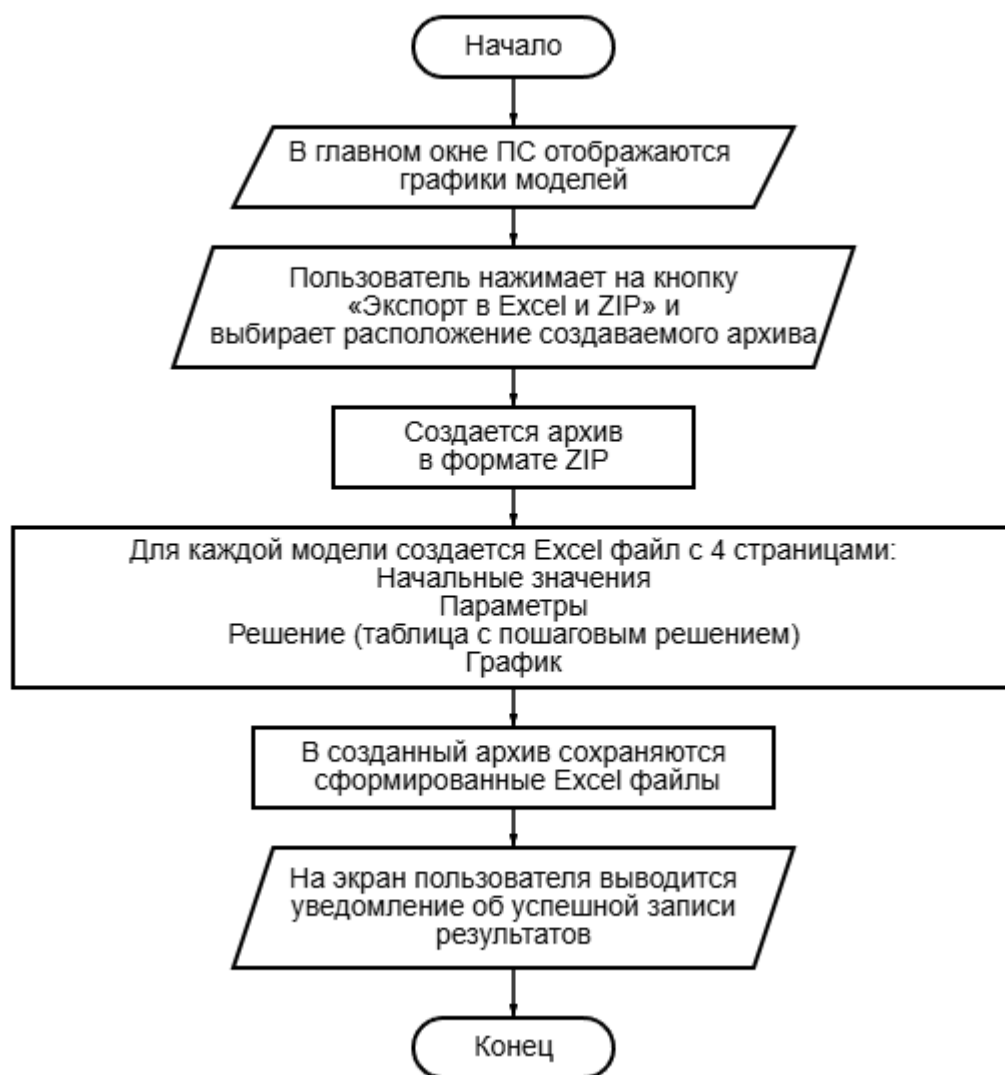


Рисунок 2.3 – Схема алгоритма экспорта результатов моделирования

2.4 Алгоритмы численного решения СДУ

Численные методы решения систем дифференциальных уравнений представлены в виде схем.

2.4.1 Метод Эйлера

Для метода Эйлера необходимы следующие входные данные: выбранная математическая модель, временной диапазон, начальные условия и параметры

модели. Структура метода Эйлера для решения представленных в данной работе СДУ приведена на рисунке 2.4.

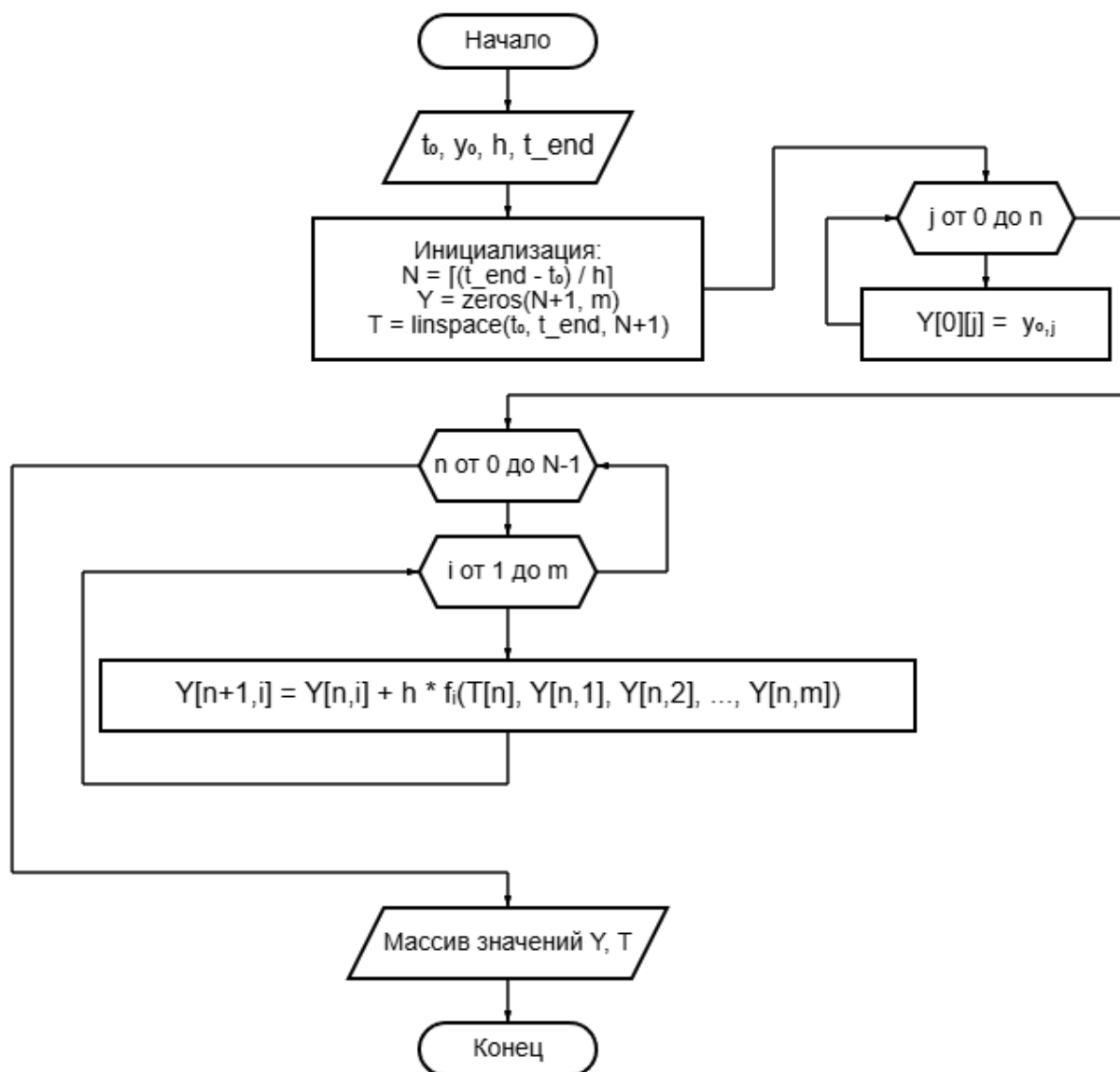


Рисунок 2.4 – Схема метода Эйлера

2.4.2 Структура метода Рунге-Кутты 4-го порядка

Для более точного численного решения СДУ с помощью метода Рунге-Кутты 4-го порядка необходимы следующие входные данные: выбранная математическая модель, начальные условия, временной диапазон и параметры

модели. Структура метода Рунге-Кутты 4-го для решения представленных в данной работе СДУ представлена на рисунке 2.5.

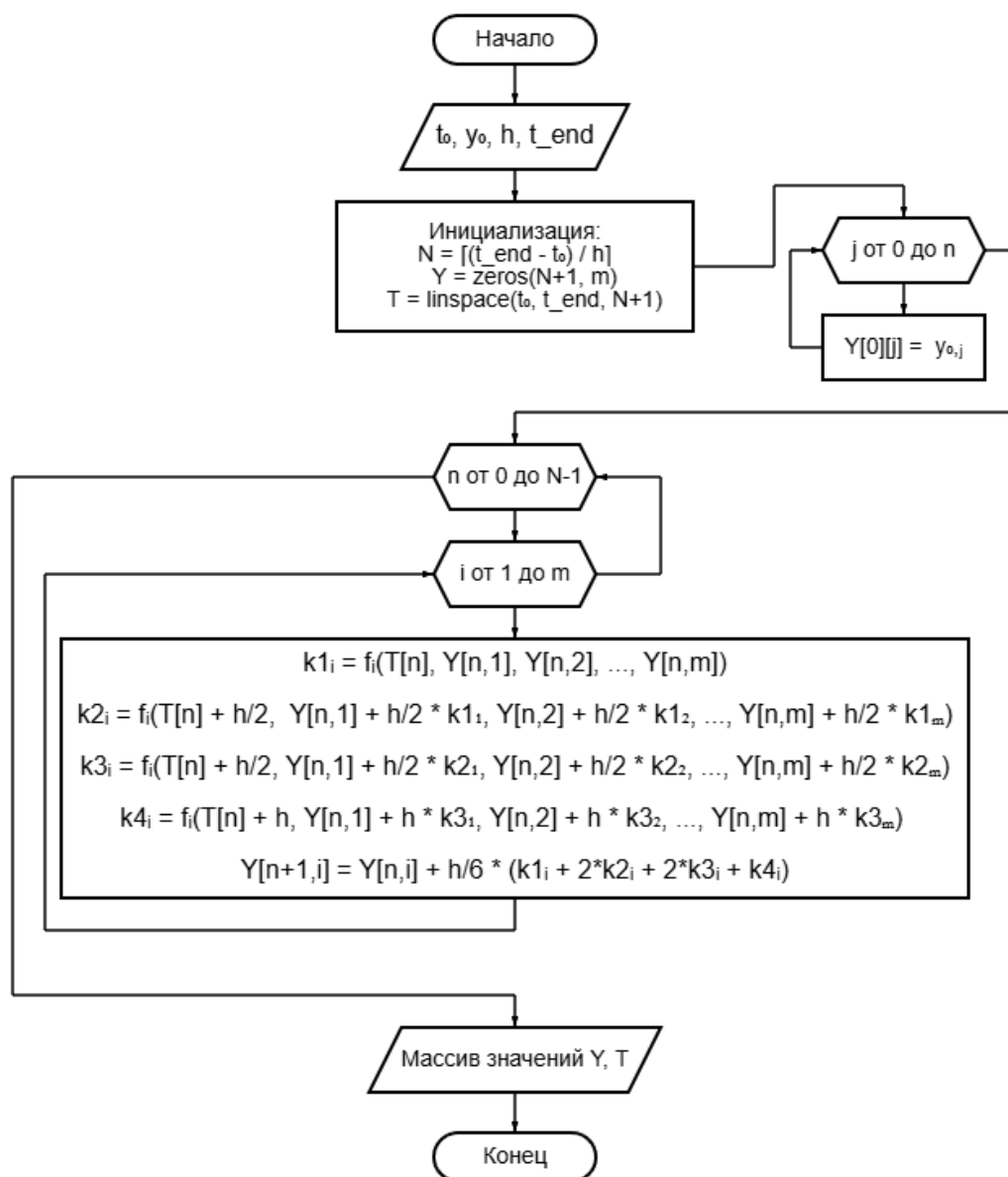


Рисунок 2.5 – Схема метода Рунге-Кутты 4-го порядка

2.5 Выводы по главе

В данной главе рассмотрена общая схема программного средства, а также схемы численных методов, применяемых для решения систем дифференциальных уравнений. Данные схемы используются для дальнейшего решения поставленной задачи.

3 Программное конструирование

В данном разделе обоснован выбор языка программирования, среды разработки и используемых в программном средстве дополнительных библиотек. Описаны основные классы и методы для выполнения поставленной задачи.

3.1 Выбор языка программирования и среды разработки

Для разработки программного средства выбран язык программирования Python версии 3.13.1. Python – это язык программирования со множеством парадигм, включая императивное, процедурное, структурное, функциональное, асинхронное и объектно-ориентированное программирование, а также метапрограммирование. Задачи обобщенного программирования решаются благодаря динамической типизации [14]. Некоторые аспекты объектно-ориентированного программирования поддерживаются через использование декораторов, в то время как более глубокая интеграция доступна с помощью дополнительных фреймворков. Реализация таких методик, как контрактное и логическое программирование, возможна с использованием библиотек или расширений. К основным архитектурным особенностям Python относятся динамическая типизация, автоматическое управление памятью, механизм обработки исключений, поддержка многопоточных вычислений с глобальной блокировкой интерпретатора (GIL) и высокоуровневые структуры данных [15]. Язык позволяет разбивать программы на модули, которые могут быть объединены в пакеты [16].

Python доступен практически на всех известных платформах, от карманных ПК до мейнфреймов. Существуют версии для Microsoft Windows, большинства дистрибутивов UNIX (включая FreeBSD и Linux), Android, Plan 9,

Mac OS и macOS, iPhone OS (iOS) версии 2.0 и выше, iPadOS, Palm OS, OS/2, Amiga, HaikuOS, AS/400, OS/390, Windows Mobile и Symbian.

В отличие от многих портируемых систем, Python обеспечивает поддержку технологий, характерных для конкретных платформ (например, Microsoft COM/DCOM). Кроме того, существует специализированная версия Python для виртуальной машины Java под названием Jython, что позволяет исполнять интерпретатор на любой системе, поддерживающей Java, при этом классы Java могут быть использованы непосредственно в Python и написаны на нем. Также есть несколько проектов, обеспечивающих интеграцию с платформой Microsoft .NET, среди которых основными являются IronPython и Python.NET. Средой разработки был выбран Visual Studio Code, т.к. включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для факторинга. Распространяется бесплатно и позиционируется как «лёгкий» редактор кода.

Для реализации программного средства выбрана среда разработки Visual Studio Code (VS Code) – бесплатный, кроссплатформенный редактор кода от Microsoft, доступный для Windows, Linux и macOS. VS Code, позиционируемый как легковесный инструмент, идеально подходит для веб- и облачной разработки. Он обладает встроенными функциями, такими как отладчик, поддержка Git, подсветка синтаксиса, IntelliSense и инструменты рефакторинга. С 2018 года доступно расширение Python с открытым исходным кодом, значительно расширяющее возможности редактирования, отладки и тестирования Python-кода.

3.2 Используемые библиотеки

Разработанное программное средство содержит несколько библиотек, позволяющих вводить даты с помощью удобного календаря, выводить на экран пользователя графики, выполнять численные операции, импортировать данные

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

из файлов в программу и экспортировать результаты вычислений в файл формата *xlsx*. Используемые в программном средстве библиотеки представлены в таблице 3.1.

Таблица 3.1 – Используемые библиотеки

Название библиотеки	Описание	Назначение
Tkinter	Событийно-ориентированная графическая библиотека на основе средств Tk (широко распространённая в мире GNU/Linux и других UNIX-подобных систем, портирована также и на Microsoft Windows). Входит в стандартную библиотеку Python.	Используется для реализации пользовательского интерфейса.
Tkcalendar	Модуль Python, предоставляющий виджеты Calendar и DateEntry для Tkinter.	Используется для удобного ввода начальной и конечной даты моделирования эпидемиологической ситуации.
Datetime	Модуль, предоставляющий классы для работы с датами и временем.	Используется для получения текущей даты с календаря компьютера.
Matplotlib	Библиотека на языке программирования Python для визуализации данных двумерной и трёхмерной графикой.	Используется для создания графических результатов моделирования и вывода их в пользовательский интерфейс.
NumPy	Библиотека с открытым исходным кодом для языка программирования Python, включающая поддержку многомерных массивов (включая матрицы) и высокоуровневых математических функций, предназначенных для работы с многомерными массивами.	Используется для создания пустых массивов и временного ряда.
pandas	Программная библиотека на языке Python для обработки и анализа данных. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.	Используется для чтения файлов формата <i>.csv</i> и корректной загрузки данных в программное средство для дальнейшей обработки.
zipfile	Данный модуль работает с ZIP-файлами, использующими расширения ZIP64. Он поддерживает расшифровку зашифрованных файлов в ZIP-архивах.	Используется для создания архива, содержащего файлы формата <i>.xlsx</i> с результатами по каждой выбранной модели.
io	Модуль предоставляет основные возможности Python для работы с различными типами операций ввода-вывода.	Используется для ввода информации в файл формата <i>xlsx</i>

3.3 Основные классы программного средства

Для реализации программного средства выбрана объектно-ориентированная парадигма. В процессе программного конструирования были выделены следующие классы: класс численных методов NumericalMethods, класс математических моделей EpidemicModels и главный класс программного средства EpidemicApp.

NumericalMethods представляет собой класс, содержащий численные методы Эйлера и Рунге-Кутты 4-го порядка для решения систем дифференциальных уравнений. Данный класс содержит методы, приведенные в таблице 3.2.

Таблица 3.2 – Методы класса NumericalMethods

Название метода	Функциональное назначение метода
euler_method	метод для нахождения решения системы дифференциальных уравнений с помощью метода Эйлера
runge_kutta_4	метод для нахождения решения системы дифференциальных уравнений с помощью метода Рунге-Кутты 4-го порядка

Класс EpidemicModels содержит методы, реализующие математические модели развития эпидемиологических ситуаций, представленные в данной работе. Содержит методы, приведенные в таблице 3.3.

