|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Министерство науки 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 дисциплине «Технология разработки программного обеспечения»  профессионального модуля «Осуществление интеграции программных модулей» | | | | | |
| **Разработка программного обеспечения для планирования модельных экспериментов** | | | | | |
| Пояснительная записка | | | | | |
| ОГУ 09.02.07. 3025. 819 ПЗ | | | | | |
|  | | | | | |
|  |  | Руководитель | |  | |
|  |  | преподаватель высшей категории | | | |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Ж.В. Михайличенко | |
|  |  | «\_\_\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | 2025 г. |
|  |  | Студент группы 23ИСП-1 | | | |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | С.Д. Симаев | |
|  |  | «\_\_\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | 2025 г. |
| Орск 2025 | | | | | |

|  |  |
| --- | --- |
| Утверждаю  председатель ПЦК дисциплин профессионального цикла | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись | А.Н. Макатаева |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. | |

**ЗАДАНИЕ**

## 

на выполнение курсовой работы

студенту\_\_\_\_\_Симаеву Сергею Денисовичу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 планирования модельных экспериментов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Срок сдачи студентом работы «14» \_\_июня\_\_\_ 2025 г.
3. Цель и задачи работы \_\_Разработать программный продукт для автоматизированного построения таблиц для проведения различных типов научных экспериментов с учётом заданного количества факторов и их уровней \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Исходные данные к работе: \_\_Учебники и интернет-источники по технологии разработки программного обеспечения и планированию модельных экспериментов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
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 «26» \_февраля\_\_\_\_\_ 2025 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Ж.В. Михайличенко\_\_\_

подпись инициалы, фамилия

Студент «26» \_февраля\_\_\_\_\_ 2025 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_С.Д. Симаев\_\_\_\_\_\_\_\_\_

подпись инициалы, фамилия

**Аннотация**

Курсовая работа посвящена ….

Пояснительная записка к курсовой работе состоит из двух глав. В первой главе………..

Во второй главе ……..

Работа содержит ?? листов текста, ?? рисунков, ?? таблиц, 1 приложение.

**Содержание**

[Введение 5](#_Toc198977475)

[1 Анализ требований и проектирование программного продукта 6](#_Toc198977476)

[1.1 Анализ предметной области 6](#_Toc198977477)

[1.2 Техническое задание на создание программного обеспечения 8](#_Toc198977478)

[1.3 Функциональное проектирование программной системы 12](#_Toc198977479)

[1.4 Проектирование пользовательского интерфейса 16](#_Toc198977480)

[2 Программная реализация и тестирование системы для планирования модельных экспериментов 20](#_Toc198977481)

[2.1 Обоснование выбора инструментальных средств разработки 20](#_Toc198977482)

[2.2 Реализация основных функций программного продукта 21](#_Toc198977483)

[2.3 Тестирование МЭПО 22](#_Toc198977484)

[2.4 Руководство пользователя 23](#_Toc198977485)

[2.5 Рекомендации по внедрению и сопровождению 23](#_Toc198977486)

[Заключение25](#_Toc198977487)

[Список использованных источников 26](#_Toc198977488)

[Приложение 27](#_Toc198977489)

# Введение

На сегодняшний день компьютерный эксперимент является наиболее значимым инструментом в областях, где имеется большой разрыв между возможностями теории и эксперимента. Это связано с тем, что сложность изучаемых систем постоянно возрастает, а требования к точности и достоверности результатов становятся все более строгими. Компьютерный эксперимент, как метод исследования, позволяет преодолеть ограничения физических экспериментов, такие как высокая стоимость, временные затраты и невозможность полного контроля над всеми факторами.

Актуальность темы курсовой работы также обусловлена тем, что современные системы характеризуются множеством взаимосвязанных факторов, которые трудно анализировать вручную. Разработка программного обеспечения, способного учитывать эти факторы и предлагать оптимальные планы экспериментов, позволит существенно упростить работу исследователей и повысить качество получаемых результатов.

Таким образом, создание инструмента для планирования модельных экспериментов является актуальной задачей, отвечающей современным тенденциям развития науки и технологий.

Целью курсовой работы является разработка программного обеспечения, предназначенного для автоматизации процесса планирования модельных экспериментов. Программа должна предоставлять пользователю возможность настраивать параметры экспериментов, выбирать оптимальные планы и анализировать полученные результаты.

