|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  ОРСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)  ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  Факультет среднего профессионального образования | |
| **Курсовая работа**  по междисциплинарному курсу «Технология разработки программного обеспечения»  профессионального модуля «Осуществление интеграции программных модулей»    **Разработка программного обеспечения для исследование компьютерной модели искусственной жизни «хищник-жертва»**  Пояснительная записка  ОГУ 09.02.07. 3024. 895 ПЗ | |
|  | Руководитель работы  преподаватель высшей категории  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ж. В. Михайличенко  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.  Студент группы 21ИСП-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. В. Криворотов  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
| Орск 2024 | |

|  |  |
| --- | --- |
| Утверждаю  председатель ПЦК дисциплин профессионального цикла | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись | Ж.В. Михайличенко |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | |

**ЗАДАНИЕ**

**на выполнение курсовой работы**

студенту \_\_\_\_\_Криворотову Евгению Владимировичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество

по специальности \_\_\_09.02.07 Информационные системы и программирование\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по междисциплинарному курсу \_\_Технология разработки программного обеспечения\_\_\_\_\_\_\_\_\_\_

1. Тема работы: \_\_\_Разработка программного обеспечения для исследования компьютерной модели искусственной жизни «хищник-жертва»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Срок сдачи студентом работы «10» \_июня\_\_\_ 2024 г.
3. Цель и задачи работы \_\_Разработать программный продукт для реализации искусственной жизни, развивающейся по заданным правилам и проанализировать её динамику\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Исходные данные к работе: \_\_Учебники и интернет-источники по технологии разработки программного обеспечения и компьютерному моделированию, включающие описание биологических моделей развития популяций \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Перечень вопросов, подлежащих разработке: \_ а) изучить предметную область, выполнить анализ требований к программному обеспечению, составить техническое задание на разработку; б) выполнить проектирование системы с помощью CASE-средств; в) для решения поставленной задачи реализовать оконное приложение на языке C# и протестировать его; г) сформулировать предложения по внедрению, эксплуатации и сопровождению разработанного программного обеспечения. Сделать выводы по результатам проделанной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Перечень графического (иллюстративного) материала: таблицы, графики, рисунки, схемы, отражающие теоретический материал и программную реализацию поставленной задачи\_\_\_\_\_\_\_

Дата выдачи и получения задания

Руководитель «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Ж.В. Михайличенко\_\_\_

подпись инициалы, фамилия

Студент «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Е.В. Криворотов\_\_\_\_\_\_

подпись инициалы, фамилия

**Аннотация**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

ОГУ 09.02.07. 3024. 895 ПЗ

Разраб.

Криворотов Е.В.

Провер.

Михайличенко Ж

Реценз.

Н. Контр.

Утверд.

Разработка программного обеспечения для исследование компьютерной модели искусственной жизни «хищник-жертва»

Лит.

Листов

\*

22ИСП-1

В курсовой работе по междисциплинарному курсу «Технология разработки программного обеспечения» профессионального модуля «Осуществление интеграции программных модулей» проведена разработка \*\*\*\*\*.

В первой главе курсовой работы \*\*\*

Во второй главе курсовой работе \*\*\*

В третьей главе курсовой работы \*\*\*

Пояснительная записка содержит \*\* страницы, в том числе \*\* рисунков, \*\* таблиц, \*\* источников, 1 приложение.

Разработка приложения выполнена \*\*\*.

**Содержание**

[Введение 5](#_Toc168490349)

[1 Анализ требований и проектирования программного продукта 6](#_Toc168490350)

[1.1 Анализ предметной области 6](#_Toc168490351)

[1.2 Техническое задание на разработку 7](#_Toc168490352)

[1.3 Построение модели 10](#_Toc168490353)

[2 Реализация и тестирование программного продукта 13](#_Toc168490354)

[2.1 Обоснование программных средств реализации 13](#_Toc168490355)

[2.2 Разработка пользовательского интерфейса 14](#_Toc168490356)

[2.3 Алгоритмизация и программирование 15](#_Toc168490357)

[2.4 Тестирование 17](#_Toc168490358)

[3 Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения 18](#_Toc168490359)

[3.1 Руководство пользователя 18](#_Toc168490360)

[3.2 План внедрения и сопровождения 21](#_Toc168490361)

[Заключение 22](#_Toc168490362)

[Список использованных источников 23](#_Toc168490363)

[Приложение А 24](#_Toc168490364)

# Введение

Моделирование систем по типу «хищник-жертва» является ключевым элементом в изучении экологических систем и динамики популяций. Разработка программного обеспечения для исследования таких моделей имеет большую практическую применимость, помогая ученым лучше понять и предсказывать изменения в экосистемах.