Таблица 3.3 – Методы класса EpidemicModels

Название метода	Функциональное назначение метода
sir_model	метод, реализующий логику SIR-модели
si_model	метод, реализующий логику SI-модели
sirs_model	метод, реализующий логику SIRS-модели
siqr_model	метод, реализующий логику SIQR-модели
seir_model	метод, реализующий логику SEIR-модели
mseir_model	метод, реализующий логику MSEIR-модели
multi_stage_model	Метод, реализующий логику M-модели
run_si_model	метод, запускающий SI-модель на указанном диапазоне времени
run_sir_model	метод, запускающий SIR-модель на указанном диапазоне времени

Название метода	Функциональное назначение метода
run_sirs_model	метод, запускающий SIRS-модель на указанном диапазоне времени
run_seir_model	метод, запускающий SEIR-модель на указанном диапазоне времени
run_siqr_model	метод, запускающий SIQR-модель на указанном диапазоне времени
run_mseir_model	метод, запускающий MSEIR-модель на указанном диапазоне времени
run_m_model	метод, запускающий M-модель на указанном графике

Класс EpidemicApp содержит методы, реализующие логику работы всей программы, а также загрузку и данных из файла в программу и экспорт результатов в Excel файл. Методы данного класса создают главное окно программного средства, осуществляют ввод данных и запуск моделирования. Содержит методы, приведенные в таблице 3.4.

Таблица 3.4 – Методы класса EpidemicApp

Название метода	Функциональное назначение метода
create_widgets	метод создания основного интерфейса
create_models_tab	метод создания вкладки выбора моделей
create_model_params_tab	метод создания вкладки для ввода параметров и начальных условий
create_params_tab	метод, для динамического создания вкладок параметров под каждую выбранную модель
create_data_tab	метод создания вкладки для загрузки и экспорта данных
create_graphs	метод настройки графиков для отображения результатов
create_validate_func	метод для валидации вводимых данных
set_default_values	метод, устанавливающий значения по умолчанию
update_model_selection	метод, ограничивающий и проверяющие количество выбранных моделей
run_simulation	метод, запускающий прогнозирование на основе выбранных моделей
load_csv_data	метод, загружающие начальные данные из CSV файла
process_csv_data	метод, обрабатывающий загруженные данные
export_results	метод, экспортирующий результаты моделирование в Excel файл(ы)
open_model_docs	метод, создающий окно со справкой по моделям

Название метода	Функциональное назначение метода
validate_sum	метод, выполняющий проверку суммы значений, введённых в поля начальных условий

Полный листинг программы приведен в Приложении Б. UML-диаграмма классов программного средства изображена на рисунке 3.1.

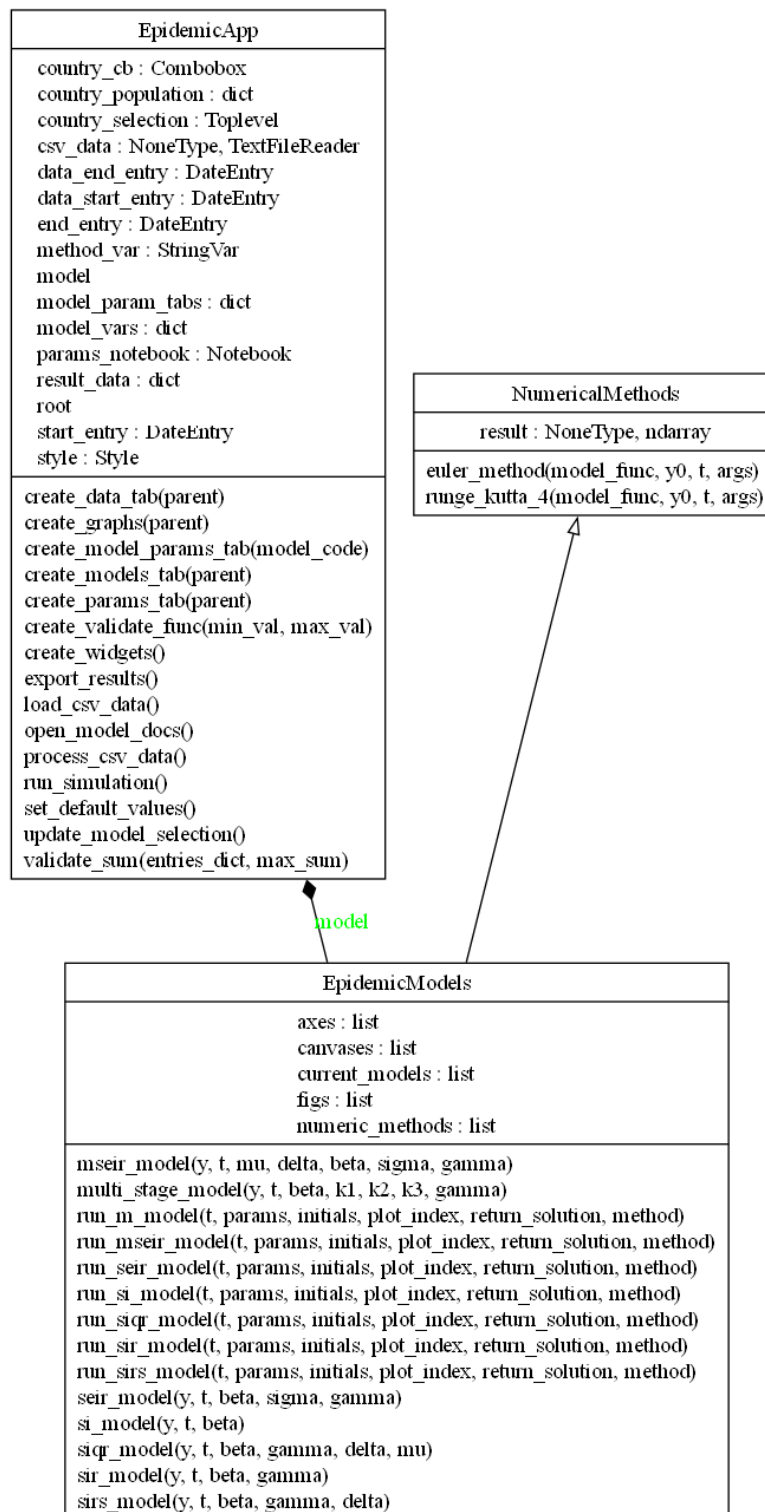


Рисунок 3.1 – UML-диаграмма классов программного средства

3.4 Структура входных и выходных данных

Программное средство получает на вход: выбранные модели и численный метод, начальные значения, параметры модели, даты моделирования. Пользователь может ввести их с помощью клавиатуры или выгрузить из файла csv, содержащий обязательные поля: Country/Region, Date, Confirmed и Recovered, как изображено на рисунке 3.2.

	A	B	C	D	E	F	G	H	I	J
1	Province/State,Country/Region,Lat,Long,Date,Confirmed,Deaths,Recovered,Active,WHO Region									
2	,Afghanistan,33.93911,67.709953,2020-01-22,0,0,0,0,Eastern Mediterranean									
3	,Albania,41.1533,20.1683,2020-01-22,0,0,0,0,Europe									
4	,Algeria,28.0339,1.6596,2020-01-22,0,0,0,0,Africa									
5	,Andorra,42.5063,1.5218,2020-01-22,0,0,0,0,Europe									
6	,Angola,-11.2027,17.8739,2020-01-22,0,0,0,0,Africa									
7	,Antigua and Barbuda,17.0608,-61.7964,2020-01-22,0,0,0,0,Americas									
8	,Argentina,-38.4161,-63.6167,2020-01-22,0,0,0,0,Americas									
9	,Armenia,40.0691,45.0382,2020-01-22,0,0,0,0,Europe									
10	Australian Capital Territory,Australia,-35.4735,149.0124,2020-01-22,0,0,0,0,Western Pacific									
11	New South Wales,Australia,-33.8688,151.2093,2020-01-22,0,0,0,0,Western Pacific									
12	Northern Territory,Australia,-12.4634,130.8456,2020-01-22,0,0,0,0,Western Pacific									
13	Queensland,Australia,-27.4698,153.0251,2020-01-22,0,0,0,0,Western Pacific									
14	South Australia,Australia,-34.9285,138.6007,2020-01-22,0,0,0,0,Western Pacific									
15	Tasmania,Australia,-42.8821,147.3272,2020-01-22,0,0,0,0,Western Pacific									

Рисунок 3.2 – Пример статических данных из файла формата csv

После окончания моделирования пользователь может выгрузить данные из программы в Excel файл, который содержит 4 листа, в каждом из которых содержится информация о модели по отдельности: начальные значения, параметры, таблица решения и график, как показано на рисунка 3.3-3.7.



Рисунок 3.3 – Названия листов в Excel файле

	A	B
1	Параметр	Значение
2	S ₀ (восприимчивые)	0.9902
3	I ₀ (инфицированные)	0.0056
4	R ₀ (выздоровевшие)	0.0042
5	E ₀ (латентные)	0.03
6	Q ₀ (изолированные)	
7	M ₀ (материнский иммунитет)	
8		
9		

Рисунок 3.4 – Лист начальных значений

	A	B	C
1	Параметр	Значение	
2	β (скорость заражения)	0.0841	
3	γ (скорость выздоровления)	0.0259	
4	δ (потеря иммунитета)		
5	σ (переход в инфекционные)	0.25	
6	μ (выход из изоляции)		
7			

Рисунок 3.5 – Лист параметров

	A	B	C	D	E	F
1	Метод решения:	Рунге-Кутта 4-го порядка				
2	Выбранная модель:	SEIR				
3						
4	Дни	S	E	I	R	
5	0	0,9902	0,03	0,0056	0,0042	
6	1	0,989453	0,024036	0,012079	0,004433	
7	2	0,988229	0,01981	0,017148	0,004814	
8	3	0,98663	0,016849	0,021209	0,005312	
9	4	0,984729	0,014809	0,024555	0,005906	
10	5	0,982577	0,013442	0,027401	0,00658	
11	6	0,980211	0,012566	0,0299	0,007323	
12	7	0,977654	0,012052	0,032167	0,008127	
13	8	0,974926	0,011803	0,034283	0,008988	
14	9	0,972036	0,011753	0,03631	0,009902	
15	10	0,968991	0,01185	0,038291	0,010868	
16	11	0,965795	0,012059	0,040261	0,011885	
17	12	0,962451	0,012353	0,042242	0,012954	
18	13	0,958957	0,012715	0,044254	0,014074	
19	14	0,955312	0,01313	0,046311	0,015246	
20	15	0,951514	0,013589	0,048424	0,016473	
21	16	0,947561	0,014084	0,0506	0,017755	
22	17	0,943449	0,01461	0,052846	0,019095	
23	18	0,939174	0,015164	0,055169	0,020493	
24	19	0,934732	0,015742	0,057572	0,021953	
25	20	0,930121	0,016344	0,060059	0,023476	

Рисунок 3.6 – Лист решения

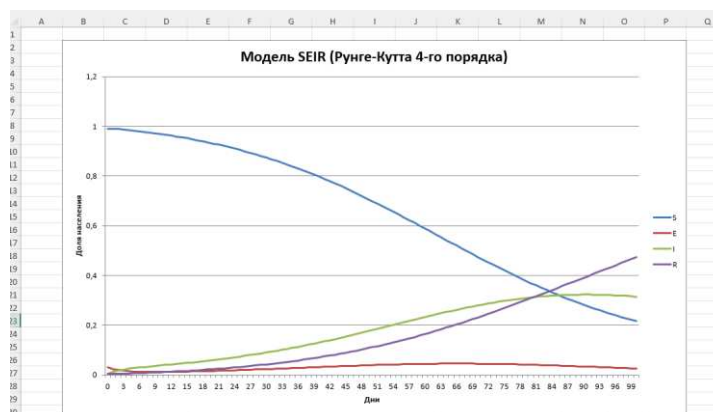


Рисунок 3.7 – Лист графика

3.5 Выводы по главе

В данном разделе обоснован выбор языка программирования и среды разработки для реализации программного средства. Описаны используемые дополнительные библиотеки. Представлены разработанные основные классы и методы программы.

4 Тестирование программного средства

В данном разделе описаны и протестированы контрольные примеры с подробным описанием входных данных и подтверждающими работу программного средства изображениями. Приводится инструкция по использованию программного средства. Проведен анализ результатов моделирования.

4.1 Описание контрольного тестирования

Целью тестирования конечного продукта является выявление и устранение дефектов, верификация соответствия функциональным требованиям, а также оценка качества программного средства и готовности к эксплуатации.

В ходе тестирования применена стратегия «черного ящика», предполагающая проверку работоспособности программного продукта без анализа его внутренней структуры или реализации. Данный подход позволил оценить функционирование программы с позиции пользователя и обнаружить любые расхождения между ожидаемым поведением и заявленными требованиями.

Таблица 4.1 демонстрирует результаты тестирования разработанного программного средства.

Таблица 4.1 – Результаты тестирования конечного продукта

№ п/п	Наименование проверки	Последовательность действий	Определение успешной проверки	Примечание
1	Выбор одной математической модели	1. Запустить программное средство 2. Выбрать одну любую математическую модель 3. Оставить начальные значения и параметры модели по умолчанию	1. Программа выводит в окно интерфейса график выбранной модели	Ошибок нет

№ п/п	Наименование проверки	Последовательность действий	Определение успешной проверки	Примечание
		4. Оставить даты начала и конца моделирования по умолчанию 5. Оставить численный метод по умолчанию (Рунге-Кутта 4-го порядка) 6. Нажать на кнопку «Запустить моделирование»		
2	Выбор двух математических моделей	Предусловие: программное средство запущено 1. Выбрать две любые математические модели 2. Оставить начальные значения и параметры моделей по умолчанию 3. Оставить даты начала и конца моделирования по умолчанию 4. Оставить численный метод по умолчанию (Рунге-Кутта 4-го порядка) 5. Нажать на кнопку «Запустить моделирование»	1. Программа выводит в окно программы графики выбранных моделей	Ошибок нет
3	Выбор трех математических моделей	Предусловие: программное средство запущено 1. Выбрать три любые математические модели 2. Оставить начальные значения и параметры моделей по умолчанию 3. Оставить даты начала и конца моделирования по умолчанию 4. Оставить численный метод по умолчанию (Рунге-Кутта 4-го порядка) 5. Нажать на кнопку «Запустить моделирование»	1. Программа выводит в окно программы графики выбранных моделей	Ошибок нет

№ п/п	Наименование проверки	Последовательность действий	Определение успешной проверки	Примечание
4	Выбор четырех математических моделей	Предусловие: программное средство запущено 1. Выбрать четыре любые математические модели 2. Оставить начальные значения и параметры моделей по умолчанию 3. Оставить даты начала и конца моделирования по умолчанию 4. Оставить численный метод по умолчанию (Рунге-Кутта 4-го порядка) 5. Нажать на кнопку «Запустить моделирование»	1. Программа выводит в окно программы графики выбранных моделей	Ошибок нет
5	Выбор более четырех математических моделей	Предусловие: программное средство запущено 1. Выбрать более четырех любых математических модели	1. При выборе пятой модели на экран выводится предупреждение о превышенном лимите выбранных моделей 2. Выбранная пятая модель не засчитывается как выбранная	Ошибок нет
6	Ввод начальных значений для любой выбранной модели	Предусловие: программное средство запущено и выбрана хотя бы одна модель 1. Перейти на вкладку «Параметры» 2. Ввести числовые значения в поля ввода начальных значений модели 3. Вернуться на вкладку «Модели» 4. Нажать на кнопку «Запустить моделирование»	1. При переходе на вкладку «Модели» программное средство проверяет корректность введенных данных. При успешной проверке пользователь не получает никаких сообщений	Ошибок нет

№ п/п	Наименование проверки	Последовательность действий	Определение успешной проверки	Примечание
			2. При запуске моделирования в окне программного средства выводятся графики выбранных моделей	
7	Ввод параметров для любой выбранной модели	<p>Предусловие: программное средство запущено и выбрана хотя бы одна модель</p> <p>1. Перейти на вкладку «Параметры» 2. Ввести числовые значения в поля ввода параметров модели 3. Вернуться на вкладку «Модели» 4. Нажать на кнопку «Запустить моделирование»</p>	<p>1. При переходе на вкладку «Модели» программное средство проверяет корректность введенных данных. При успешной проверке пользователь не получает никаких сообщений</p> <p>2. При запуске моделирования в окне программного средства выводятся графики выбранных моделей</p>	Ошибок нет
8	Ввод начальной и конечной даты моделирования	<p>Предусловие: программное средство запущено и выбрана хотя бы одна модель</p> <p>1. Перейти на вкладку «Параметры» 2. Ввести начальную и конечную дату моделирования в формате ДД.ММ.ГГГГ 3. Вернуться на вкладку «Модели»</p>	1. При запуске моделирования в окне программного средства выводятся графики выбранных моделей с выбранным временным диапазоном	Ошибок нет

№ п/п	Наименование проверки	Последовательность действий	Определение успешной проверки	Примечание
		4. Нажать на кнопку «Запустить моделирование»		
9	Загрузка данных из csv файла. Обработка файла	Предусловие: программное средство запущено 1. Перейти на вкладку «Данные» 2. Нажать на кнопку «Загрузить данные из CSV» 3. Выбрать файл с данными из открывшегося окна приложения «Проводник»	1. На экране пользователя появится окно с предложением выбрать страну и временной промежуток, за который будут выбраны данные	Ошибок нет
10	Загрузка данных из csv файла. Загрузка данных в программу	Предусловие: программное средство запущено, выбраны файл с данными, страна и временной диапазон 1. Перейти на вкладку «Модели» и выбрать от 1 до 4 моделей 2. Перейти на вкладку «Параметры» и проверить успешность расчета начальных значений, параметров для каждой модели и начальной/конечной даты моделирования 3. Вернуться на вкладку «Модели» 4. Нажать на кнопку «Запустить моделирование»	1. При запуске моделирования в окне программного средства выводятся графики выбранных моделей, расчет которых произведен на основе загруженных в программу данных	Ошибок нет
11	Экспорт результатов	Предусловие: программное средство запущено и произведено моделирование 1. Перейти на вкладку «Данные» 2. Нажать на кнопку «Экспорт в Excel и ZIP»	1. В памяти компьютера сохранен архив с Excel файлами по каждой выбранной модели	Ошибок нет

№ п/п	Наименование проверки	Последовательность действий	Определение успешной проверки	Примечание
		3. Выбрать расположение ZIP файла с результатами моделирования из открывшегося окна приложения «Проводник» 4. Открыть в приложении «Проводник» место, в котором был сохранен архив с результатами моделирования 5. Распаковать ZIP файл в любом удобном месте 6. Перейти в раздел, куда были сохранены распакованные результаты моделирования 7. Открыть любой Excel файл	2. В каждом файле создано четыре листа: «Начальные данные», «Параметры», «Решение», «График» 3. В листе «Начальные данные» записаны введенные начальные значения модели 4. В листе «Параметры» записаны введенные значения параметров модели 5. В листе «Решение» указан численный метод, а также таблица с числовыми значениями по дням моделирования 6. В листе «График» сохранен график модели	

4.2 Инструкция по использованию программного средства

Разработанное программное средство предназначено для моделирования развития эпидемиологической ситуации с помощью представленных в данной работе математических моделей: SIR, SI, SIRS, SIQR, SEIR, MSEIR и M-модель. Для начала работы с программой пользователю необходимо выбрать 1-4

модели, выбрать начальную и конечную дату моделирования, а также ввести начальные значения долей населения и значения параметров выбранных моделей. Интерфейс пользователя поддерживает вариант, при котором пользователь может загрузить в программу файл со своими данными.