Для достижения указанной цели необходимо решить следующие задачи:

* анализ предметной области, изучая существующие методы и подходы к планированию модельных экспериментов, выделяя их преимущества и недостатки;
* проектирование архитектуры программного обеспечения, включающее в себя разработку модулей для ввода данных, расчёта планов экспериментов и анализа результатов;
* реализация программного обеспечения с использованием C#, .NET Framework и Visual Studio;
* проведение функционального тестирования на различных наборах данных, чтобы убедиться в корректности расчётов и нефункционального для проверки удобства использования;
* подготовка рекомендаций по внедрению программной системы.
* оформление пояснительной записки к курсовой работе.

# 1 Анализ требований и проектирование программного продукта

## 1.1 Анализ предметной области

Модельный эксперимент есть метод научного познания для изучения объекта-оригинала путём создания и исследования субъектом его прототипа, замещающего реальную систему с определённых сторон, интересующих познание, и с последующим переносом полученной информации на систему.

Планирование модельных экспериментов – это методология, направленная на организацию и проведение экспериментов с использованием моделей (математических, имитационных или физических) для получения достоверной информации об исследуемой системе при минимальных затратах ресурсов. Целью планирования является определение условий проведения эксперимента таким образом, чтобы максимизировать информативность каждого измерения или испытания.

Планирование модельных экспериментов преследует две основные цели:

* сокращение общего объёма испытаний при соблюдении требовании к достоверности и точности их результатов;
* повышение информативности каждого из экспериментов в отдельности.

Для правильной организации модельного эксперимента исследователь должен располагать следующей информацией:

* исследователь и на этапе планирования эксперимента должен помнить, к какому классу относится моделируемая система (статическая или динамическая, детерминированная или стохастическая и так далее);
* он должен определить, какой режим работы системы его интересует: стационарный (установившийся) или нестационарный;
* необходимо знать, в течение какого промежутка времени следует наблюдать за поведением (функционированием) системы;
* нужно знать, какой объём испытаний сможет обеспечить требуемую точность оценок исследуемых характеристик системы.

В случае, когда на работу объекта влияет множество как внутренних, так и внешних случайных факторов функциональная искомая зависимость становится довольно сложной. Для получения таких зависимостей используется математический аппарат «планирование экспериментов». Он позволяет при ограниченном числе экспериментов получить абстрактное, но довольно точное описание функциональной зависимости в виде регрессии, состоящей из ряда всевозможных сочетаний значений внутренних и внешних факторов.

Исследователь системы, чтобы получить от модели максимум информации, должен проанализировать её работу во всех режимах, для всех возможных сочетаний внешних и внутренних факторов и повторять каждый эксперимент множество раз. Таким образом, полученные данные, из-за большого объёма, трудно поддаются анализу, да и время, затраченное на их получение чрезмерно. Для устранения указанных недостатков прибегают к так называемому «планированию» экспериментов.

Существуют два основных варианта постановки задачи планирования эксперимента:

* выбрать план, который при ограниченном числе опытов позволял бы получить наиболее достоверное значения искомой функциональной зависимости;
* выбрать план, при котором статистическая оценка функциональной зависимости может быть получена с заданной точностью при минимальном числе опытов.

Поиск плана эксперимента производится в так называемом факторном пространстве.

Факторное пространство – это множество внешних и внутренних параметров модели, значения которых исследователь может контролировать в ходе подготовки и проведения модельного эксперимента.

Во многих случаях факторы могут носить не только количественный, но и качественный характер. Значения факторов обычно называют уровнями. Если при проведении эксперимента исследователь может изменять уровни факторов, эксперимент называется активным, в противном случае – пассивным.

Каждый из факторов имеет верхний и нижний уровни, расположенные симметрично относительно некоторого нулевого уровня. Точка в факторном пространстве, соответствующая нулевым уровням всех факторов, называется центром плана.

Как правило, план эксперимента строится относительно одного (основного) выходного скалярного параметра Y, который называется наблюдаемой переменной. Если моделирование используется как инструмент принятия решения, то в роли наблюдаемой переменной выступает показатель эффективности.

При этом предполагается, что значение наблюдаемой переменной, полученное в ходе эксперимента, складывается из двух составляющих по формуле (1).