Целью данной курсовой работы является разработка программного обеспечения для исследования компьютерной модели искусственной жизни «хищник-жертва». Это ПО будет использоваться для моделирования взаимодействий между хищниками и их жертвами в различных условиях, чтобы изучить динамику популяций и влияние различных факторов на выживаемость видов.

Для достижения этой цели будут решены следующие задачи:

1. Изучение существующих моделей искусственной жизни и систем «хищник-жертва»;
2. Разработка алгоритма для моделирования взаимодействий между хищниками и жертвами;
3. Реализация алгоритма в программном обеспечении, включая создание пользовательского интерфейса для визуализации модели;
4. Тестирование программного обеспечения на различных сценариях и анализ результатов.

В центре исследования находится компьютерная модель, которая воссоздает взаимодействия между хищниками и жертвами в условиях искусственной жизни. Наша основная задача – создание программного обеспечения, которое позволит нам глубоко исследовать эту модель.

Мы будем использовать различные методы исследования, включая анализ для изучения отдельных элементов модели, синтез для объединения этих элементов в целостную систему, сравнение для определения сходств и различий между разными моделями, и обобщение для выявления общих закономерностей и принципов.

# Анализ требований и проектирования программного продукта

## Анализ предметной области

Тема курсовой работы относится к области компьютерного экологического моделирования.

Компьютерное моделирование даёт инструменты и методы для создания детализированных и точных моделей реального мира в цифровой форме. Это может включать в себя всё, от простых математических уравнений до сложных трехмерных визуализаций. Эти модели могут быть использованы для проведения экспериментов, которые могут быть сложными, дорогостоящими или неосуществимыми с имеющимися ресурсами в условиях реального мира.

Экологическое моделирование уже использует эти инструменты для создания моделей экологических систем, таких как популяции животных, растительные сообщества или целые экосистемы. Эти модели могут быть использованы для изучения взаимодействий между различными видами, понимания влияния человека на окружающую среду или прогнозирования изменений в экосистемах на воздействия. В экологическом моделировании можно выделить два основных направления: моделирование взаимодействия организмов друг с другом и с окружающей средой («классическая» экология); моделирование, связанное с состоянием и охраной окружающей среды (социальная экология).

Построение модели базируется на абстрагировании от конкретной природы явлений или изучаемого объекта-оригинала и состоит из двух этапов – сначала создание качественной, а затем и количественной модели. Чем больше значимых свойств будет выявлено и перенесено на компьютерную модель, тем более приближенной она окажется к реальной модели, тем большими возможностями сможет обладать система, использующая данную модель.

Модель «хищник-жертва» важна для понимания динамики взаимодействия между двумя видами, где один является хищником, а другой - его жертвой. Она используется при решении различных экономических задач, в том числе при определении оптимального уровня государственного сектора экономики. Модель позволяет изучать циклические и иррегулярные колебания структуры экономики в динамике, а также прогнозировать структурные изменения в экономике. В биологии эта модель помогает понять, как взаимодействуют популяции хищников и их жертв и является одной из наиболее точных.

Моделирование такого рода систем может быть сложным и требует глубокого понимания как математических, так и экологических принципов. Например, малейшие изменения в модели могут привести к качественному изменению поведения системы.

Программная модель «хищник-жертва», разрабатываемая в курсовой работе, разворачивается на ограниченном клеточном поле, внутри которого сущности жертв способны передвигаться и при возможности размножаться, а сущности хищников ищут жертв, поедают их, и впоследствии их популяция тоже множится. Пользователь может задавать начальные параметры модели: количеств сущностей каждого вида, размер клеточного поля и, при необходимости, ограничение времени. Модель проигрывается в течение ограниченного времени, а все изменения пользователем наблюдаются либо на самом поле, либо на графиках популяций. Модель проигрывается в течение ограниченного времени, а все изменения пользователем наблюдаются либо самом поле, либо на графике популяций.