В любой момент времени пользователь может нажать на кнопку «Справка по моделям», в которой каждой модели дано краткое описание, область применения (наиболее подходящие для выбранной модели инфекции), система дифференциальных уравнений и рекомендованные значения для начальных условий и параметров выбранной модели. Пример окна со справочной информацией показан на рисунке 4.1.

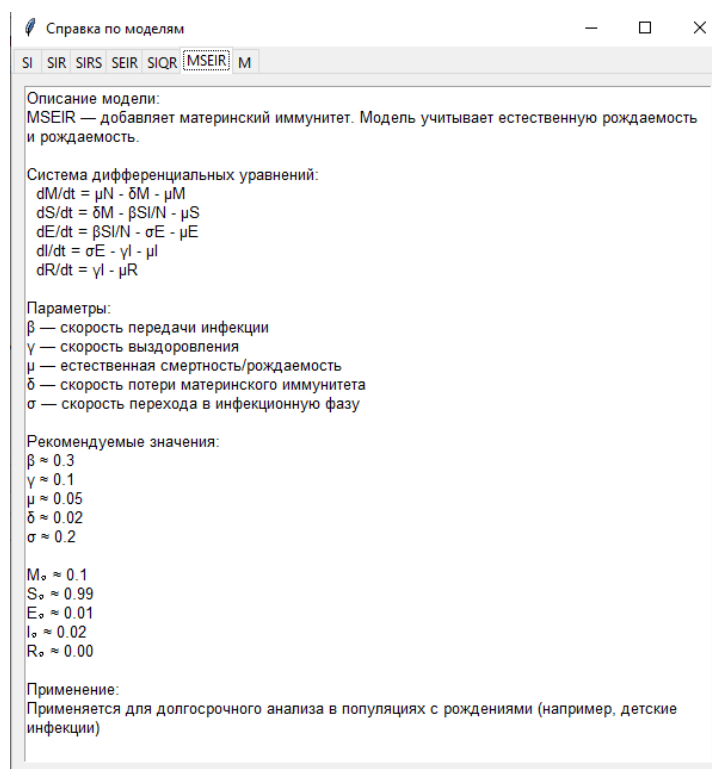


Рисунок 4.1 – Окно справочной информации

Запуск моделирования осуществляется через нажатие кнопки «Запустить моделирование». После нажатия в окне программного средства появятся графики выбранных моделей на указанном отрезке времени. Пользователь может сохранить результат моделирования в формате Excel файла, нажав на кнопку «Экспорт в файл». Создается архив, в котором сохранены по

отдельности результаты каждой выбранной модели в Excel файлах с информацией на 4-х листах:

- «Начальные условия» – лист, в котором сохранены введенные пользователем начальные значения для долей населения;
- «Параметры» – лист, содержащий введенные пользователем значения для параметров модели;
- «Решение» – лист, в котором сохранено по дням решение математической модели с помощью численного метода;
- «График» – лист, содержащий результирующий график моделирования развития эпидемиологической ситуации.

Начальное состояние интерфейса программного средства представлено на рисунке 4.2.

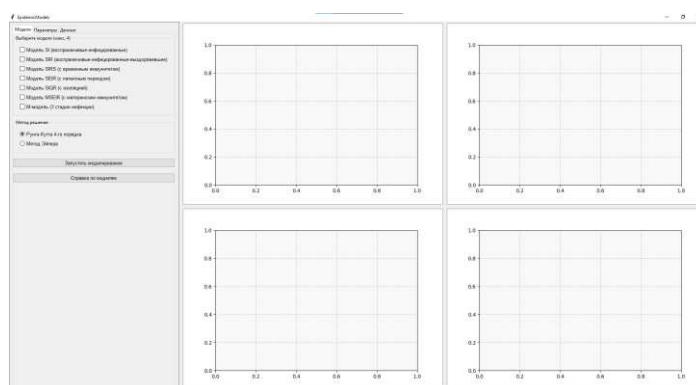


Рисунок 4.2 – Начальное состояние

Конечное состояние интерфейса программного средства показано на рисунке 4.3.

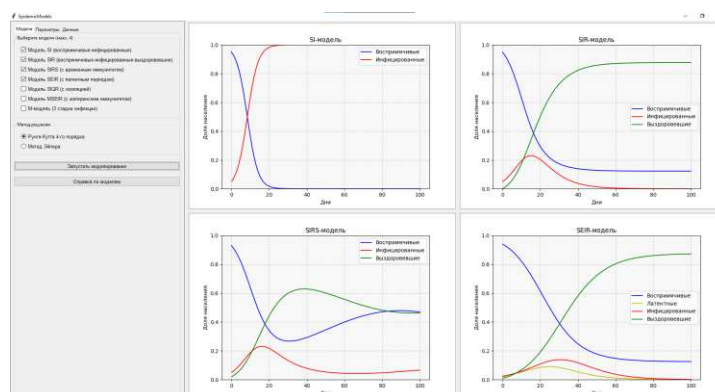


Рисунок 4.3 – Конечное состояние

4.3 Описание загрузки данных из файла

Для загрузки заранее составленных данных для моделирования необходимо перейти на вкладку «Данные» и нажать на кнопку «Загрузить данные из CSV», что показано на рисунке 4.4.

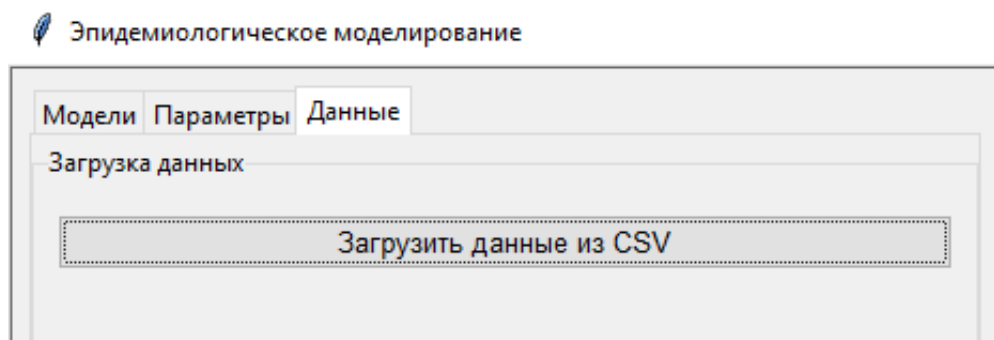


Рисунок 4.4 – Раздел загрузки данных из файла

После этого программа попытается выгрузить необходимые данные из файла: даты, количество населения, восприимчивые, инфицированные и выздоровевшие. Если данные из файла не были успешно выгружены, то на экран пользователя будет выведено уведомление об ошибке с указанием отсутствующего столбца в файле, представленное на рисунке 4.5.

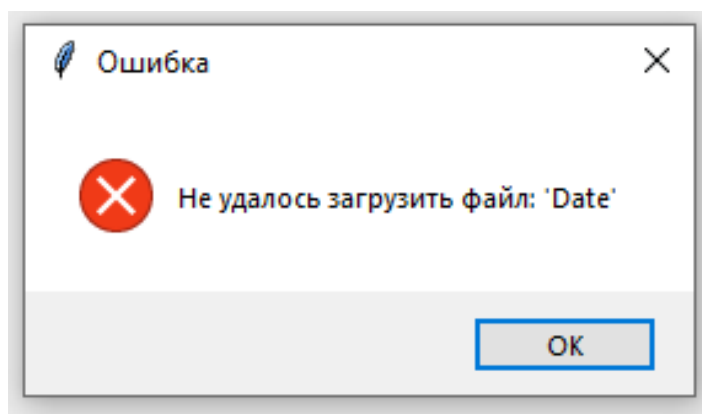


Рисунок 4.5 – Ошибка загрузки данных

Если данные загружены успешно, то программное средство выведет на экран окно выбора страны, а также диапазона дат, и выбранная пользователем информация в указанные сроки будет считаться начальными данными для моделирования, как показано на рисунке 4.6.

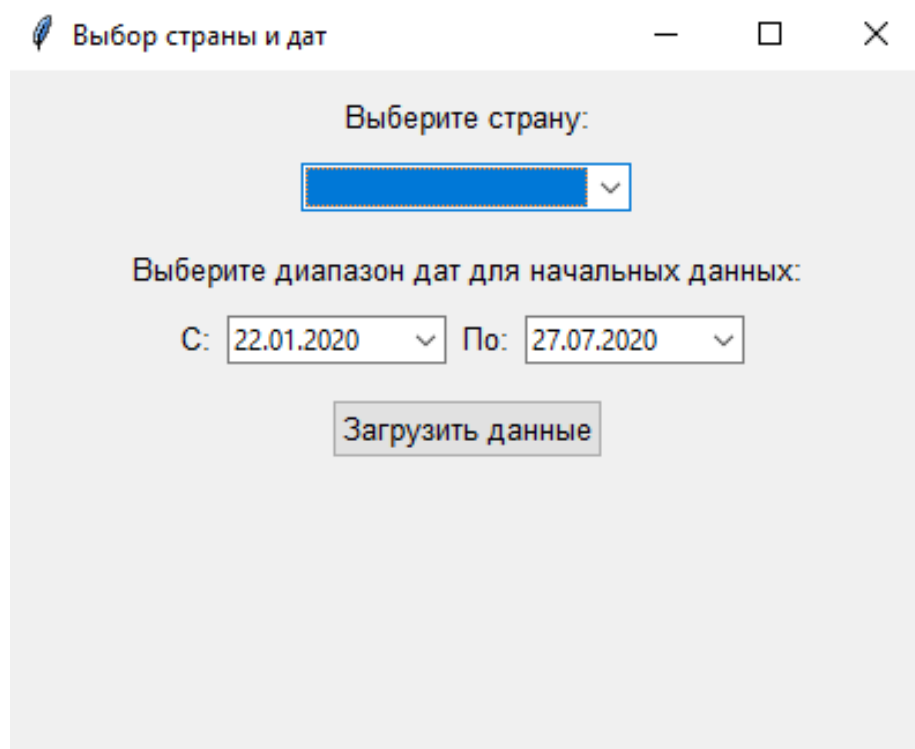


Рисунок 4.6 – Окно выбора страны и дат

После нажатия на кнопку «Загрузить данные» на экране пользователя появится уведомление об успехе операции с указанием страны и периода, выбранных пользователем. Пример такого уведомления представлен на рисунке 4.7.

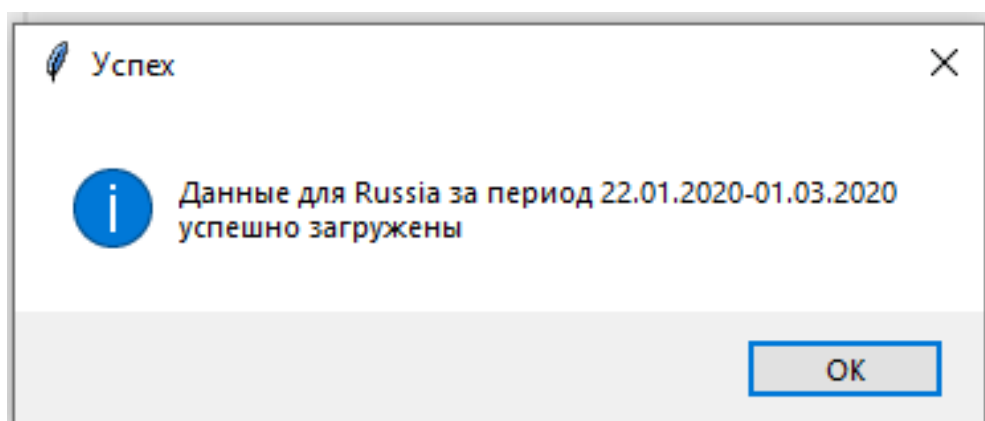


Рисунок 4.7 – Уведомление об успешной загрузке данных

Загруженные данные нормализуются и доли населения S_0 (восприимчивых), I_0 (инфицированных) и R_0 (выздоровевших) в сумме должны составлять единицу. Программное средство также автоматически рассчитывает

параметры скорости заражения и скорости выздоровления на основе загруженных данных. Пример распределения начальных значений изображен на рисунке 4.8.

Параметры модели	
β	0.0841
γ	0.0259

Начальные значения	
S_0	0.9902
I_0	0.0056
R_0	0.0042

Рисунок 4.8 – Распределение долей населения и параметров инфекции

Дата начала прогнозирования автоматически определяется как конечная дата выбора начальных данных из файла. Конечная дата прогнозирования отличается от начальной на 3 месяца, что показано на рисунке 4.9.

Временной диапазон

Начальная дата:
27.07.2020

Конечная дата:
04.11.2020

Рисунок 4.9 – Временной диапазон прогнозирования

Пользователь может заранее или после загрузки данных из CSV файла выбрать начальную и конечную дату прогнозирования, примеры которых представлены на рисунках 4.10 и 4.11.

Начальная дата:

27.07.2020

July 2020

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	29	30	1	2	3	4	5
28	6	7	8	9	10	11	12
29	13	14	15	16	17	18	19
30	20	21	22	23	24	25	26
31	27	28	29	30	31	1	2
32	3	4	5	6	7	8	9

Рисунок 4.10 – Выбор начальной даты

Конечная дата:

04.11.2020

November 2020

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
44	26	27	28	29	30	31	1
45	2	3	4	5	6	7	8
46	9	10	11	12	13	14	15
47	16	17	18	19	20	21	22
48	23	24	25	26	27	28	29
49	30	1	2	3	4	5	6

Рисунок 4.11 – Выбор конечной даты

Пользователь может заранее или после загрузки данных из CSV файла выбрать от одной до четырех математических моделей. Если этот выбор превышает ограничение, то на экран будет выведено сообщение о превышении лимита, изображенное на рисунке 4.12.

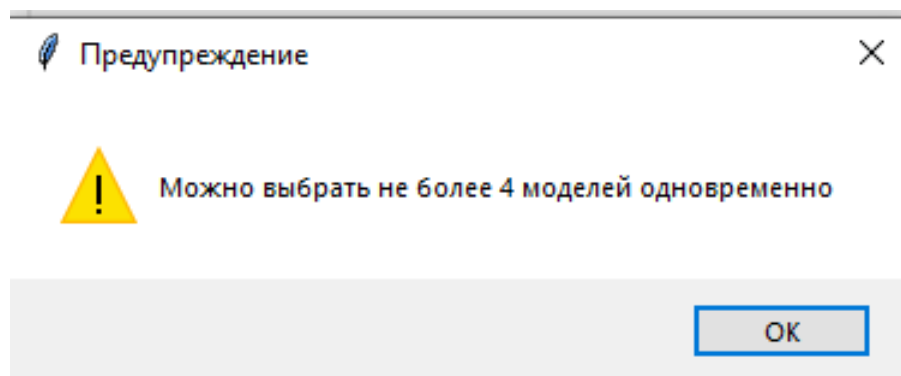


Рисунок 4.12 – Предупреждение об превышении лимита

Пользователь также может заранее или после выгрузки данных выбрать численный метод: Рунге-Кутта 4-го порядка или Эйлера. Чтобы начать моделирование пользователю необходимо нажать на кнопку «Запустить моделирование».

4.4 Демонстрация математических моделей

Представленные в данной работе модели отличаются друг от друга как параметрами, так и начальными условиями, каждую из них необходимо тестировать отдельно. Во всех контрольных примерах берется временной диапазон в 60 дней, начальная дата которого 15.05.2025, а конечная 14.07.2025. Все контрольные примеры решаются с помощью численного метода Рунге-Кутта 4-го порядка.

4.4.1 SIR-модель

SIR-модель состоит из трех начальных значений (восприимчивые, инфицированные и выздоровевшие) и двух параметров (скорость заражения и

скорость выздоровления). Для первого примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.1$. Результат прогнозирования для введенных значений показан на рисунке 4.13.

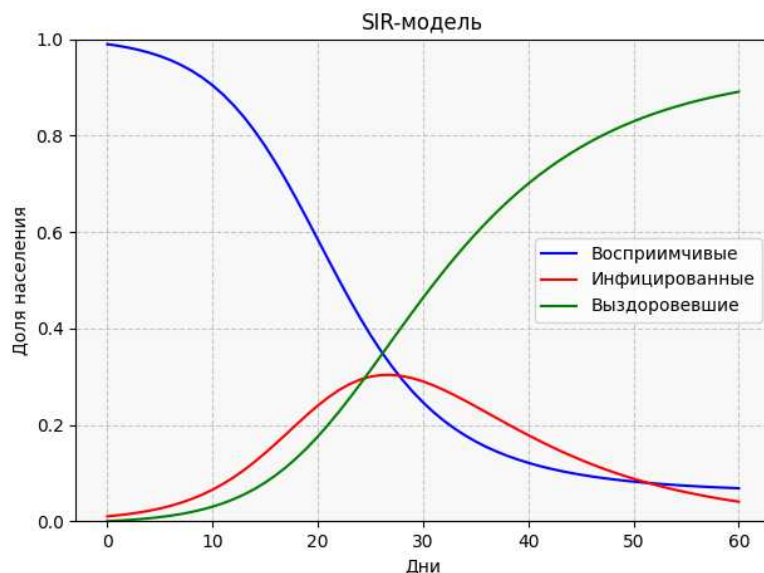


Рисунок 4.13 – Результат SIR-модели для первого примера

Для второго примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.5$. Результат прогнозирования для данного примера представлен на рисунке 4.14.