, (1)

где f(x) – функция отклика (неслучайная функция факторов);

е(х) – ошибка эксперимента (случайная величина);

х – точка в факторном пространстве (определённое сочетание уровней факторов).

Существует два основных варианта постановки задачи планирования модельного эксперимента:

1 Из всех допустимых выбрать такой план, который позволил бы получить наиболее достоверное значение функции отклика f(x) при фиксированном числе опытов.

2 Выбрать такой допустимый план, при котором статистическая оценка функции отклика может быть получена с заданной точнос­тью при минимальном объёме испытаний.

Решение задачи планирования в первой постановке называется страте­гическим планированием эксперимента, во второй – тактическим планиро­ванием.

Полный факторный эксперимент (ПФЭ) – это метод планирования экспериментов, при котором реализуются все возможные комбинации факторов на выбранных уровнях. Этот подход позволяет получить максимально полную информацию о взаимодействии факторов и их влиянии на выходную переменную.

Основные характеристики ПФЭ:

* Число опытов: если рассматривается k факторов, каждый из которых имеет два уровня (например, верхний и нижний), то общее число опытов равно двум в степени k. Например, для трёх факторов потребуется восемь опытов.
* Преимущества: позволяет оценить все возможные взаимодействия между факторами, обеспечивает высокую точность и достоверность результатов.
* Недостатки: значительное количество опытов при большом числе факторов делает этот метод ресурсоёмким, может быть непрактичным для сложных систем с большим числом факторов.

Частичный факторный эксперимент (ЧФЭ) – это метод планирования экспериментов, при котором исследуется лишь часть комбинаций уровней факторов из полного факторного эксперимента (ПФЭ). Это позволяет сократить число опытов, сохраняя возможность оценки ключевых эффектов. ЧФЭ особенно актуален при большом количестве факторов, где ПФЭ становится ресурсоёмким.

Основные разновидности ЧФЭ:

* Дробные реплики: ЧФЭ строится как часть ПФЭ, где часть комбинаций уровней факторов исключается. Например, в ЧФЭ типа 2 в степени k–p число опытов сокращается в два в степени p раз за счёт объединения эффектов высших взаимодействий с основными эффектами или низшими взаимодействиями. Такие реплики обозначаются через генераторы, определяющие, какие взаимодействия смешиваются.
* Специализированные планы: для минимизации числа опытов применяются планы, учитывающие априорные знания о системе. Например, если известно, что некоторые взаимодействия незначимы, их можно исключить из модели, упростив эксперимент.

Особенности ЧФЭ:

Точность: при правильном планировании оценки коэффициентов остаются достоверными даже при сокращении опытов.

Применение: используется в биологии, химии, инженерии для исследования сложных систем с ограниченными ресурсами.

## 1.2 Техническое задание на создание программного обеспечения

Техническое задание (ТЗ) – это документ, который содержит подробные требования к проекту и определяет цели, задачи, последовательность действий и методы их выполнения. ТЗ составляется по ГОСТ 34.602 – 2020 Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.

ТЗ на создание программного обеспечения для планирования модельных экспериментов представлено ниже.

1 Общие сведения.

а) Полное наименование системы и её условное обозначение Программное обеспечение для планирования модельных экспериментов (МЭПО);

б) Наименование заказчика: факультет среднего профессионального образования Орского гуманитарно-технологического института (филиала) ОГУ;

Наименование разработчика: студент второго курса группы 23ИСП-1 Симаев С.Д.

в) Документ, на основании которого создаётся МЭПО: Протокол закрепления тем и выдачи заданий курсовых работ по междисциплинарному курсу «Технология разработки программного обеспечения» профессионального модуля «Осуществления интеграции программных модулей» специальности 09.02.07 Информационные системы и программирование от 26.02.2025 г.;

г) Плановый срок создания системы 14.06.2025 г.;

Плановый срок окончания работ по созданию системы: 14.06.2025 г.

2 Цели и назначение создания автоматизированной системы.

а) Цели создания системы: автоматизация процесса планирования модельных экспериментов, предоставление пользователю возможности настраивать параметры экспериментов, выбор оптимальных планов и анализ полученных результатов;

б) Назначение системы: автоматизации процесса планирования модельных экспериментов;

3 Характеристика объекта автоматизации.