## Техническое задание на разработку

Техническое задание на разработку программного обеспечения – это документ, который содержит подробное описание требований к программному продукту. Техническое задание на разработку программного обеспечения для исследования компьютерной модели искусственной жизни «хищник-жертва» составлено согласно ГОСТ 34.602–2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» и представлено ниже.

1 Общие сведения

а) Наименование автоматизированной системы (АС): «Автоматизированная система модели искусственной жизни «хищник-жертва»;

б) Наименование заказчика: факультет среднего профессионального образования Орского гуманитарно-технологического института (филиала) ОГУ в лице преподавателя высшей категории Михайличенко Ж. В.;

в) Наименование разработчика: студент второго курса группы 22ИСП-1 Криворотов Е. В.;

г) Документ на основании, которого создается АС: протокол закрепления тем курсовой работы по дисциплине «технология разработки программного обеспечения» от 19.02.2024;

д) Дата начала работ: 19.02.2024;

е) Дата окончания работ: 10.06.2024.

2 Цели и назначение создания АС.

Цели создания АС: облегчение исследования модели взаимодействия хищников и жертв в природе и влияния их популяции друг на друга.

Назначение АС: визуализации поведения модели искусственной жизни и подбор условий для получения успешных результатов в исследовании модели.

3 Характеристика объекта автоматизации.

Модель искусственной жизни «хищник-жертва» представляет собой динамическую модель, которая описывает взаимодействие двух видов – «хищник» и «жертва». Длительное совместное существование хищников и жертв приводит к формированию системы взаимодействия, при которой обе группы устойчиво сохраняются на изучаемой территории.

Объектом автоматизации являются процессы моделирования:

* Создание сущностей;
* Взаимодействия между сущностями;
* Анализ данных.

1. Требования к АС.

а) Требования к функциям, выполняемым АС:

* Создание виртуального мира;
* Создание сущностей;
* Реализация взаимодействия между сущностями;
* Визуализация данных;
* Изменение сценариев поведения модели;
* Получение и анализ данных.

б) Требования к видам обеспечения АС:

* Требования к математическому обеспечению: использование модели Лотки-Вольтерры, уравнения Ферхюльтса.
* Требования к информационному обеспечению: ввод исходных данных в диалоге с пользователем, визуализация на экране монитора, вывод результатов моделирования на экран монитора.
* Требования к лингвистическому обеспечению: использование русского языка в пользовательском интерфейсе, диалог с пользователем через графический интерфейс;
* Требования к программному обеспечению: использование операционной системы Windows, программирование на языке C# с использованием интегрированной среды разработки Microsoft Visual Studio 2019, Microsoft Word для документирования, Ramus;
* Требования к техническому обеспечению: наличие мыши или клавиатуры, свободное место на жестком диске не менее 50 МБ, объем оперативной памяти не менее 64 МБ, разрешение экрана не менее 800x600;
* Требования к организационному обеспечению: взаимодействие пользователя с системой организовано в виде ввода исходных данных, наблюдением за динамикой искусственной жизни, получения и сохранения результатов моделирования.

в) Общие технические требования к АС:

* Требования к численности и квалификации пользователей: система предназначена для одного пользователя, имеющего базовые навыки работы с компьютером;
* Требования к эргономике и технической эстетике: система должны иметь интуитивно понятный интерфейс с привлекающими внимание цветами
* Требования к защите информации: написать

5 Состав и содержание работ по созданию АС.

В таблице 1 показаны этапы разработки «Автоматизированной система модели искусственной жизни «хищник-жертва».

Таблица 1 – Этапы разработки АС

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 1 Анализ требований | 19.02.2024 – 05.03.2024 | Анализ предметной области, изучение программных аналогов, разработка технического задания на создание АС |
| 2 Проектирование | Даты? | Построение архитектуры системы, структуры хранения данных и интерфейса |
| 3 Программирование | Даты? | Реализация алгоритмов, разработка программного кода, создание технической документации |
| 4 Тестирование | Даты? | Проверка соответствия результатов, выдаваемых разработанным ПО на соответствие желаемым результатам |
| 5 Внедрение | Даты? | Установка разработанной системы, обучение работе над ней и документирование |
| 6 Эксплуатация и сопровождение со стороны разработчика | Даты? | Внесение изменений в ПО в соответствии с документацией. |

6 Порядок разработки АС.