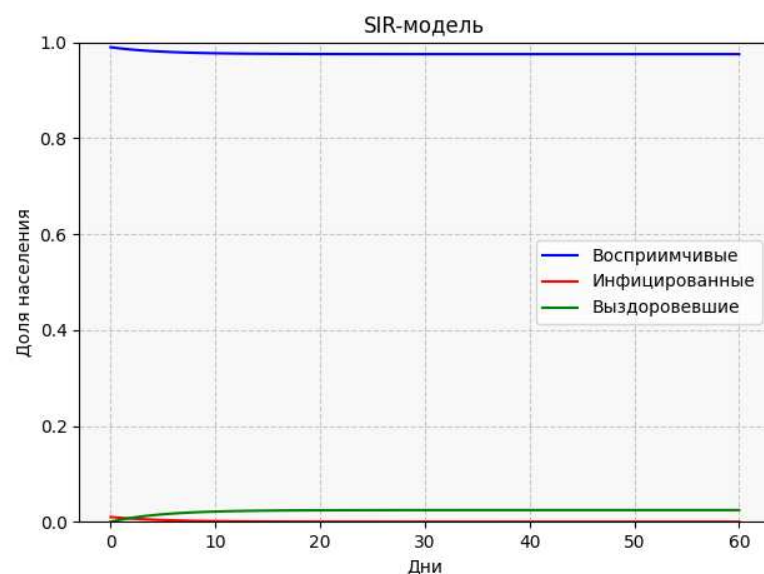


Рисунок 4.14 – Результат SIR-модели для второго примера

На графике видно, что инфицированные выздоравливают быстрее, а так как начальное количество данной доли населения мало, то эпидемия заканчивается в первые же дни своего появления. Увеличивая долю инфицированных до 0.25 и уменьшая восприимчивых до 0.75, остальные параметры остаются без изменения. Результат прогнозирования с измененными данными показан на рисунке 4.15.

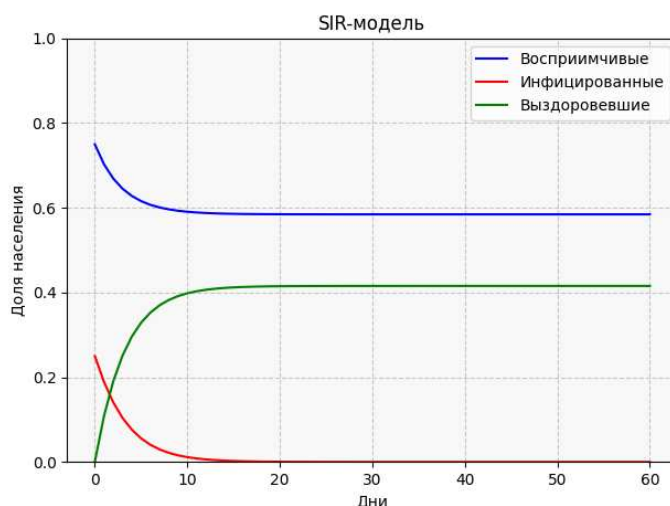


Рисунок 4.15 – Результат прогнозирования с измененными начальными данными для второго примера

На графике можно увидеть, что несмотря на увеличение количества восприимчивых, эпидемия все равно заканчивается в первый десяток дней. Пересечение графиков, выздоровевших и инфицированных примерно похоже на пересечение этих же графиков в предыдущем варианте второго примера.

4.4.2 SI-модель

SI-модель состоит из двух начальных условий (восприимчивые и инфицированные) и одного параметра (скорость заражения). Для первого примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $\beta = 0.3$. Результат моделирования изображен на рисунке 4.16.

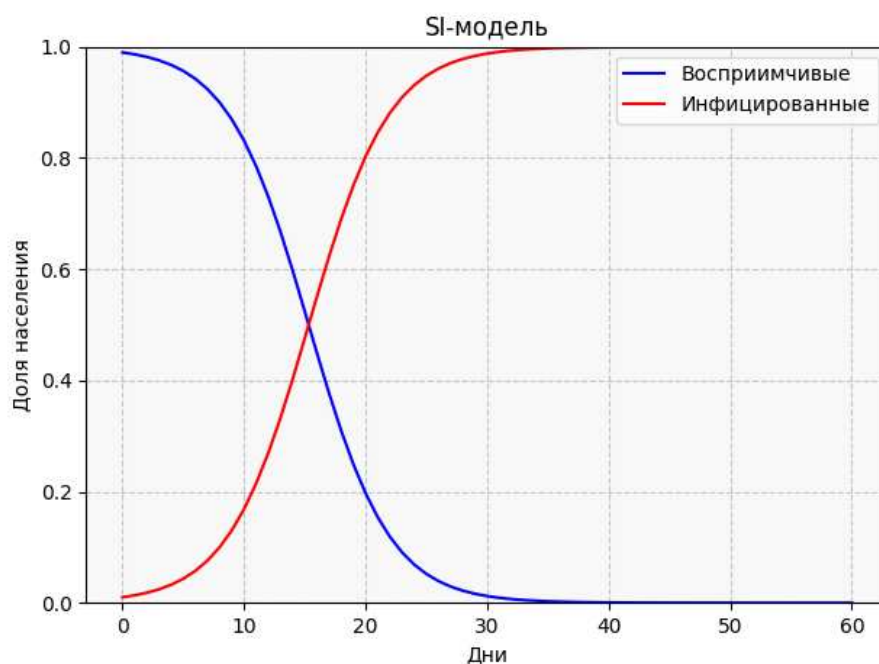


Рисунок 4.16 – Результат SI-модели для первого примера

Для второго примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $\beta = 0.6$. Результат прогнозирования показан на рисунке 4.17.

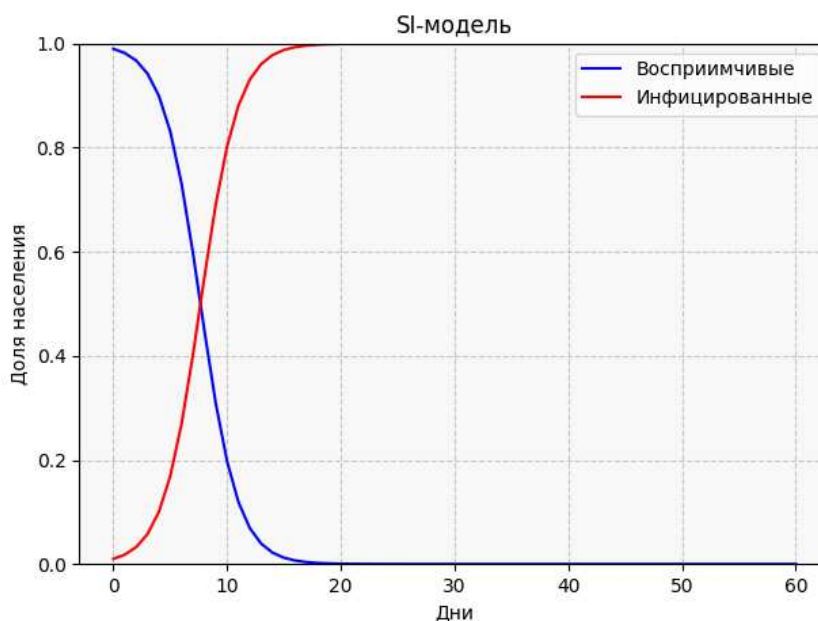


Рисунок 4.17 - Результат SI-модели для второго примера

На графике видно, что с увеличением скорости заражения рост инфицированных ускоряется. Количество инфицированных и восприимчивых к

болезни наступает в первый десяток дней, когда в предыдущем примере это происходило только во второй десяток.

4.4.3 SIRS-модель

SIRS-модель состоит из трех начальных значений (восприимчивые, инфицированные и выздоровевшие) и трех параметров (скорость заражения, скорость выздоровления и скорость потери иммунитета). Для первого примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.1$, $\delta = 0.01$. Результат прогнозирования показан на рисунке 4.18.

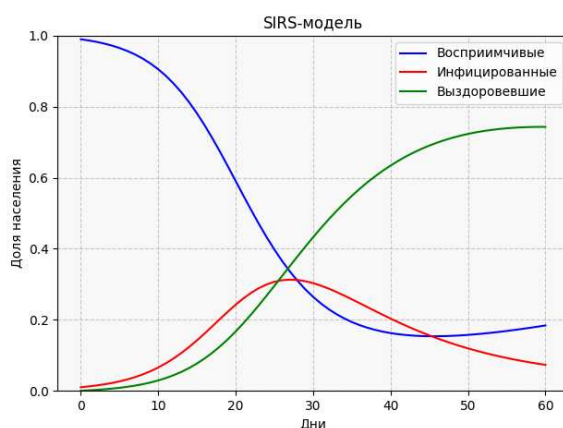


Рисунок 4.18 – Результат SIRS-модели для первого примера

Для второго примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.1$, $\delta = 0.5$. Результат моделирования для второго примера изображен на рисунке 4.19.

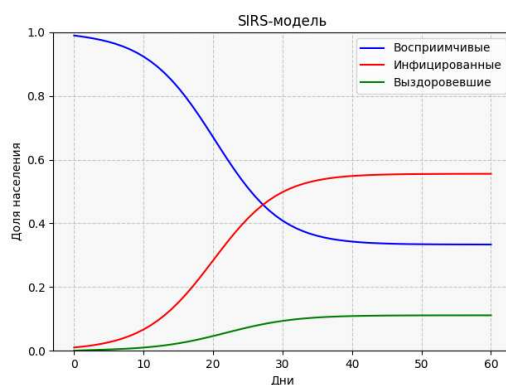


Рисунок 4.19 – Результат SIRS-модели для второго примера

На графике видно, что увеличение параметра скорости потери иммунитета изменяет динамику течения эпидемии. Выздоровевшая доля населения быстрее теряет иммунитет, а значит быстрее заболевает повторно. Если в первом примере инфицированных к 60 дню было около 0.1, то во втором контрольном примере почти 0.6 и их количество практически не изменяется.

4.4.4 SIQR-модель

SIQR-модель состоит из четырех начальных значений (восприимчивые, инфицированные, изолированные и выздоровевшие) и четырех параметров (скорость заражения, скорость выздоровления, скорость перехода на карантин и скорость выздоровления изолированных). Для первого примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $Q_0 = 0.0$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.1$, $\delta = 0.05$, $\mu = 0.05$. Результат для первого примера представлен на рисунке 4.20.

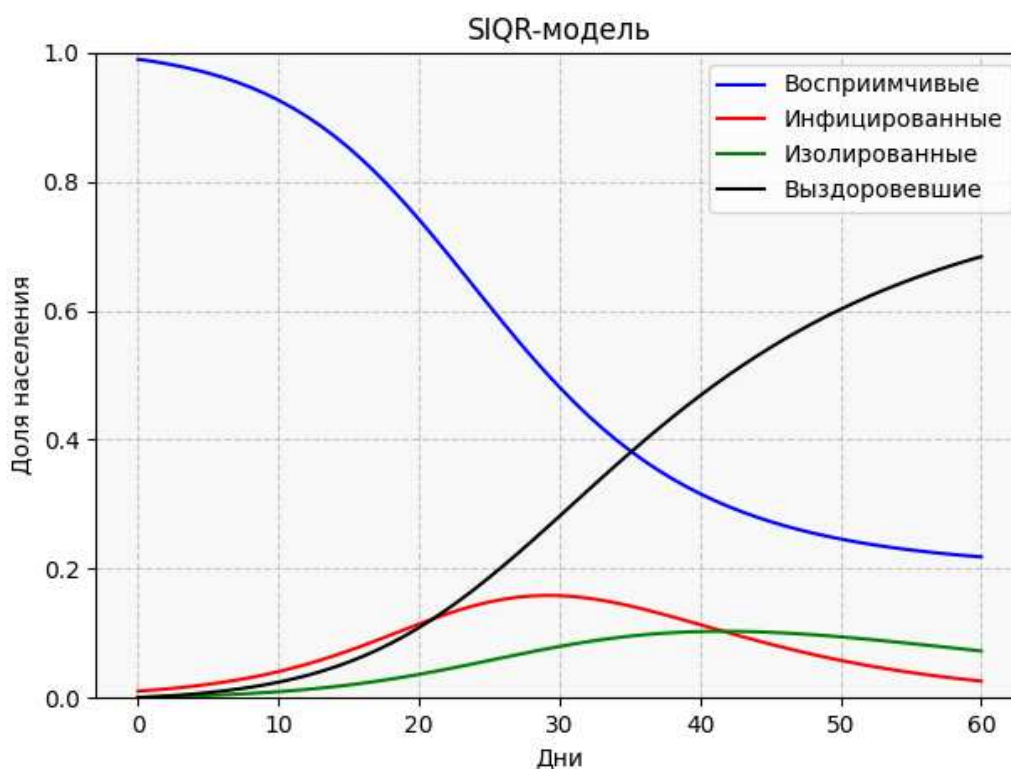


Рисунок 4.20 – Результат SIQR-модели для первого примера

Для второго примера вводятся следующие начальные данные: $S_0 = 0.99$, $I_0 = 0.01$, $Q_0 = 0.0$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.1$, $\delta = 0.1$, $\mu = 0.05$. Результат прогнозирования показан на рисунке 4.21.

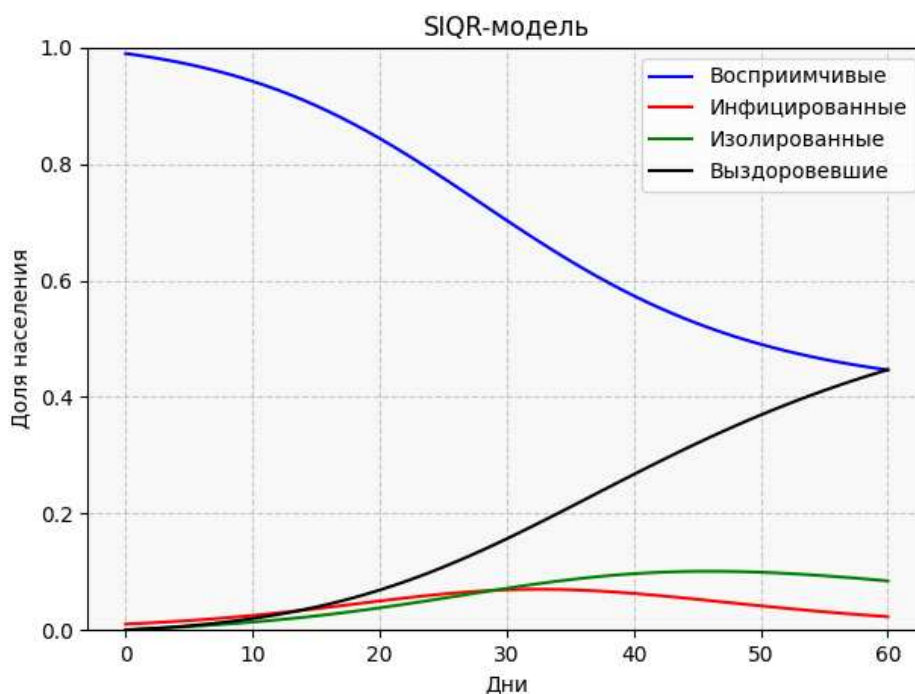


Рисунок 4.21 – Результат SIQR-модели для второго примера

На графике видно, что увеличение параметра скорости изоляции инфицированных сильно влияет на течение эпидемии. В первом случае количество инфицированных начинало снижаться с увеличением роста изолированных, но их пик наступил позже и находится в пределах 0.1 доли населения, когда пик инфицированных был примерно 0.7-0.8. Во втором случае наблюдается ситуация, когда изолированных намного больше, чем инфицированных. Количество выздоровевших и восприимчивых совпадает ровно на 60 день, а рост инфицированных снижается до 0.05-0.1.

4.4.5 SEIR-модель

SEIR-модель состоит из четырех начальных значений (восприимчивые, инфицированные, латентные и выздоровевшие) и трех параметров (скорость

заражения, скорость выздоровления и скорость выхода из латентного периода). Для первого примера вводятся следующие начальные данные: $S_0 = 0.99$, $E_0 = 0.1$, $I_0 = 0.1$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.1$, $\sigma = 0.2$. Результат моделирования возможного развития эпидемиологической ситуации представлен на рисунке 4.22.

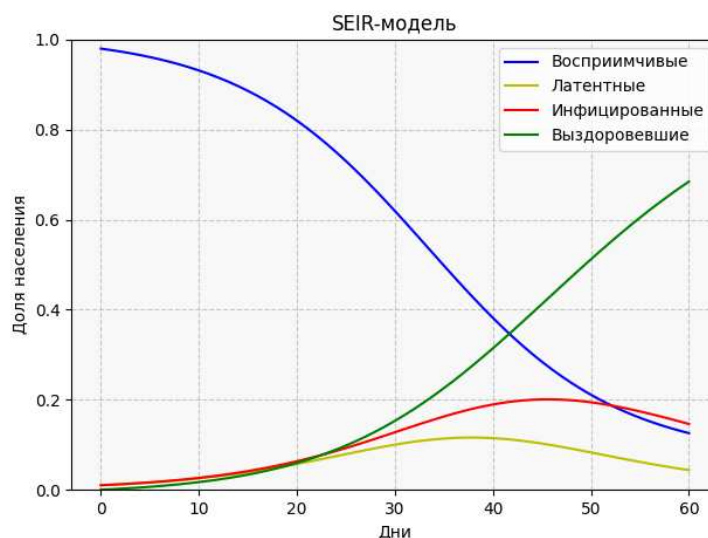


Рисунок 4.22 – Результат SEIR-модели для первого примера

Для второго примера вводятся следующие начальные данные: $S_0 = 0.99$, $E_0 = 0.0$, $I_0 = 0.01$, $R_0 = 0.0$, $\beta = 0.3$, $\gamma = 0.1$, $\sigma = 0.4$. Результат моделирования изображен на рисунке 4.23.

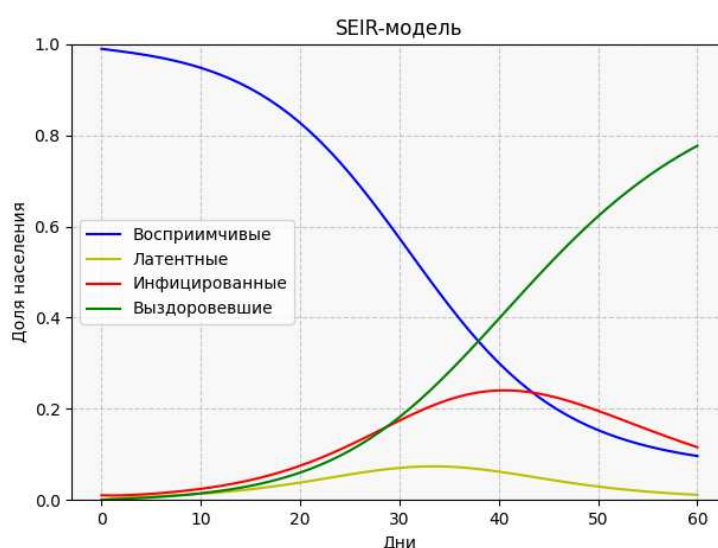


Рисунок 4.23 – Результат SEIR-модели для второго примера

На графике видно, что увеличение параметра выхода из латентного состояния способствует резкому росту количества инфицированных. Пик эпидемии наступает раньше, чем в первом случае. Количество инфицированных также становится большим. В первом случае в пике их было не больше 0.2, когда во втором случае их пик приходится на 0.25-0.3

4.4.6 MSEIR-модель

MSEIR-модель состоит из пяти начальных значений (младенцы с материнским иммунитетом, восприимчивые, инфицированные, латентные и выздоровевшие) и пяти параметров (скорость заражения, скорость выздоровления, скорость выхода из латентного периода, естественная смертность/рождаемость и скорость потери материнского иммунитета). Для первого примера вводятся следующие начальные данные: $M_0 = 0.3$, $S_0 = 0.69$, $E_0 = 0.0$, $I_0 = 0.01$, $R_0 = 0.0$, $\beta = 0.5$, $\gamma = 0.14$, $\sigma = 0.2$, $\delta = 0.1$, $\mu = 0.01$. Результат моделирования показан на рисунке 4.24.

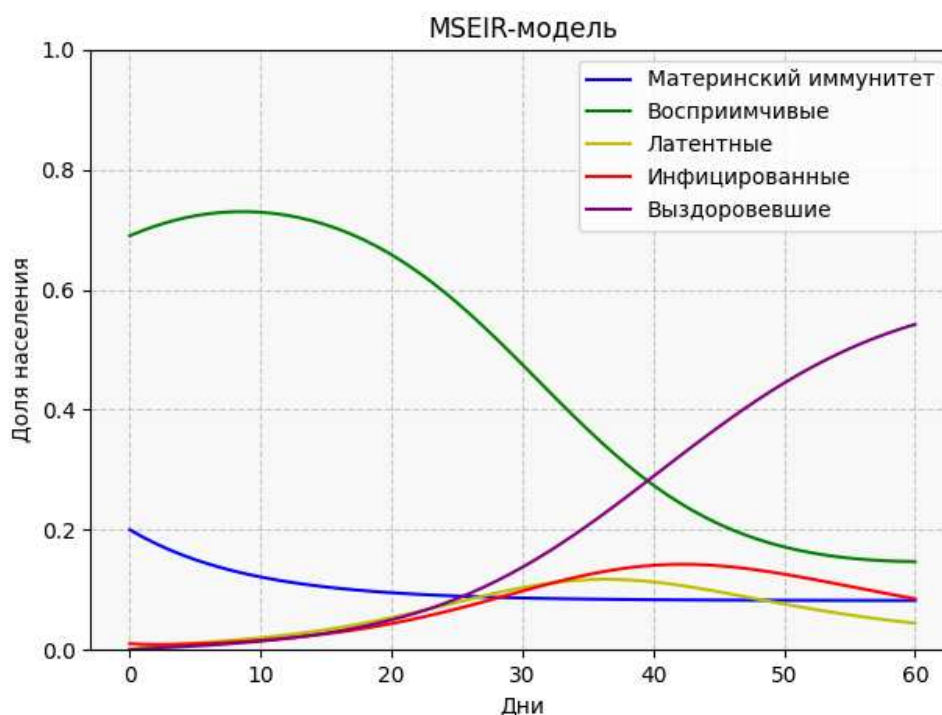


Рисунок 4.24 – Результат MSEIR-модели для первого примера

Для второго примера вводятся следующие начальные данные: $M_0 = 0.3$, $S_0 = 0.69$, $E_0 = 0.0$, $I_0 = 0.01$, $R_0 = 0.0$, $\beta = 0.5$, $\gamma = 0.14$, $\sigma = 0.2$, $\delta = 0.01$, $\mu = 0.01$. Результат прогнозирования развития эпидемиологической ситуации представлен на рисунке 4.25.

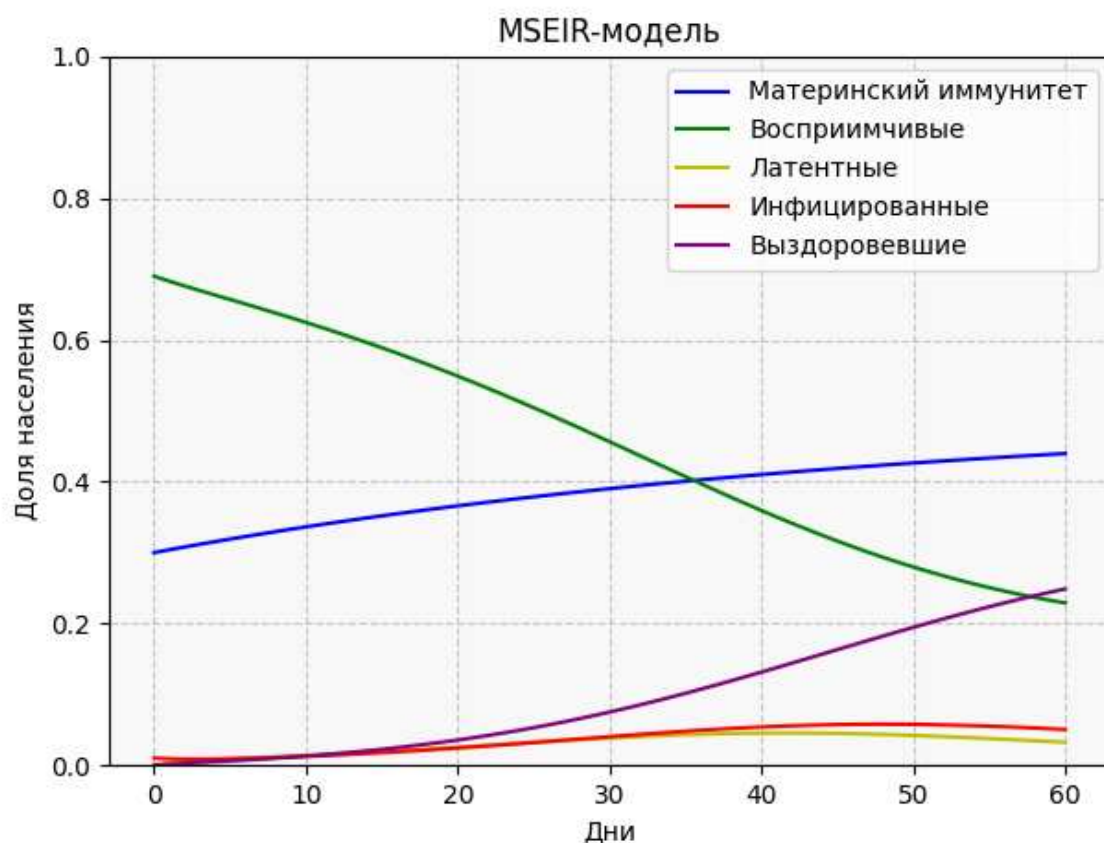


Рисунок 4.25 – Результат MSEIR-модели для второго примера

На графике видно, что уменьшение параметра потери материнского иммунитета снижает пик инфицированных, а также способствует сдерживанию эпидемии, так как к 60 дню количество заболевших снижается.

4.4.7 М-модель

М-модель содержит в своей основе SIRS-модель, которая состоит из трех начальных значений (восприимчивые, инфицированные и выздоровевшие) и трех параметров (скорость заражения, скорость выздоровления и скорость

потери иммунитета). Поскольку Multi-stage модель поддерживает многостадийность, то этап инфицирования разделится на 3 подгруппы: I_0 , I_1 и I_2 , где для каждой необходимо задать числовое значение, которое в сумме с остальными не будет превышать единицы. Параметр скорости заражения заменен на коэффициент скорости перехода на следующую стадию инфекции k_i , где $1 \leq i \leq 3$. Для первого примера вводятся следующие начальные данные: $S_0 = 0.90$, $I_0 = 0.10$, $I_1 = 0.0$, $I_2 = 0.0$, $R_0 = 0.0$, $\beta = 0.5$, $\gamma = 0.05$, $k_1 = 0.3$, $k_2 = 0.2$, $k_3 = 0.1$. Результат прогнозирования на основе введенных данных показан на рисунке 4.26.

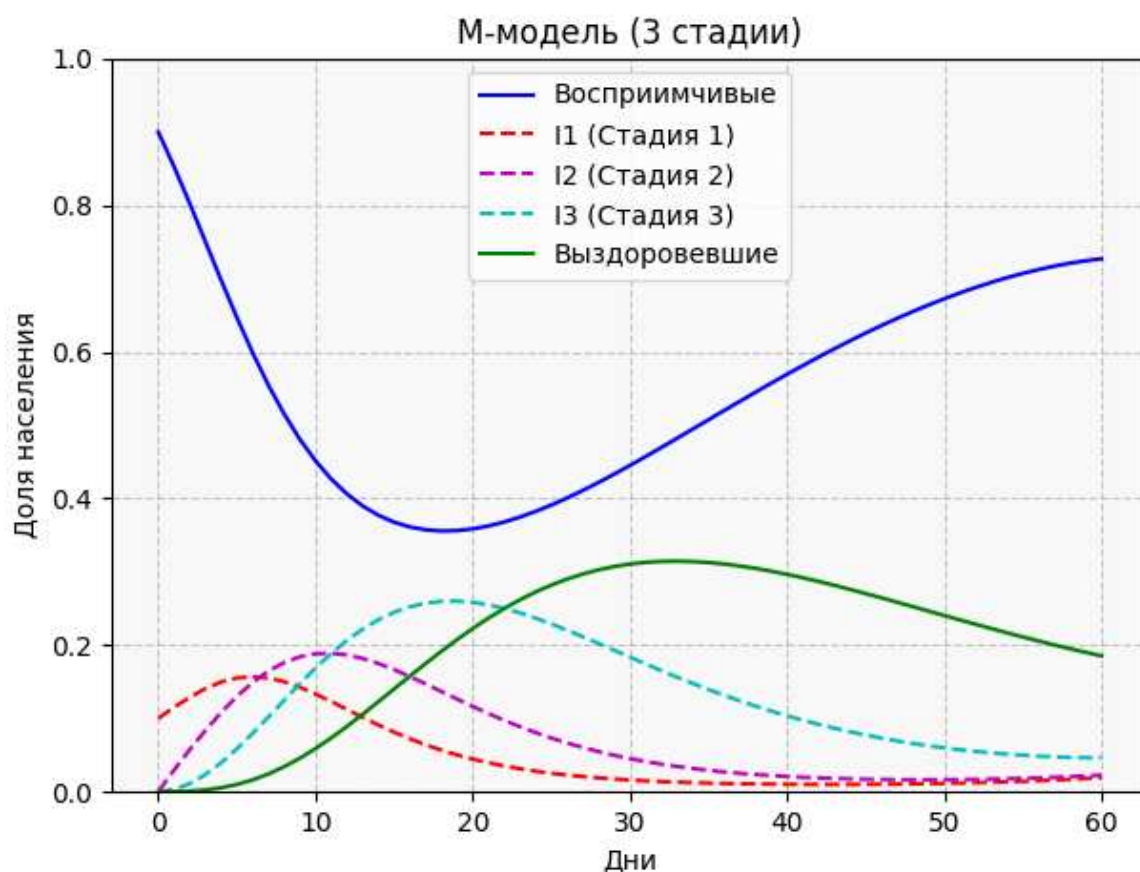


Рисунок 4.26 – Результат М-модели для первого примера

На графике видно, что постепенно люди, болеющие первой стадией заболевания, начинают переходить во вторую стадию, пик которой наступает к 10 дню. После данного пика начинают преобладать люди, болеющие третьей

стадией болезни, пик которых наступает ближе к 20 дню. После этого инфицированная группа людей начинает переходить в группу выздоровевших.

Для второго примера увеличим коэффициенты k_2 и k_3 до значения 0.3, тем самым сделав скорость перехода между стадиями инфекции равной. Результат моделирования с измененными входными данными изображен на рисунке 4.27.

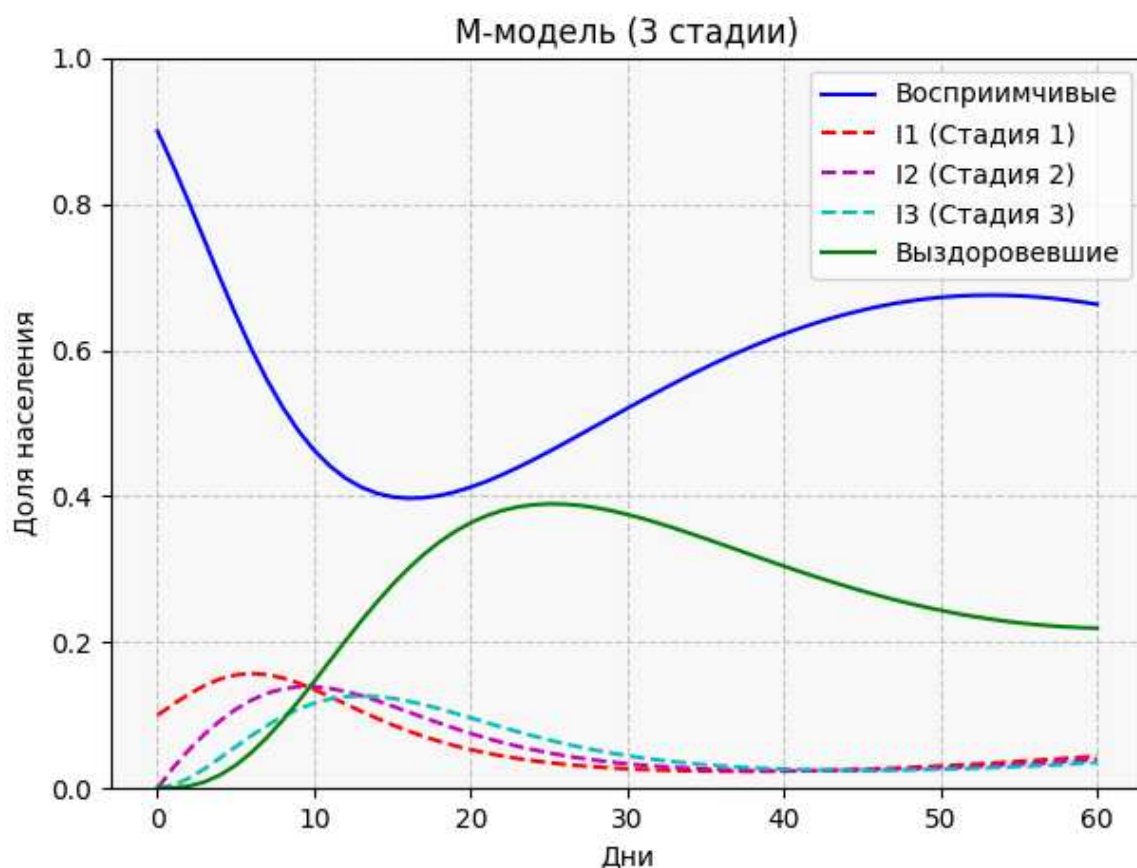


Рисунок 4.27 – Результат М-модели для второго примера

На основе результатов второго примера можно заметить, что разница между стадиями к 10 дню становится примерно одинаковой, так как количество инфицированных на первой, второй и третьей стадиях выравнивается. Введенные входные данные также повлияли на экстремумы кривых восприимчивых и выздоровевших. К 30 дню во втором контрольном примере количество восприимчивых увеличилось примерно наполовину, а к 35 дню

можно заметить увеличившееся количество выздоровевших и их медленное уменьшение к 60 дню.

4.5 Анализ результатов моделирования

На основе представленных в данной работе математических моделей и их тестировании можно выделить следующие особенности:

- SIR-модель подходит для моделирования заболеваний, иммунитет для которых постоянен и сохраняется на всю жизнь. К таким заболеваниям относится ветряная оспа;

- SI-модель подходит для моделирования заболеваний, у которых отсутствует этап выздоровления. К таким заболеваниям относится гепатит В и С, а также ВИЧ/СПИД инфекции;

- SIRS-модель позволяет прогнозировать развитие эпидемиологической ситуации для более продолжительного срока, так как она учитывает потерю иммунитета у населения. Данная модель наиболее подходит для моделирования распространения ротавирусных и респираторных вирусов;

- SEIR-модель позволяет моделировать инфекции, у которых есть латентный (инкубационный) период. К таким инфекциям относится COVID-19;

- SIQR-модель подходит для моделирования развития эпидемиологической ситуации, при которой необходимо проанализировать воздействие карантинных мер. Данная модель наиболее полезна при моделировании развития эпидемии COVID-19 и других респираторных инфекций;

- MSEIR-модель подходит для моделирования распространения инфекций наиболее опасных для новорожденных. К таким заболеваниям относится корь;

- M-модель подходит для моделирования распространения множества инфекционных заболеваний, при которых необходимо отслеживать количество

пациентов, болеющих различными стадиями заболевания. Данная модель также позволяет изменять её основу, заменяя одну математическую модель на другую.