Этапы будут выполняться разработчиком в прямой последовательности в соответствии с приведенной в пункте пять таблицей этапов разработки АС с обязательным контролем и консультацией с заказчиком.

7 Порядок контроля и приемки АС.

Контроль за разработкой осуществляется заказчиком и принимается в указанные сроки – 10.06.2024.

8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу АС в действие.

Программный продукт не должен требовать специальной инсталляции, а мог бы переноситься на компьютер пользователя путём копирования. Обучение пользователя работе с системой не требуется.

9 Требования к документированию.

* Документирование программного кода (комментарии);
* Техническое задание;
* Пояснительная записка к курсовой работе;
* План внедрения и сопровождения;
* Руководство пользователя.

10 Источники разработки.

* Протокол закрепления тем курсовой работы по дисциплине «технология разработки программного обеспечения» от 19.02.2024;
* ГОСТ 34.602–2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»;
* Работы студенческие. Общие требования и правила оформления. СТО 02069024.101–2015. – Оренбург: Изд-во ОГУ, 2015. – 89 с.
* Добавить ГОСТ на блок-схемы
* Добавить ГОСТ на стадии разработки АС

## Построение модели

Проектирование информационной системы (ИС) играет ключевую роль в её жизненном цикле. Оно позволяет определить цели и требования к системе, спроектировать её архитектуру, функциональность и взаимодействие с пользователями. Функциональная модель является важным инструментом проектирования, позволяющим описать бизнес-процессы и взаимодействие компонентов системы, обеспечивая понимание и согласование между заказчиками и разработчиками.

IDEF0 («Icam DEFinition for Function Modeling») — это методология моделирования функций, предназначенная для описания производственных функций. Она предоставляет язык функционального моделирования для анализа, разработки, реинжиниринга и интеграции информационных систем, бизнес-процессов или анализа программной инженерии.

IDEF0 позволяет подробно описать решения, действия и деятельность организации или системы. Он был разработан на основе языка графического моделирования Structured Analysis and Design Technique (SADT) и помогает анализировать и коммуницировать функциональную перспективу системы. Методология обеспечивает точное описание и способствует согласованности использования и интерпретации. IDEF0 способен графически представить разнообразные бизнес-процессы, производство и другие виды операций предприятия на любом уровне детализации. Существует множество коммерческих продуктов, специально поддерживающих разработку и анализ диаграмм и моделей IDEF0. IDEF0 также может быть дополнен методом Integration Definition for Information Modeling (IDEF1x) для данных-интенсивных систем.

Контекстная модель в формате IDEF0 для проекта «Программное обеспечение для исследования искусственной жизни» описывает работу компьютерной модели «хищник-жертва» (A0). Входные данные для этой модели включают размер поля, количество объектов на поле и максимальное время работы модели. Эти данные преобразуются в итоговое время работы модели, график изменения популяции объектов и визуальное представление взаимодействия объектов. Процесс управляется с помощью руководства пользователя и правил жизни. Управление осуществляется компьютером и пользователем. Модель помогает понять, как работает система на высоком уровне.

Контекстная диаграмма IDEF0 модели показана на рисунке 1.

Изображение выглядит как текст, снимок экрана, Параллельный, линия

Автоматически созданное описание

Рисунок 1 – Контекстная диаграмма IDEF0

Для того, чтобы лучше понять функционирование системы, необходимо определить декомпозицию. Декомпозиция первого уровня выглядит следующим образом:

Блок A1 – «Получение данных от пользователя» принимает на вход размер поля, количество объектов на поле и максимальное время работы модели. Эти данные управляются с помощью руководства пользователя и исполняются пользователем и компьютером, что приводит к формированию начальных условий.

Блок A2 – «Воспроизведение работы модели» принимает на вход начальные условия. Под управлением правил жизни с помощью компьютера происходит воспроизведение работы модели, что приводит к формированию визуального представления взаимодействия объектов и информации о состоянии модели.

Блок A3 – «Формирование и вывод результатов» принимает на вход информацию о состоянии модели. Компьютером формируются и выводятся результаты, что приводит к построению графика изменения популяции объектов и вывода итогового времени работы модели.

Диаграмма IDEF0 декомпозиции первого уровня системы показана на рисунке 2.

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Автоматически созданное описание

Рисунок 2 – Диаграмма декомпозиции первого уровня IDEF0

Все построенные диаграммы будут использованы для разработки программного обеспечения, моделирующего искусственную жизнь «Хищник-жертва».

# Реализация и тестирование программного продукта

## Обоснование программных средств реализации

Для реализации автоматизированной системы модели искусственной жизни «хищник-жертва» решено использовать язык программирования C# и интегрированную среду разработки Visual Studio 2022, тип проекта - Windows Forms на платформе .NET Framework.

C# – объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java.

C# обладает следующими преимуществами:

* полностью поддерживает парадигму объектно-ориентированного программирования, что делает его гибким и мощным инструментом для разработки;
* является типизированным языком, что увеличивает надёжность и производительность кода;
* автоматически освобождает память, занимаемую неиспользуемыми объектами, что упрощает управление памятью;
* благодаря широким возможностям .NET, C# может использоваться для разработки приложений, работающих на различных платформах.

Visual Studio – это мощная среда разработки, которую можно использовать для написания, редактирования, отладки и сборки кода, а затем для развертывания приложения. Visual Studio предлагает следующие преимущества:

* предлагает все необходимые инструменты для создания ваших приложений в одном месте;
* можно писать код на C++, C#, JavaScript, TypeScript, Python и многих других языках;
* можно создавать приложения для любой платформы;
* позволяет эффективно работать над кодом в команде;
* позволяет добавлять дополнительные расширения, которых нет в других редакторах кода.

Вместе C# и Visual Studio предоставляют мощные и гибкие инструменты для разработки программного обеспечения. Учитывая все вышеперечисленные преимущества языка программирования C# и интегрированной среды разработки Visual Studio, выбор данного программного обеспечения вполне обоснован для разработки в рамках курсовой работы автоматизированной системы модели искусственной жизни «хищник-жертва».

## Разработка пользовательского интерфейса

Программный интерфейс – это набор протоколов и спецификаций, которые позволяют различным программным компонентам обмениваться данными и взаимодействовать друг с другом. Программные интерфейсы обеспечивают точки взаимодействия между аппаратными и программными компонентами. Они помогают всем компонентам внутри системы общаться друг с другом через систему ввода-вывода и подробные протоколы, позволяя им функционировать независимо. Программные интерфейсы также помогают пользователям взаимодействовать с различными типами устройств через аппаратные устройства, такие как клавиатуры, мыши и сенсорные экраны.

Пользовательский интерфейс должен быть доступным, последовательным, простым в использовании, интуитивно понятным и адаптивным. Все функции и информация должны быть легко доступны, а элементы интерфейса должны быть последовательными во всех частях приложения. Интерфейс должен быть простым и минималистичным, чтобы пользователи могли легко понять, как использовать приложение. Он должен быть интуитивно понятным, чтобы новые пользователи могли быстро освоиться. Это обеспечивает широкую доступность и удобство использования приложения.

Интерфейс программного обеспечения для исследования компьютерной модели искусственной жизни «хищник-жертва» должен соответствовать всем вышеперечисленным требованиям. Структура интерфейса и его компоненты показаны на рисунке 3.