Подводя итог анализа моделей, стоит отметить, что для каждой инфекции необходимо выбирать наиболее подходящий инструментарий, позволяющий представить полную картину возможного развития эпидемиологической ситуации. В таблице 4.2 приведены инфекции и модели, наиболее подходящие для их моделирования.

Таблица 4.2 – Инфекции, подходящие для моделирования

Название модели	Наиболее подходящие инфекции для моделирования
SIR	Грипп, ветряная оспа, корь, эпидемический паротит
SI	ВИЧ, гепатит В/С и другие хронические вирусные инфекции
SIRS	Риновирусы, ротавирусные инфекции, стрептококк группы А
SIQR	COVID-19, туберкулез, Эбола, другие инфекции с жесткими карантинными мерами
SEIR	COVID-19, туберкулез, корь, ветряная оспа
MSEIR	Краснуха, корь, ветряная оспа, инфекции с пассивным материнским иммунитетом
М-модель	Туберкулез, ВИЧ-инфекция, малярия, гепатит В/С, коронавирусные и герпетические инфекции

4.6 Выводы по главе

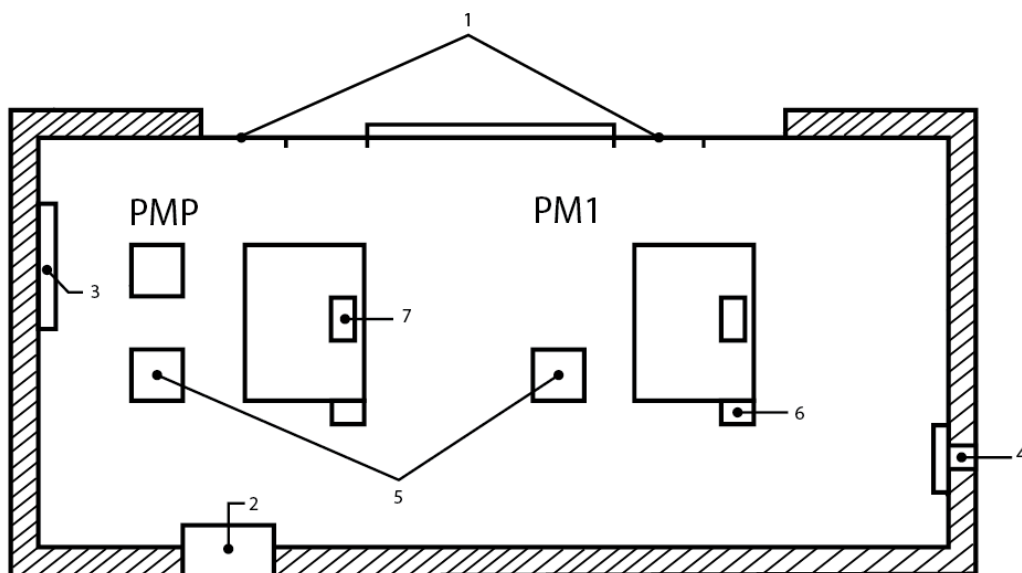
В данной главе протестировано программное средство и реализованные математические модели с использованием различных начальных данных. На основе анализа, представленных в данной работе моделей, выделены инфекции наиболее подходящие для моделирования развития эпидемиологической ситуации.

5 Безопасность и экологичность при разработке программного средства

В данном разделе предлагаются рекомендации по оптимальному освещению для раздела «Общие требования безопасности» в плане-конспекте вводного инструктажа для сотрудников отдела программного обеспечения ДГТУ. Цель этих рекомендаций – минимизировать вредное влияние работы за компьютером на зрение.

5.1 Оценка опасности вредных факторов на рабочем месте разработчика

В состав рабочего места разработка программной реализации математического моделирования развития эпидемиологической ситуации входят персональный компьютер с видеодисплейным терминалом (ПК), принтер, ксерокс и искусственный светильник. План расположения рабочих мест представлен на рисунке 5.1.



1 – окна с форточкой; 2 – дверь; 3 – доска с документацией; 4 – кондиционер;
5 – документация в папках; 6 – искусственный светильник; 7 – ПК; РМР –
рабочее место руководителя; PM1 – рабочее место сотрудника.

Рисунок 5.1 – План расположения рабочих мест в помещении

В помещении, где расположены дисплеи и панели управления ЭВМ, соблюдены следующие требования санитарных норм:

- ширина проходов перед оборудованием составляет не менее 1 метра;
- расстояние от оборудования до стен также не менее 1 метра;
- рабочие места расположены на расстоянии не менее 1,2 метра друг от друга.

Данное соответствие указанным нормам подтверждает, что условия труда соответствуют установленным требованиям. Рабочее помещение находится на втором этаже здания. Кроме того, помещение оборудовано автоматической пожарной сигнализацией и средствами пожаротушения, к которым обеспечен свободный доступ, что также соответствует требованиям пожарной безопасности.

Работа программиста классифицируется как легкий физический труд (категория I, подкатегория Ia), характеризующийся сидячей работой с минимальными физическими усилиями и энергозатратами до 139 Вт.

В помещении, где работает программист, отсутствуют источники избыточного тепла и вредных веществ. Отопление обеспечивается централизованной системой. При недостаточном отоплении допускается использование дополнительных обогревателей (радиаторов, кондиционеров с функцией обогрева). В летний период для поддержания комфортной температуры используются системы кондиционирования или естественная вентиляция. Регулярно проводится проветривание и влажная уборка помещения.

В помещении основными генераторами шума выступают системы охлаждения компьютеров, печатающее оборудование и человеческая речь. Вибрационные воздействия отсутствуют.

В отделе разработки программного обеспечения, где сотрудники проводят большую часть рабочего времени за компьютерами, освещенность является критически важным фактором рабочей среды. Нормативные значения

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

освещенности, согласно СНиП 23-05-95, должны находиться в диапазоне от 150 до 300 люкс.

Для определения наиболее опасного вредного фактора для разработчика составлена таблица 5.1, в которой дана оценка опасности для каждого фактора.

Таблица 5.1 – Оценка опасных вредных факторов на рабочем месте

№	Опасный вредный фактор	В чем выражается	Защита от опасного вредного фактора	Оценка опасности вредного фактора по 10-ти бальной шкале
1	Шум	Шум от работающих устройств и внешний шум с улицы	Малошумное оборудование; звукоизолирующие стены, экраны, окна, двери	3
2	Освещение	Неверно организованное освещение помещения	Антибликовые покрытия, эргономика рабочего места, регулируемое освещение	6
3	Микроклимат	Несоответствующие нормам температура воздуха и уровень влажности в помещении	Проветривание или кондиционирование помещения и влажная уборка	4
4	Вибрации	Создаваемые работающими устройствами вибрации	Виброизолирующие технологии	2

На основе приведенной оценки опасных факторов на рабочем месте разработчика можно выделить фактор освещения помещения. Работа программиста требует высокой концентрации и связана с интенсивной нагрузкой на зрение, что может вызывать усталость и влиять на нервную систему. Длительное пребывание перед монитором часто приводит к головным болям и дискомфорту в глазах. Чтобы снизить негативное воздействие компьютера на здоровье, необходимо соблюдать правила безопасности, правильно организовывать рабочий процесс и обустраивать рабочее место.

5.2 Расчет освещения на рабочем месте разработчика

В отделе разработки программного обеспечения, где сотрудники проводят большую часть рабочего времени за компьютерами, освещенность является критически важным фактором рабочей среды. Для измерения этого параметра используется люксметр модели Ю-117. Нормативные значения освещенности, согласно СНиП 23-05-95, должны находиться в диапазоне от 150 до 300 люкс. В следующем разделе подробно рассмотрена концепция рационального освещения и предложены практические рекомендации по обеспечению требуемого уровня освещенности в данном отделе.

Для качественного выполнения зрительных задач требуется не только достаточная освещенность, но и определенные качественные характеристики освещения. К ним относятся равномерное распределение света, отсутствие слепящего эффекта (блескости), правильный выбор фона и оптимальный контраст между объектом и фоном.

Согласно ГОСТ Р 56228-2014, светоотражающие свойства поверхностей оцениваются коэффициентами отражения (ρ), пропускания (τ) и поглощения (β), которые являются безразмерными величинами. ГОСТ 55710-2013 выделяет два типа блескости: прямую (от ярких источников света) и отраженную (от зеркальных поверхностей). Отраженная блескость, попадая в поле зрения, снижает зрительную чувствительность и работоспособность, вызывая слепимость. Слепящий эффект зависит от яркости поверхности, направленной в глаз, и от контраста объекта с фоном (K). Контраст определяется как отношение разницы яркостей объекта и фона к яркости объекта. Чем темнее фон, тем сильнее слепящий эффект. Согласно ГОСТ Р 56228-2014, средний контраст объекта с фоном должен составлять 0,2-0,5.

Фоновая поверхность, непосредственно окружающая объект, должна иметь коэффициент отражения в диапазоне 0,2-0,5. Для предотвращения дискомфорта от яркого света, необходимо оптимизировать высоту подвеса

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		69

светильников, принимая во внимание их мощность, угол падения света на рабочую зону и характеристики отражающих поверхностей. Повышение контраста между объектами и фоном является более эффективным и экономичным способом улучшения видимости, чем простое увеличение общей освещенности. При этом необходимо учитывать цветовые характеристики и коэффициенты отражения как объектов, так и фона. Для достижения равномерного распределения яркости в поле зрения рекомендуется использовать светлые оттенки (например, бежевый, салатовый, бирюзовый) для окраски стен и потолка.

Учитывая тот факт, что программисты проводят много времени, работая с мониторами, крайне важно обеспечить правильное освещение для поддержания здоровья и эффективности центрального зрения. Основные принципы организации освещения рабочего места включают:

- сбалансированный контраст между яркостью экрана и окружающим пространством;
- исключение работы за компьютером в темноте или полумраке.

Для освещения рабочих, офисных и жилых помещений используются как естественный, так и искусственный свет. В большинстве случаев для эффективной организации работы необходимо сочетание этих двух видов освещения. Далее подробно рассмотрен каждый из них.

Для определения параметров общего равномерного освещения рабочего пространства, визуализированного на рисунке 5.1, используется метод коэффициента использования светового потока. Данный метод учитывает вклад отраженного света от поверхностей стен и потолка в общую освещенность. Расчет требуемого светового потока группы светильников с люминесцентными лампами, выраженного в люменах, осуществляется на основе формулы 5.1:

$$\Phi_{\text{Л.РАСЧ}} = \frac{100 \cdot S \cdot Z \cdot K \cdot E_{\text{Н}}}{N \cdot n \cdot \eta} \quad (5.1)$$

где $E_{\text{Н}}$ – нормированная минимальная освещенность, лк;

Z – коэффициент минимальной освещенности (для люминесцентных ламп принимает значение, равное 1,1);

K – коэффициент запаса;

η – коэффициент использования светового потока ламп, зависит от КПД и кривой распределения силы света светильника, коэффициента отражения от потолка и стен, а также от высоты подвеса светильников;

N – количество светильников, шт.;

n – количество ламп в одном светильнике, шт.

Показатель помещения i , необходимый для определения величины η рассчитывается по формуле 5.2:

$$i = \frac{A \cdot B}{H_{\text{п}} \cdot (A + B)}, \quad (5.2)$$

где A и B – длина и ширина рабочего помещения соответственно, м;

$H_{\text{п}}$ – высота подвеса светильника, м.

Исходные данные помещения, изображенного на рисунке 5.1, представлены в таблице 5.2.

Таблица 5.2 – Исходные данные помещения

Параметр помещения	Значение параметра помещения
Длина помещения A , м	8
Ширина помещения B , м	4
Высота помещения h , м	3
Коэффициент отражения от потолка	30
Коэффициент отражения от стен	10
Коэффициент отражения от пола	10
Нормируемая освещенность E_H , лк	300
Длина одного светильника $l_{\text{св}}$, м	1,534
Количество ламп в одном светильнике, шт.	2

Для начала необходимо вычислить высоту подвеса светильника над рабочей поверхностью, м, по формуле 5.3:

$$H_{\Pi} = h - h_p - h_c, \quad (5.3)$$

где h_p – высота рабочей поверхности, м, принимает значение, равное 0,8;

$h_c = 0,25(h - h_p)$ – расстояние светового центра светильника от потолка(свес), м.

$$h_c = 0,25(3 - 0,8) = 0,55 \text{ м},$$

$$H_{\Pi} = 3 - 0,8 - 0,55 = 1,65 \text{ м}.$$

Далее определяем индекс помещения i по формуле 5.3:

$$i = \frac{8 \cdot 4}{1,65 \cdot (8 + 4)} \approx 1,6.$$

Значение коэффициента использования светового потока η для светильников с люминесцентными лампами составляет:

$$\eta = 48.$$

Значение коэффициента запаса для помещения, с малыми выделениями пыли, составляет:

$$K = 1,3.$$

Оптимальное расстояние L между рядами светильников, м, определяется из отношения 5.4:

$$\lambda = \frac{L}{H_{\Pi}}, \quad (5.4)$$

где λ – коэффициент оптимального расстояния, который при освещении рядами люминесцентных ламп принимается по умолчанию 1,1.

$$L = \lambda \cdot H_{\Pi},$$

$$L = 1,1 \cdot 1,65 \approx 1,82 \text{ м}.$$

Ближайшее расстояние от стен до светильников, принимает значение 5.5:

$$l = 0,3 \cdot L,$$

$$l \approx 0,3 \cdot 1,82 \approx 0,54 \text{ м}. \quad (5.5)$$

При выполнении условия равномерного освещения общее количество светильников, шт., находится по следующей формуле (5.6):

$$N = N_{\text{ш}} \cdot N_{\text{дл}}, \quad (5.6)$$

где $N_{\text{ш}}$ – количество светильников по ширине (количество рядов), шт.
полученное по формуле 5.7:

$$N_{\text{ш}} = \frac{B}{L}, \quad (5.7)$$

$$N_{\text{ш}} = \frac{4}{1,82} \approx 2 \text{ шт.}$$

$N_{\text{дл}}$ – количество светильников по длине (для светильников с люминесцентными лампами), шт. рассчитанное по формуле 5.8:

$$N_{\text{дл}} = \frac{(A - l_{\text{св}})}{l_{\text{св}}}, \quad (5.8)$$

$$N_{\text{дл}} = \frac{(8 - 1,534)}{1,534} \approx 4 \text{ шт.}$$

Общее количество светильников по формуле 5.7 составляет:

$$N = 4 \cdot 2 = 8 \text{ шт.}$$

Расчет светового потока одной лампы проведен по формуле 5.1 и определен тип лампы и ее мощность.

$$\Phi_{\text{л.расч}} = \frac{100 \cdot (8 \cdot 4) \cdot 1,1 \cdot 1,3 \cdot 300}{8 \cdot 2 \cdot 48} = 1788 \text{ лм.}$$

В соответствии с расчетами и параметрами люминесцентных ламп общего назначения, в данном помещении необходимо установить светодиодные лампы с мощностью $W = 10$ Вт, силой тока $I = 0,35$ А и напряжением $U = 104 \pm 0,4$ В.

Для удовлетворения проектируемой системой требований СНиП 23-05-95 по освещенности должно быть выполнено следующее условие:

$$E_{\text{ф}} \geq E_{\text{н}} (300 \text{ лк по усл.}).$$

Тогда осуществим расчет фактической освещенности, лк по формуле 5.9:

$$E_{\text{ф}} = \frac{\Phi_{\text{п.табл}} \cdot N \cdot n \cdot \eta}{K \cdot S \cdot Z \cdot 100}, \quad (5.9)$$

$$E_{\text{ф}} = \frac{1800 \cdot 8 \cdot 2 \cdot 48}{1,3 \cdot 32 \cdot 1,1 \cdot 100} \approx 302 \text{ лк.}$$

Далее необходимо определить потребляемую мощность осветительной установки, Вт, по формуле 5.10:

$$P = W \cdot N \cdot n \cdot k_p, \quad (5.10)$$

где W – мощность лампы, Вт;

N – число светильников, шт.;

n – число ламп в светильнике;

k_p – коэффициент, учитывающий потери пускорегулирующей аппаратуры, принимает значение 1,25.

$$P = 10 \cdot 8 \cdot 2 \cdot 1,25 = 200 \text{ Вт.}$$

Схема размещения светильников на плане помещения представлена на рисунке 5.2.

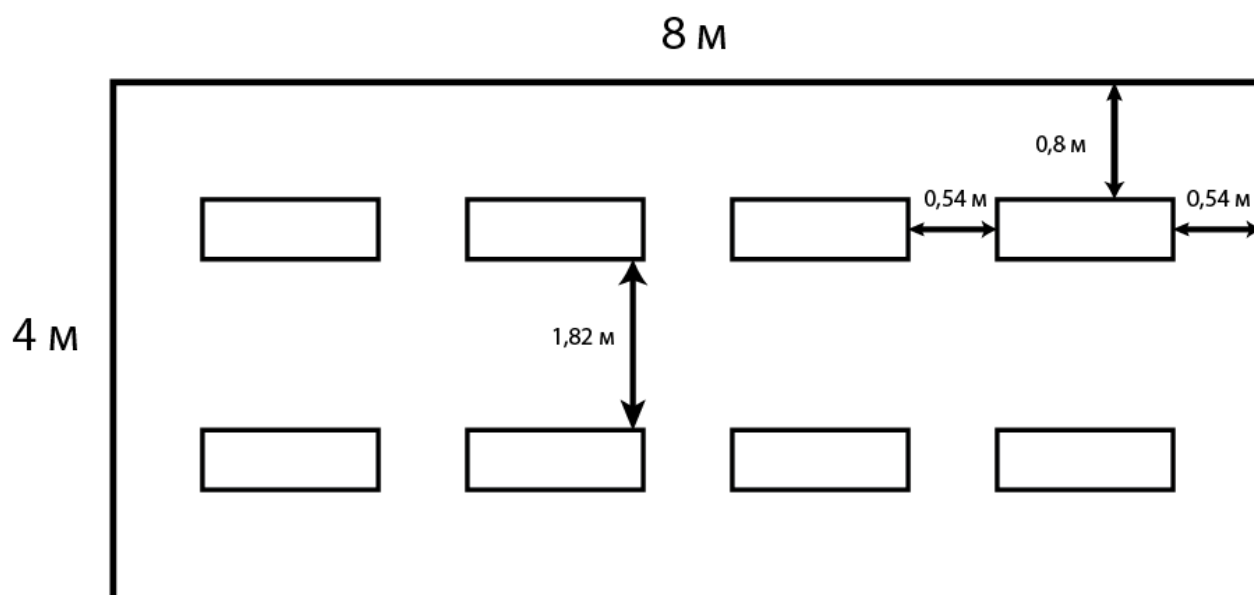


Рисунок 5.2 – Схема размещения светильников

5.3 Экологическая обстановка в районе предприятия

В Ростовской области существует множество экологических проблем. Множество промышленных предприятий и автотранспортных средств ежедневно выбрасывают большое количество вредных веществ в атмосферу. Это приводит к появлению смога и увеличению числа заболеваний дыхательных путей у населения.

Выхлопные газы – основной источник токсичных веществ двигателя внутреннего сгорания – это неоднородная смесь различных газообразных веществ. В своем составе они содержат около 300 веществ, большинство из которых токсичны.

Выхлопные газы и выбросы заводов и фабрик, пыль, сажа, дым, смешиваясь с туманом, образуют смог, который очень токсичен. Однако, власти области предпринимают меры для сокращения выбросов, но, чтобы достичь значимых результатов, необходимо активное участие всех граждан и бизнес-сектора.

В составе выхлопных газов имеются:

- угарный газ;
- оксиды азота;
- ароматические углеводороды;
- соли тяжёлых металлов (соли кадмия, ртути, меди, свинца).

Для снижения уровня загрязнения воздуха можно предпринять следующие действия:

- использование альтернативных видов топлива;
- сокращение выбросов загрязняющих веществ;
- использование оборудования для управления технологическими процессами;
- разбавление загрязняющих веществ в воздухе;
- посадка деревьев в местах с высоким уровнем загрязнения.

5.4 Электро- и пожароопасность на рабочем месте разработчика

В рабочем пространстве разработчика электропитание оборудования осуществляется от сети переменного тока с напряжением 220-230 В и частотой 50 Гц. Используемое оборудование относится к классу 0 электробезопасности.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		75

Это означает, что оно имеет только рабочую изоляцию и не предусматривает заземление. Выбор изоляции для оборудования и его компонентов должен соответствовать рабочему напряжению. Изоляция доступных для прикосновения частей должна предотвращать поражение электрическим током и электрической дугой. В данном случае, все такие участки изолированы. Заземление не требуется, поскольку оборудование относится к классу 0.

Опасность поражения электрическим током возникает в следующих случаях:

- прикосновение к открытым токоведущим частям компьютера;
- неисправность самого оборудования;
- повреждение сетевых розеток или внутренней электропроводки.

Для защиты от поражения электрическим током необходимо:

- обеспечить изоляцию всех токоведущих частей компьютера, используя защитные ограждения и другие средства;
- заземлить все металлические конструкции, которые могут оказаться под напряжением;
- провести инструктаж по электробезопасности для персонала.

Помещение имеет III степень огнестойкости и относится к категории П по пожарной опасности. Основные причины возникновения пожара:

- использование неисправного оборудования с дефектами в схеме;
- короткое замыкание;
- одновременное использование большого количества мощных электроприборов;
- нарушение правил пожарной безопасности;
- неисправность проводки.

Пожарная безопасность объекта достигается комплексом мер, включающим предотвращение пожара, активную противопожарную защиту

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		76

(например, огнетушители ОХП-10 и ОУ-5) и организационно-технические мероприятия. Эти меры должны совместно исключать или минимизировать воздействие опасных факторов пожара на людей, при этом вероятность такого воздействия не должна превышать 10^{-6} в год на человека. Для достижения этой цели необходимо: проводить пожарную профилактику (включая разработку планов эвакуации и инструктажи), создать эффективную систему пожаротушения, установить автоматическую пожарную сигнализацию, использовать только центральное отопление и регулярно проверять электрооборудование.

5.5 Выводы по главе

В данном разделе рассмотрены вопросы безопасности и экологичности разработки программного обеспечения в соответствии с нормативными требованиями. Предложены меры по обеспечению безопасных условий труда, которые должны быть отражены в плане вводного инструктажа. Особое внимание уделено освещению, для которого, на основе нормативных документов, даны рекомендации, произведен расчет системы искусственного освещения и представлена схема размещения светильников.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		77

Заключение

В ходе работы над выпускной квалификационной работой создано программное средство для математического моделирования динамики эпидемиологической ситуации. Разработанная программа предоставляет возможность проведения численных экспериментов с использованием различных параметров модели, анализа распространения инфекции и оценки эффективности мер по ее контролю.

Результаты тестирования подтвердили корректность работы программы и ее пригодность для решения актуальных задач в области эпидемиологии. Интерфейс с возможностью настройки параметров делает его удобным для демонстрации принципов моделирования, проведения лабораторных работ и самостоятельных студенческих исследований.

Созданное программное средство может быть использовано как для научных исследований, так и в качестве вспомогательного средства для принятия решений в сфере управления при возникновении реальной эпидемической угрозы. Программное средство может также применяться в учебных целях в предметах, связанных с математическим моделированием и вычислительной математикой.

Программное средство использует парадигму объектно-ориентированного программирования, что позволяет расширить его функциональные возможности, благодаря внедрению других моделей и численных методов в качестве дальнейшего направления развития проекта.

Разработанное программное средство и результаты исследования представлены на конференции «Актуальные проблемы науки и техники – 2025», что свидетельствует об актуальности проблемы моделирования эпидемиологической ситуации и практической значимости выполненной работы.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		78

Перечень используемых информационных источников

1. Бернулли, Д. Эссе о новом анализе смертности, вызванной оспой, и преимуществах инокуляции для её предотвращения / Д. Бернулли. – 1766. – В кн.: Ван дер Варден Б.Л. (ред.). Сочинения Даниэля Бернулли. Т. 2. – Базель: Birkhäuser, 1982. – С. 235–267.
2. Кермак, У. Математическая теория эпидемий / У. Кермак, А. МакКендрик. – 1927. – В кн.: Избранные труды по математической эпидемиологии. – Москва: Наука, 1980. – С. 45–72.
3. Арсеньев, В.В. Математические методы в биологии / В.В. Арсеньев. – Москва: URSS, 2015. – 320 с. – ISBN 978-5-9710-2073-0.
4. Воронин, А.А. Математические модели распространения эпидемий / А.А. Воронин. – Москва: МГУ, 2019. – 208 с. – ISBN 978-5-19-011784-6.
5. Андерсон, Р.М. Инфекционные болезни человека: динамика и контроль / Р.М. Андерсон, Р.М. Мэй. – Москва: Мир, 1993. – 768 с. – ISBN 978-5-03-003789-6.
6. Брауэр, Ф. Математические модели в эпидемиологии / Ф. Брауэр, К. Кастильо-Чавес, Ч. Фэн. – Москва: Мир, 2020. – 600 с. – ISBN 978-5-03-004567-9.
7. Дикманн, О. Математическая эпидемиология инфекционных болезней: построение моделей, анализ и интерпретация / О. Дикманн, Я.А.П. Хестербек. – Москва: Мир, 2002. – 320 с. – ISBN 978-5-03-003456-2.
8. Ален, Л.Дж.С. Введение в математическую биологию / Л.Дж.С. Ален. – Москва: Мир, 2009. – 320 с. – ISBN 978-5-03-003456-7.
9. Мюррей, Дж. Математическая биология. Т. 1, 2 / Дж. Мюррей. – Москва: ИКИ-РЦД, 2010–2011. – 776 с. – ISBN 978-5-93972-765-3.
10. Хеткот, Х.В. Математика инфекционных болезней / Х.В. Хеткот. – Москва: Мир, 2001. – 55 с.

11. Эдельштейн-Кешет, Л. Математические модели в биологии / Л. Эдельштейн-Кешет. – Москва: Мир, 2006. – 640 с. – ISBN 978-5-03-003789-1.

12. Сиденко, С. Методы численного анализа / С. Сиденко. – Москва: Физматлит, 2018. – 416 с. – ISBN 978-5-9221-1813-9.

13. Кутта, М. Введение в численные методы / М. Кутта. – Москва: Наука, 1987. – 256 с.

14. Бутлер, М. Программирование на Python: от простого к сложному / М. Бутлер. – Санкт-Петербург: Питер, 2021. – 352 с. – ISBN 978-5-4461-0923-3.

15. Гудман Б. Python и анализ данных / Б. Гудман. – Москва: Диалектика, 2022. – 251 с. – ISBN 978-5-8459-2531-2.

16. Лутц, М. Изучаем Python / М. Лутц. – Санкт-Петербург: Символ-Плюс, 2021. – 1600 с. – ISBN 978-5-6046440-6-4.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		80

Приложение А Техническое задание на программное средство

СОГЛАСОВАНО


Доцент кафедры «ПОВТиАС»

 Т.А. Медведева

« 05 » мая 2025 г.

УТВЕРЖДЕНО

Зав. кафедрой «ПОВТиАС»

 В.В. Долгов

« 05 » мая 2025 г.

А.1. Общие сведения

А.1.1 Наименование программного средства

Программное средство «EpidemicModels».

А.1.2 Область применения

Разработанное программное средство может быть использовано в медицинской сфере для моделирования эпидемиологических ситуаций, прогнозирования их развития и своевременного принятия мер в соответствии с прогнозами. Оно также может использоваться для изучения предметов, связанных с математическим моделированием и вычислительной математикой.

А.2 Основание для разработки

Разработка ведется на основании документа «Учебный план для студентов ВУЗа» направление 02.03.03 «Математическое обеспечение и администрирование информационных систем» кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» факультета «Информатика и вычислительная техника» Донского государственного технического университета.

					02.03.03.350000.000 ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

А.3 Назначение разработки

А.3.1 Функциональное назначение

Функциональное назначение программного средства заключается в прогнозировании развития эпидемиологической ситуации с помощью математических моделей.

А.3.2 Эксплуатационное назначение

Эксплуатационное назначение состоит в использовании программного средства на компьютере.

А.4 Требования к программе

А.4.1 Требования к функциональным характеристикам

Программное средство должно осуществлять следующий функционал:

- выбор представленных моделей (от 1 до 4): SIR, SI, SIQR, SEIR, MSEIR, SIRS, Multi-stage (М-модель);
- ввод соответствующих начальных значений выбранной модели;
- ввод соответствующих параметров выбранной модели;
- вызов окна теоретической справки по реализованным моделям;
- ввод реальных данных из файла формата csv;
- ввод даты начала и окончания моделирования;
- выбор численного метода для решения СДУ;
- расчет выбранных моделей с помощью численного метода;
- вывод полученных результатов в графическом виде на экран;
- экспорт результатов в табличном и графическом виде в файл формата *xlsx*.

А.4.2 Требования к надежности

Надежное функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно–технических мероприятий, перечень которых приведен ниже:

- организация бесперебойного питания технических средств;
- организация правильного использования модулей;
- защиты программного средства от несанкционированного проникновения;
- копирование (архивирование) данных, необходимых для восстановления работы программного средства;
- регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие вирусов.

А.4.2.1 Входные данные

Входными данными являются числовые значения для начальных условий и параметров моделей с помощью клавиатуры или с использованием данных из файла формата csv, выбор математических моделей, ввод начальной и конечной даты прогнозирования, выбор численного метода.

А.4.2.2 Выходные данные

Выходными данными являются результаты расчетов математических моделей в графическом виде в интерфейсе, а также в табличном и графическом виде в файле формата *xlsx*.

А.4.3 Условия эксплуатации

Для функционирования программного продукта необходимо соблюдение всех требований и правил эксплуатации компьютерной техники. Для работы с

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		83

программным средством необходимы навыки работы с персональным компьютером на уровне пользователя. Дополнительные требований и ограничений не вводится.

А.4.4 Требование к составу и параметрам технических средств

Состав технических средств пользователя:

- процессор не менее 4 ядер и 2.8 ГГц;
- оперативная память объемом не менее 8 Гб.

А.4.5 Требования к исходным кодам и языкам программирования

Программное средство должно быть реализовано на языке программирования Python, с использованием библиотек matplotlib, pandas и numpy.

А.4.7 Требования к транспортировке и хранению

Условия транспортирования, места хранения, условия складирования и сроки хранения в различных условиях должны соответствовать требованиям, предъявляемым к носителям информации, на которых будет содержаться данное программное изделие.

Допустимы все способы транспортирования и хранения, не нарушающие целостность используемого носителя данных. Программное средство может храниться на любом носителе информации, имеющее возможность подключения к персональному компьютеру.

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		84

А.5 Требования к программной документации

Программная документация должна состоять из следующих листов:

- титульный лист;
- пояснительная записка к дипломной работе;
- техническое задание по ГОСТ 15.016-2016 ЕСПД;
- исходный код программного средства по ГОСТ Р 51904-2002 ЕСПД.

А.6 Стадии и этапы разработки

Стадии и этапы разработки программного средства:

- постановка задачи (с 21.04.25 по 27.04.25);
- изучение предметной области (с 27.04.25 по 29.04.25);
- разработка алгоритма решения задачи (с 29.04.25 по 07.05.25);
- разработка программы (с 07.05.25 по 14.05.25);
- тестирование программы (с 14.05.25 по 16.05.25).
- разработка пояснительной записки к программному средству (с 17.05.2025 по 02.07.2025).

А.7 Порядок контроля и приемки

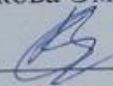
Порядок и контроль приемки определяются заведующим кафедрой «ПОВТиАС» и основаны на демонстрации знаний технологии и умении создавать программные средства для различных предметных областей.

Главным требованием к приемке является наличие правильно работающего программного средства с тестовым примером и отчета, представленного в печатном виде.

Разработчик технического задания:

/Волкова Эмилия Юрьевна/

«03» мая 2025 г.



Изм.	Лист	№ докум.	Подпись	Дата

02.03.03.350000.000 ПЗ

Лист

85

Приложение Б Исходный код программного средства

Листинг Б.1 – Исходный код программного средства

```
import tkinter as tk
from tkinter import ttk, messagebox, filedialog
from tkcalendar import DateEntry
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import numpy as np
import pandas as pd
import zipfile
from io import BytesIO

class NumericalMethods:
    def __init__(self):
        self.result = None
    def euler_method(self, model_func, y0, t, args):
        y = np.zeros((len(t), len(y0)))
        y[0] = y0
        for i in range(1, len(t)):
            dt = t[i] - t[i-1]
            dy = model_func(y[i-1], t[i-1], *args)
            y[i] = y[i-1] + dy * dt
        self.result = y.T
    def runge_kutta_4(self, model_func, y0, t, args):
        y = np.zeros((len(t), len(y0)))
        y[0] = y0
        for i in range(1, len(t)):
            dt = t[i] - t[i-1]
            h = dt
            k1 = model_func(y[i-1], t[i-1], *args)
            k2 = model_func(y[i-1] + 0.5*h*k1, t[i-1] + 0.5*h,
*args)
            k3 = model_func(y[i-1] + 0.5*h*k2, t[i-1] + 0.5*h,
*args)
            k4 = model_func(y[i-1] + h*k3, t[i-1] + h, *args)
            y[i] = y[i-1] + (h/6.0)*(k1 + 2*k2 + 2*k3 + k4)
        self.result = y.T
class EpidemicModels(NumericalMethods):
    def __init__(self):
        self.current_models = []
        self.numeric_methods = []
        self.axes = []
        self.canvases = []
        self.figs = []
    def sir_model(self, y, t, beta, gamma):
        S, I, R = y
        dSdt = -beta * S * I
```

```

        dIdt = beta * S * I - gamma * I

        dRdt = gamma * I
        return np.array([dSdt, dIdt, dRdt])
def si_model(self, y, t, beta):
    S, I = y
    dSdt = -beta * S * I
    dIdt = beta * S * I
    return np.array([dSdt, dIdt])
def sirs_model(self, y, t, beta, gamma, delta):
    S, I, R = y
    dSdt = -beta * S * I + delta * R
    dIdt = beta * S * I - gamma * I
    dRdt = gamma * I - delta * R
    return np.array([dSdt, dIdt, dRdt])
def sigr_model(self, y, t, beta, gamma, delta, mu):
    S, I, Q, R = y
    dSdt = -beta * S * I
    dIdt = beta * S * I - gamma * I - delta * I
    dQdt = delta * I - mu * Q
    dRdt = gamma * I + mu * Q
    return np.array([dSdt, dIdt, dQdt, dRdt])
def seir_model(self, y, t, beta, sigma, gamma):
    S, E, I, R = y
    dSdt = -beta * S * I
    dEdt = beta * S * I - sigma * E
    dIdt = sigma * E - gamma * I
    dRdt = gamma * I
    return np.array([dSdt, dEdt, dIdt, dRdt])
def mseir_model(self, y, t, mu, delta, beta, sigma, gamma):
    M, S, E, I, R = y
    N = M + S + E + I + R
    dMdt = mu * N - delta * M - mu * M
    dSdt = delta * M - beta * S * I / N - mu * S
    dEdt = beta * S * I / N - sigma * E - mu * E
    dIdt = sigma * E - gamma * I - mu * I
    dRdt = gamma * I - mu * R
    return np.array([dMdt, dSdt, dEdt, dIdt, dRdt])
def multi_stage_model(self, y, t, beta, k1, k2, k3, gamma):
    S, I1, I2, I3, R = y
    dSdt = -beta * S * I1 + gamma * R
    dI1dt = beta * S * I1 - k1 * I1
    dI2dt = k1 * I1 - k2 * I2
    dI3dt = k2 * I2 - k3 * I3
    dRdt = k3 * I3 - gamma * R
    return np.array([dSdt, dI1dt, dI2dt, dI3dt, dRdt])
class EpidemicApp:
    def __init__(self, root):
        self.root = root
        self.root.title("EpidemicModels")