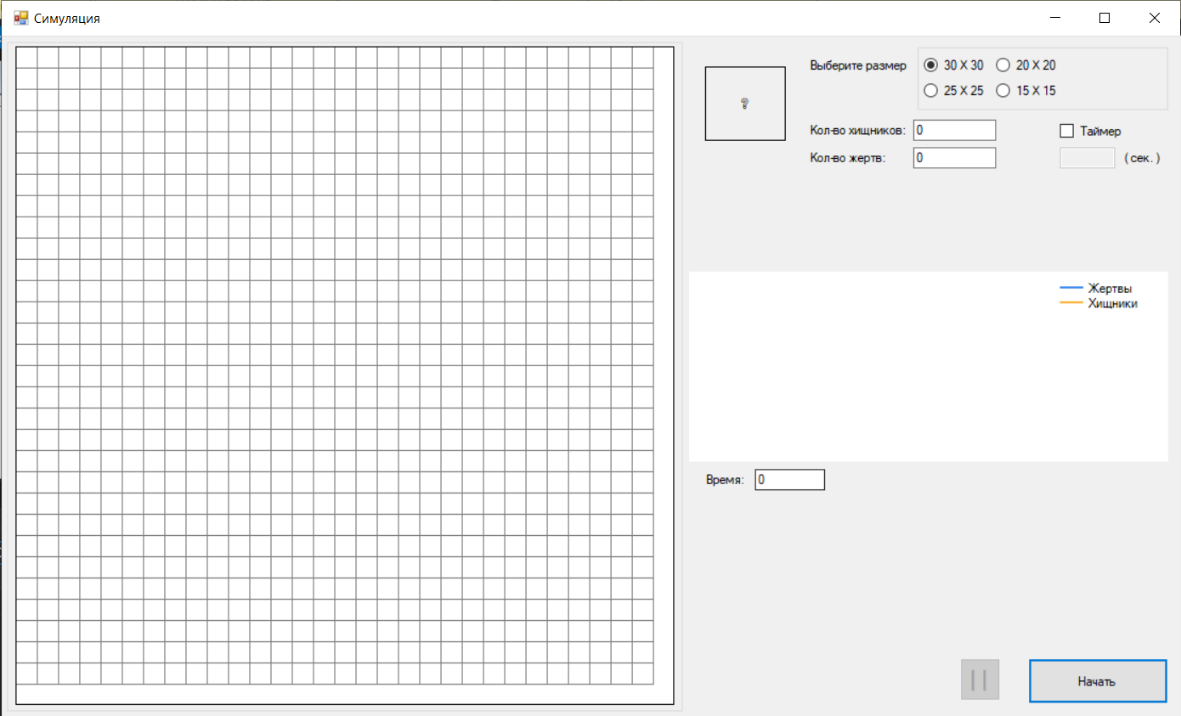


Рисунок 3 – Структура интерфейса

Интерфейс приложения содержит несколько разделов. Большую часть окна занимает поле, где будет происходить симуляция искусственной жизни. В правой части окна сверху можно настроить параметры модели (размер поля, количество особей, максимально время работы), а также открыть руководство пользователя. Ниже расположен график для динамического отслеживания изменения популяции сущностей и счетчик времени. В самом низу расположена большая кнопка для запуска модели, либо же её остановки в процессе работы.

## Алгоритмизация и программирование

Алгоритм — совокупность точно заданных правил решения некоторого класса задач или набор инструкций, описывающих порядок действий исполнителя для решения определённой задачи.

Алгоритму присущи следующие свойства:

* Определённость: Каждый шаг алгоритма должен быть четко определен и не подлежать двусмысленности.
* Конечность: Алгоритм должен иметь конечное количество шагов.
* Результативность: Алгоритм должен решать задачу и давать результат.
* Массовость: Алгоритм должен быть применим ко всем объектам, которые удовлетворяют входным условиям.
* Эффективность: Алгоритм должен быть эффективным, то есть использовать минимальное количество ресурсов (время, память и так далее) для достижения результата.

Алгоритмы удобно представлять в виде блок-схем согласно ГОСТ 19.701–90 «Единая система программной документации схемы алгоритмов, программ, данных и систем».

Для автоматизированной системы модели искусственной жизни «хищник-жертва» блок-схема алгоритма представлена на рисунке 4.

Изображение выглядит как диаграмма, текст, План, Технический чертеж

Автоматически созданное описание

Изображение выглядит как диаграмма, текст, План, Технический чертеж

Автоматически созданное описание

Рисунок 4 – Блок-схема алгоритма

Для реализации функций автоматизированной системы модели искусственной жизни «хищник-жертва» разработаны следующие методы:

1. button1\_Click(object sender, EventArgs e) – активируется при нажатии на кнопку. Считывает выбранные пользователем параметры модели, отрисовывает начальную сетку поля, формирует сущности хищников и жертв и расставляет их начальные позиции на поле. В конце происходит активация таймера для проигрывания модели.
2. SizeSelecter() – проверяет выбранный пользователем размер поля и возвращает его значение.
3. EntitiesConstruct(int[,] entities, int gridsize, List<Predator> predators, List<Prey> preys) – принимает на вход пустую матрицу для хранения координат сущностей, пустые списки сущностей и размер поля. Затем формируется список координат, которые случайным образом присваиваются создаваемым сущностям, и формируется сама матрица с координатами сущностей.
4. timer1\_Tick(object sender, EventArgs e) – начинает работу после запуска таймера и выполняется через определенные интервалы времени (по умолчанию – 100 мс). Определяет поведение сущностей и в соответствии с ним меняет модель. В конце измененная модель отрисовывается на поле.
5. button2\_Click(object sender, EventArgs e) – открывает руководство пользователя в формате документа.
6. button3\_Click(object sender, EventArgs e) – останавливает работу модели и возобновляет её при повторном нажатии.

## Тестирование

# Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения

## Руководство пользователя

Обзор возможностей программы

«Автоматизированная система модели искусственной жизни «хищник-жертва» ­­­– это программная модель, которая помогает отслеживать связь влияния популяций хищников и жертв в природе друг на друга.

Системные требования

Для стабильной и эффективной работы «Автоматизированной системы модели искусственной жизни «хищник-жертва» рекомендуется использовать следующую конфигурацию:

Частота процессора (CPU): 3.20 GHz

Количество ядер процессора (CPU): 2 ядра

Объем оперативной памяти (RAM): 2 GB

Объем свободного места на диске (HDD): 100 GB

Операционная система (OS): Windows 10

Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с Название Продукта.

Перед началом работы, пожалуйста, ознакомьтесь с системными требованиями.

Основные понятия и термины

Перед началом работы в автоматизированной системы модели искусственной жизни «хищник-жертва» рекомендуем ознакомиться с основными понятиями и терминами:

Компьютер - электронно-вычислительная машина, обрабатывающая и хранящая информацию, производящая вычисления и передающая результаты этих вычислений под управлением заложенных программ.

Программа - последовательность машинных команд, предназначенная для достижения конкретного результата.

C# - объектно-ориентированный язык программирования общего назначения.

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений.

Модель «хищник – жертва» – совместное существование двух популяций, в котором численность вида-жертвы в присутствии вида-хищника растет медленнее, а вида-хищника – быстрее.

Искусственная жизнь – изучение жизни, живых систем и их эволюции при помощи созданных человеком моделей и устройств.

Запуск

Для запуска автоматизированной системы модели искусственной жизни нажмите на ярлык программы.