```

```

self.root.geometry("1200x800")
self.root.minsize(1000, 700)
self.style = ttk.Style()

self.style.configure('TFrame', background='#f0f0f0')
self.style.configure('TLabel', background='#f0f0f0',
font=('Arial', 10))
self.style.configure('TButton', font=('Arial', 10))
self.style.configure('TCheckbutton', background='#f0f0f0',
font=('Arial', 10))
self.style.configure('TRadiobutton', background='#f0f0f0',
font=('Arial', 10))
self.model = EpidemicModels()
self.result_data = {}
self.country_population = {}
self.create_widgets()
def create_widgets(self):
    main_paned = ttk.PanedWindow(self.root,
orient=tk.HORIZONTAL)
    main_paned.pack(fill=tk.BOTH, expand=True)
    control_frame = ttk.Frame(main_paned, width=350,
relief=tk.RIDGE, padding=10)
    main_paned.add(control_frame, weight=0)
    graph_frame = ttk.Frame(main_paned)
    main_paned.add(graph_frame, weight=1)
    control_notebook = ttk.Notebook(control_frame)
    control_notebook.pack(fill=tk.BOTH, expand=True)
    models_tab = ttk.Frame(control_notebook)
    control_notebook.add(models_tab, text="Модели")
    self.create_models_tab(models_tab)
    params_tab = ttk.Frame(control_notebook)
    control_notebook.add(params_tab, text="Параметры")
    self.create_params_tab(params_tab)
    data_tab = ttk.Frame(control_notebook)
    control_notebook.add(data_tab, text="Данные")
    self.create_data_tab(data_tab)
    self.create_graphs(graph_frame)
def create_model_params_tab(self, model_code):
    frame = ttk.Frame(self.params_notebook)
    validate_num =
self.root.register(self.create_validate_func(0, 1))
    param_group = ttk.LabelFrame(frame, text="Параметры
модели", padding=10)
    param_group.pack(fill=tk.X, pady=5)
    init_group = ttk.LabelFrame(frame, text="Начальные
значения", padding=10)
    init_group.pack(fill=tk.X, pady=5)
    param_entries = {}
    init_entries = {}
    param_definitions = {"SI": [("beta", "β"), "SIR":
[("beta", "β"), ("gamma", "γ"), "SIRS": [("beta", "β"), ("gamma",

```

```

"γ"), ("delta", "δ")], "SEIR": [("beta", "β"), ("sigma", "σ"),
("gamma", "γ")], "SIQR": [("beta", "β"), ("gamma", "γ"), ("delta",
"δ"), ("mu", "μ")], "MSEIR": [("mu", "μ"), ("delta", "δ"), ("beta",
"β"), ("sigma", "σ"), ("gamma", "γ")], "M": [("beta", "β"), ("k1",
"k1"), ("k2", "k2"), ("k3", "k3"), ("gamma", "γ")]
    init_definitions = {"SI": [("S0", "S0"), ("I0",
"I0")], "SIR": [("S0", "S0"), ("I0", "I0"), ("R0", "R0")], "SIRS":
[("S0", "S0"), ("I0", "I0"), ("R0", "R0")], "SEIR": [("S0", "S0"),
("E0", "E0"), ("I0", "I0"), ("R0", "R0")], "SIQR": [("S0", "S0"),
("I0", "I0"), ("Q0", "Q0"), ("R0", "R0")], "MSEIR": [("M0", "M0"),
("S0", "S0"), ("E0", "E0"), ("I0", "I0"), ("R0", "R0")], "M":
[("S0", "S0"), ("I10", "I10"), ("I20", "I20"), ("I30", "I30"),
("R0", "R0")]}
    for code, label in param_definitions.get(model_code, []):
        row = ttk.Frame(param_group)
        row.pack(fill=tk.X, pady=2)
        ttk.Label(row, text=label,
width=25).pack(side=tk.LEFT)
        entry = ttk.Entry(row, width=8, validate="key",
validatecommand=(validate_num, '%P'))
        entry.pack(side=tk.RIGHT)
        param_entries[code] = entry
    for code, label in init_definitions.get(model_code, []):
        row = ttk.Frame(init_group)
        row.pack(fill=tk.X, pady=2)
        ttk.Label(row, text=label,
width=25).pack(side=tk.LEFT)
        entry = ttk.Entry(row, width=8, validate="key",
validatecommand=(validate_num, '%P'))
        entry.pack(side=tk.RIGHT)
        init_entries[code] = entry
    def on_entry_change(*args):
        if not self.validate_sum(init_entries):
            messagebox.showwarning("Ошибка", "Сумма начальных
значений не должна превышать 1")
        for entry in param_entries.values():
            entry.bind("<FocusOut>", on_entry_change)
        for entry in init_entries.values():
            entry.bind("<FocusOut>", on_entry_change)
        self.model_param_tabs[model_code] = {"frame":
frame, "param_entries": param_entries, "init_entries": init_entries}
        self.params_notebook.add(frame, text=model_code)
    def create_models_tab(self, parent):
        models_group = ttk.LabelFrame(parent, text="Выберите
модели (макс. 4)", padding=10)
        models_group.pack(fill=tk.BOTH, pady=5)
        self.model_vars = {}
        models = [('SI', 'Модель SI (восприимчивые-
инфицированные)'), ('SIR', 'Модель SIR (восприимчивые-

```

```

инфицированные-выздоровевшие)'), ('SIRS', 'Модель SIRS (с временным
иммунитетом)'), ('SEIR', 'Модель SEIR (с латентным
периодом)'), ('SIQR', 'Модель SIQR (с изоляцией)'), ('MSEIR',

'Mодель MSEIR (с материнским иммунитетом)'), ('M', 'М-модель (3
стадии инфекции)') ]
    for model_code, model_desc in models:
        var = tk.BooleanVar()
        cb = ttk.Checkbutton(models_group, text=model_desc,
variable=var, command=self.update_model_selection)
        cb.pack(anchor='w', padx=5, pady=2)
        self.model_vars[model_code] = var
    method_group = ttk.LabelFrame(parent, text="Метод
решения", padding=10)
    method_group.pack(fill=tk.BOTH, pady=5)
    self.method_var = tk.StringVar(value="runge_kutta")
    ttk.Radiobutton(method_group, text="Рунге-Кутта 4-го
порядка", variable=self.method_var,
value="runge_kutta").pack(anchor='w', padx=5, pady=2)
    ttk.Radiobutton(method_group, text="Метод Эйлера",
variable=self.method_var, value="euler").pack(anchor='w', padx=5,
pady=2)
    ttk.Button(parent, text="Запустить моделирование",
command=self.run_simulation).pack(fill=tk.X, pady=10)
    ttk.Button(parent, text="Справка по моделям",
command=self.open_model_docs).pack(fill=tk.X, pady=5)
    def create_params_tab(self, parent):
        canvas = tk.Canvas(parent)
        scrollbar = ttk.Scrollbar(parent, orient="vertical",
command=canvas.yview)
        scrollable_frame = ttk.Frame(canvas)
        scrollable_frame.bind("<Configure>", lambda e:
canvas.configure(scrollregion=canvas.bbox("all")))
        canvas.create_window((0, 0), window=scrollable_frame,
anchor="nw")
        canvas.configure(yscrollcommand=scrollbar.set)
        canvas.pack(side="left", fill="both", expand=True)
        scrollbar.pack(side="right", fill="y")
        self.params_notebook = ttk.Notebook(scrollable_frame)
        self.params_notebook.pack(fill=tk.BOTH, expand=True)
        self.model_param_tabs = {}
        time_group = ttk.LabelFrame(scrollable_frame,
text="Временной диапазон", padding=10)
        time_group.pack(fill=tk.X, pady=5)
        ttk.Label(time_group, text="Начальная
дата:").pack(anchor='w', pady=(0, 5))
        self.start_entry = DateEntry(time_group,
date_pattern='dd.mm.yyyy')
        self.start_entry.pack(fill=tk.X, pady=(0, 10))

```

```

        ttk.Label(time_group, text="Конечная
дата:").pack(anchor='w', pady=(0, 5))
        self.end_entry = DateEntry(time_group,
date_pattern='dd.mm.yyyy')
        self.end_entry.pack(fill=tk.X)
        def create_data_tab(self, parent):
            load_group = ttk.LabelFrame(parent, text="Загрузка
данных", padding=10)
            load_group.pack(fill=tk.BOTH, pady=5, expand=True)
            ttk.Button(load_group, text="Загрузить данные из CSV",
command=self.load_csv_data).pack(fill=tk.X, pady=5)
            export_group = ttk.LabelFrame(parent, text="Экспорт
результатов", padding=10)
            export_group.pack(fill=tk.BOTH, pady=5)
            ttk.Button(export_group, text="Экспорт в Excel и ZIP",
command=self.export_results).pack(fill=tk.X, pady=5)
        def create_graphs(self, parent):
            self.model.axes = []
            self.model.canvases = []
            self.model.figs = []
            for i in range(4):
                fig = plt.Figure(figsize=(6, 4), dpi=100)
                ax = fig.add_subplot(111)
                ax.grid(True, linestyle='--', alpha=0.7)
                ax.set_facecolor('#f8f8f8')
                canvas = FigureCanvasTkAgg(fig, master=parent)
                canvas_widget = canvas.get_tk_widget()
                canvas_widget.grid(row=i//2, column=i%2, padx=5,
pady=5, sticky='nsew')
                canvas_widget.config(borderwidth=2, relief=tk.GROOVE)
                self.model.figs.append(fig)
                self.model.axes.append(ax)
                self.model.canvases.append(canvas)
        def process_csv_data(self):
            country = self.country_cb.get()
            if not country:
                messagebox.showwarning("Ошибка", "Выберите страну")
                return
            try:
                start_date = self.data_start_entry.get_date()
                end_date = self.data_end_entry.get_date()
                if start_date > end_date:
                    messagebox.showerror("Ошибка", "Начальная дата не
может быть позже конечной")
                    return
                df_country = self.csv_data[
                    (self.csv_data["Country/Region"] == country) &
                    (self.csv_data["Date"] >=
pd.to_datetime(start_date)) &
                    (self.csv_data["Date"] <=
pd.to_datetime(end_date))]

```



```

].copy()
if df_country.empty:
    messagebox.showerror("Ошибка", "Нет данных для
выбранного диапазона дат")
    return
latest = df_country.iloc[-1]

country = self.country_cb.get()
total_population =
self.country_population.get(country, 1)
if total_population <= 0:
    total_population = 1
S0 = (total_population - latest["Confirmed"] -
latest["Recovered"] - latest["Deaths"]) / total_population
I0 = latest["Confirmed"] / total_population
R0 = (latest["Recovered"] + latest["Deaths"]) /
total_population
for tab in self.model_param_tabs.values():
    init_entries = tab["init_entries"]
    for key in ("S0", "I0", "R0"):
        if key in init_entries:
            init_entries[key].delete(0, tk.END)
        if "S0" in init_entries:
            init_entries["S0"].insert(0, f"{S0:.4f}")
        if "I0" in init_entries:
            init_entries["I0"].insert(0, f"{I0:.4f}")
        if "R0" in init_entries:
            init_entries["R0"].insert(0, f"{R0:.4f}")
    self.start_entry.set_date(end_date)
    self.end_entry.set_date(end_date +
timedelta(days=100))
    messagebox.showinfo("Успех", f"Данные для {country} за
период {start_date.strftime('%d.%m.%Y')} -
{end_date.strftime('%d.%m.%Y')} успешно загружены")
    except Exception as e:
        messagebox.showerror("Ошибка", f"Не удалось обработать
данные: {str(e)}")
    try:
        df_country["Infected"] = df_country["Confirmed"] -
df_country["Recovered"] - df_country["Deaths"]
        df_country["Removed"] = df_country["Recovered"] +
df_country["Deaths"]
        df_country["Susceptible"] = (total_population -
df_country["Infected"] - df_country["Removed"]) / total_population
        I = df_country["Infected"].values / total_population
        R = df_country["Removed"].values / total_population
        S = df_country["Susceptible"].values
        beta_list = []
        gamma_list = []
        for i in range(1, len(df_country)):
            I_prev = I[i - 1]

```

```

        I_curr = I[i]
        R_prev = R[i - 1]
        R_curr = R[i]
        S_prev = S[i - 1]
        dI = I_curr - I_prev
        dR = R_curr - R_prev
        gamma_i = None

        if I_prev > 1e-6:
            gamma_i = dR / I_prev
            if np.isfinite(gamma_i):
                gamma_list.append(gamma_i)
        if I_prev > 1e-6 and S_prev * I_prev > 1e-6:
            gamma_eff = gamma_i if gamma_i is not None
    else 0.1
        beta_i = (dI + gamma_eff * I_prev) / (S_prev *
I_prev)
        if np.isfinite(beta_i):
            beta_list.append(beta_i)
        gamma = np.clip(np.mean(gamma_list), 0.01, 1.0) if
gamma_list else 0.1
        beta = np.clip(np.mean(beta_list), 0.01, 1.0) if
beta_list else 0.3
        for tab in self.model_param_tabs.values():
            param_entries = tab["param_entries"]
            if "beta" in param_entries:
                param_entries["beta"].delete(0, tk.END)
                param_entries["beta"].insert(0, f"{beta:.4f}")
            if "gamma" in param_entries:
                param_entries["gamma"].delete(0, tk.END)
                param_entries["gamma"].insert(0,
f"{gamma:.4f}")
        except Exception as e:
            print("Не удалось вычислить параметры:", e)
    def export_results(self):
        if not self.result_data:
            messagebox.showwarning("Нет данных", "Сначала
выполните моделирование")
            return
        save_path =
filedialog.asksaveasfilename(defaultextension=".zip", filetypes=[("
ZIP files", "*.zip")], initialfile="epidemic_results.zip")
        if not save_path:
            return
        try:
            mem_zip = BytesIO()
            with zipfile.ZipFile(mem_zip, mode='w',
compression=zipfile.ZIP_DEFLATED) as zf:
                for model_name, df in self.result_data.items():
                    excel_buffer = BytesIO()

```

```

        with pd.ExcelWriter(excel_buffer,
engine='xlsxwriter') as writer:
            model_tab =
self.model_param_tabs.get(model_name)
            if not model_tab:
                continue
            initials_data = {}
            for key, entry in
model_tab["init_entries"].items():

                initials_data[key] = entry.get()
                initials_df = pd.DataFrame({'Параметр':
['S0 (восприимчивые)', 'I0 (инфицированные)', 'R0
(выздоровевшие)', 'E0 (латентные)', 'Q0 (изолированные)', 'M0
(материнский иммунитет)'], 'Значение': [initials_data.get("S0",
""), initials_data.get("I0", ""), initials_data.get("R0",
""), initials_data.get("E0", ""), initials_data.get("Q0",
""), initials_data.get("M0", "")]})
                initials_df.to_excel(writer,
sheet_name='Начальные данные', index=False)
                params_data = {}
                for key, entry in
model_tab["param_entries"].items():
                    params_data[key] = entry.get()
                    params_df = pd.DataFrame({'Параметр': ['β
(скорость заражения)', 'γ (скорость выздоровления)', 'δ (потеря
иммунитета)', 'σ (переход в инфекционные)', 'μ (выход из
изоляции)'], 'Значение': [params_data.get("beta",
""), params_data.get("gamma", ""), params_data.get("delta",
""), params_data.get("sigma", ""), params_data.get("mu", "")]})
                    params_df.to_excel(writer,
sheet_name='Параметры', index=False)
                    method_info = pd.DataFrame({'Информация':
['Метод решения:', 'Выбранная модель:'], 'Значение':
[self.method_var.get().replace("runge_kutta", "Рунге-Кутта 4-го
порядка").replace("euler", "Метод Эйлера"), model_name]})
                    method_info.to_excel(writer,
sheet_name='Решение', startrow=0, index=False, header=False)
                    df.to_excel(writer, sheet_name='Решение',
startrow=3, index=True)
                    worksheet = writer.sheets['Решение']
                    worksheet.write(3, 0, 'Дни')
                    graph_sheet =
writer.book.add_worksheet('График')
                    chart = writer.book.add_chart({'type':
'line'})

                    max_row = len(df) + 4
                    categories =
f"='Решение'!$A$5:$A${max_row}"
                    for i, col in enumerate(df.columns, 1):
                        col_letter = chr(65 + i)

```

```

        chart.add_series({'name':
f"='Решение'!${col_letter}$4",'categories': categories,'values':
f"='Решение'!${col_letter}$5:${col_letter}${max_row}",})
        chart.set_x_axis({'name': 'Дни'})
        chart.set_y_axis({'name': 'Доля
населения'})

        chart.set_title({'name': f'Модель
{model_name} ({method_info.iloc[0,1]})'})
        graph_sheet.insert_chart('B2', chart,
{'x_scale': 2, 'y_scale': 2})
        zf.writestr(f"{model_name}.xlsx",
excel_buffer.getvalue())
        with open(save_path, "wb") as f:
            f.write(mem_zip.getvalue())
        messagebox.showinfo("Экспорт завершён", f"Файлы
успешно сохранены в архив:\n{save_path}")
    except Exception as e:
        messagebox.showerror("Ошибка", f"Не удалось
экспортировать данные: {str(e)}")
if __name__ == "__main__":
    root = tk.Tk()
    app = EpidemicApp(root)
    root.mainloop()

```

					02.03.03.350000.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		95

Приложение В. Протокол предзащиты

Зав. кафедрой

«ПОВТиАС»

В.В. Долгов

«03» июня 2025 г.

ПРОТОКОЛ заседания комиссии по предварительной защите выпускной квалификационной работы от «03» июня 2025 г.

ПРИСУТСТВОВАЛИ:

доцент

Т.А. Медведева

доцент

А.В. Криворучко

доцент

С.А. Швидченко

СЛУШАЛИ: доклад обучающегося группы ВМО41 Волковой Э.Ю., по защите выпускной квалификационной работы «Программная реализация математического моделирования развития эпидемиологической ситуации».

ЗАДАННЫЕ ВОПРОСЫ:


1. Где может применяться ваша программа на практике?
2. Как можно доработать проект в будущем?
- 3.
- 4.

РЕШЕНО ОТМЕТИТЬ СЛЕДУЮЩЕЕ:

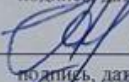
1. Увеличить размер схем в презентации
2. Изменить формулировку цели задачи
- 3.
- 4.

ПОСТАНОВИЛИ: доклад обучающегося о проделанной работе и ответы на вопросы соответствуют требованиям, предъявляемым к выпускным квалификационным работам. Обучающийся Волкова Э.Ю. рекомендован к защите.


Члены комиссии:

 03.06.25

Т.А. Медведева

 03.06.25

А.В. Криворучко

 03.06.25

С.А. Швидченко