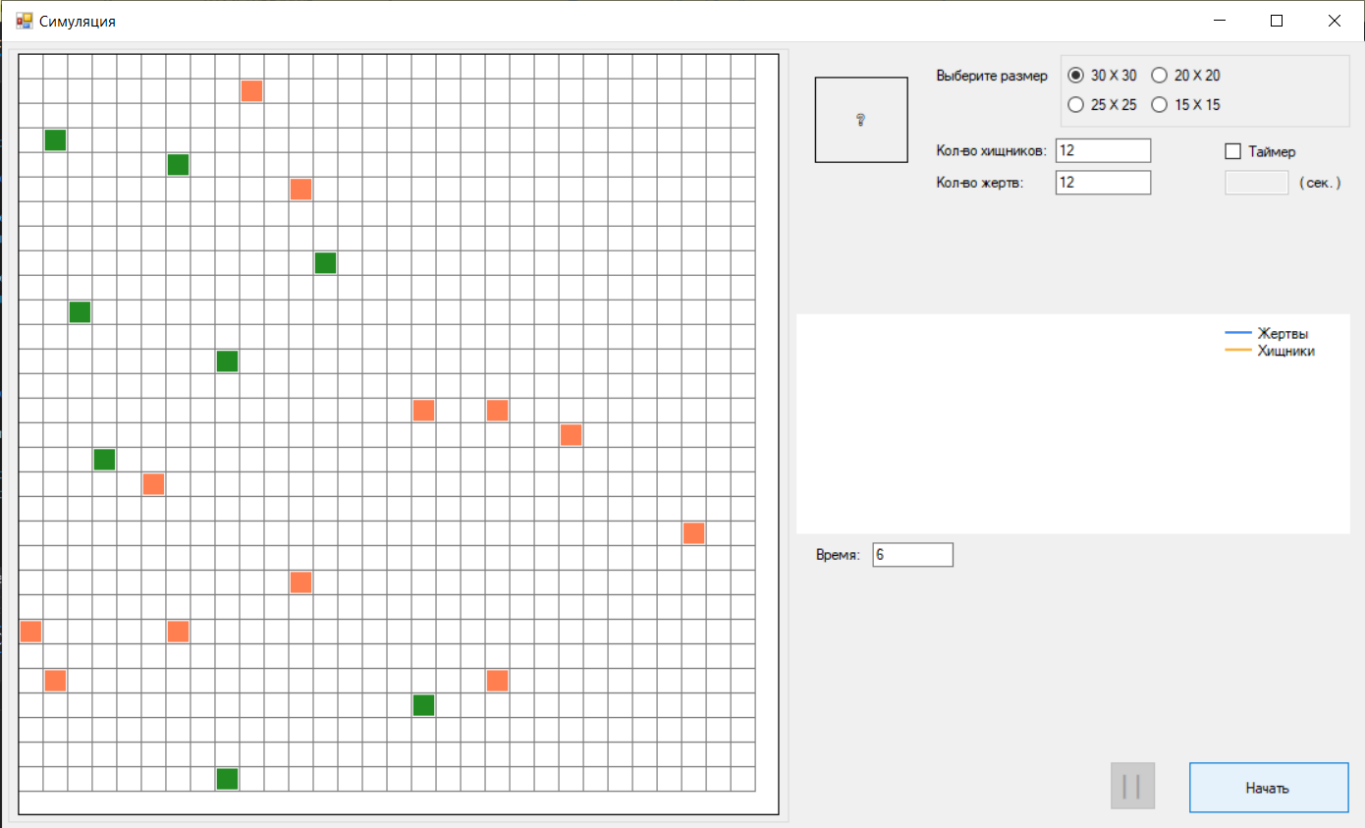
Пользовательский интерфейс

Этот раздел описывает основные элементы пользовательского интерфейса «Автоматизированной системы модели искусственной жизни «хищник-жертва» версии 1.0: основных режимов работы, предназначение окон и экранов, доступные операции.

Главное окно программы

Главное окно программы «Автоматизированной системы модели искусственной жизни «хищник-жертва» позволяет выполнять следующие операции:

* Открытие руководство пользователя
* Наблюдение за работой модели и ее данными
* Запуск и остановка работы модели



**Настройки программы**

**Настройки программы** позволяют редактировать параметры работы модели.

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

**Режимы работы**

«Автоматизированной системы модели искусственной жизни «хищник-жертва» работает только в пользовательском режиме, котором пользователю доступны все функции.

**Примеры использования**

В данном разделе собраны примеры реального использования Название Продукта, демонстрирующие применения продукта в различных отраслях.

**Кейс #1: <НЕ ЗАБЫТЬ ...>**

*Компания: <НЕ ЗАБЫТЬ ...>*

*Отрасль: <НЕ ЗАБЫТЬ ...>*

**Задача**

Перед клиентом стояла следующая задача - <НЕ ЗАБЫТЬ ...>

**Особые условия**

Следует отметить, что задачу необходимо было решить в условиях <НЕ ЗАБЫТЬ ...>

**Проблема**

До начала использования Название Продукта клиенту приходилось <НЕ ЗАБЫТЬ ...>

**Процесс**

Внедрение Название Продукта в рабочий процесс происходило по следующей схеме: <НЕ ЗАБЫТЬ ...>

**Результат**

Результатами использования Название Продукта явилось:

* <НЕ ЗАБЫТЬ ...>
* <НЕ ЗАБЫТЬ ...>

**Вывод**

На примере этого кейса, мы можем сделать вывод, что <НЕ ЗАБЫТЬ ...>

## План внедрения и сопровождения

# Заключение

На одну страницу примерно.

Подвести итоги проделанной работы.

# Список использованных источников

1. Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург: Изд-во ОГУ, 2015. – 89 с.
2. 7-10 источников в алф порядке
3. ГОСТ 19.701-90 Единая система программной документации схемы алгоритмов, программ, данных и систем
4. https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC

# Приложение А

(обязательное)

**Текст программы**