а) Объектом автоматизации является система планирования модельных экспериментов, предназначенная для поддержки исследований. Основные процессы, подлежащие автоматизации: планирование экспериментов, выполнение расчётов;

б) сведения об условиях эксплуатации объекта автоматизации: планирование экспериментов без автоматизации обычно выполняется вручную и требует значительных временных и трудовых затрат. На начальном этапе исследователь определяет цель эксперимента, выбирает факторы и их уровни, а также составляет план измерений. Затем проводится ручной расчёт комбинаций параметров, что может быть утомительным и подверженным ошибкам, особенно при работе с большим количеством переменных. Для сбора данных используются традиционные инструменты, а анализ результатов выполняется с помощью базовых математических методов или программ типа Excel. Однако такой подход часто не позволяет эффективно обрабатывать сложные зависимости между факторами и требует дополнительного времени на проверку корректности данных.

4 Требования к автоматизированной системе.

а) Требования к структуре системы в целом:

* Система должна быть модульной, включающей в себя модуль для ввода данных, построение планов экспериментов;
* Пользовательский интерфейс должен предоставлять удобство ввода данных;

б) Требования к функциям, выполняемым системой:

* Возможность выбора типа плана эксперимента;
* Ввод параметров выбранного эксперимента (факторного пространства);
* Расчёт количества экспериментов;
* Формирование плана полного факторного эксперимента;
* Формирование плана рандомизированного факторного эксперимента;
* Формирование плана эксперимента с изменением факторов по одному;
* Формирование плана дробного факторного эксперимента;
* Формирование плана факторного эксперимента «Латинский квадрат»;
* Вывод плана эксперимента на экран монитора;

в) Требования к видам обеспечения системы:

* К математическому обеспечению: использование алгоритмов для автоматической генерации комбинаций параметров экспериментов, реализация методов анализа данных;
* К информационному обеспечению: возможность экспорта данных для дальнейшего использования в научных публикациях или отчётах;
* К лингвистическому обеспечению: поддержка русского языка;
* К программному обеспечению: поддержка операционной системы Windows 10, редактор диаграмм MS Visio, CASE-средство Ramus Educational, текстовый процессор MS Word, среда разработки Visual Studio, браузеры (OperaGX, FireFox);
* К техническому обеспечению: персональный компьютер с процессором Intel Core i3 или аналог с тактовой частотой не менее 1,6 ГГц, минимум 4 ГБ оперативной памяти, дисковое пространство не менее 500 МБ, видеокарта с поддержкой Full HD разрешения (1920x1080), принтер для печати планов экспериментов;
* К организационному обеспечению: наличие руководства пользователя, организация технической поддержки, возможность функционирования с одним пользователем, имеющим базовые навыки работы с компьютером;

г) Общие технические требования к системе: Система должна обрабатывать запрос не более чем за пять секунд, интерфейс системы должен корректно отображаться на устройствах с различными размерами экранов;

5 Состав и содержание работ по созданию автоматизированной системы.

Состав работ по созданию МЭПО представлен в таблице 1.

Таблица 1 - Состав работ по созданию МЭПО

|  |  |  |
| --- | --- | --- |
| Номер и название этапа | Сроки исполнения | Описание работ |
| 1 Анализ требований | 26.02.2025 – 12.03.2025 | Формируются цели и задачи системы. Определяется стоимость разработки. Устанавливается область применения МЭПО. Определяются требования к МЭПО. Определяются граничные условия. Формируется и подписывается ТЗ. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Номер и название этапа | Сроки исполнения | Описание работ |
| 2 Проектирование | 13.02.2025 ­– 30.03.2025 | Определение архитектуры системы. Построение функциональной модели. Выявление внешних условий функционирования. Проектирование интерфейсов и распределение функционала между пользователем и системой. Определение перечня документов. Определение мероприятий для внедрения МЭПО. |
| 3 Программирование | 31.03.2025 – 10.05.2025 | Строятся прототипы системы. Разрабатывается программный код. Выполняется отладка программы. Создаётся техническая документация. |
| 4 Тестирование | 11.05.2025 – 25.05.2025 | Испытание МЭПО. Проведение функционального и нефункционального тестирования. |
| 5 Внедрение | 26.05.2025 – 14.06.2025 | Установка системы. Консультация пользователей. Документирование. |
| 6 Эксплуатация и сопровождение | С 14.06.2025 | Адаптация МЭПО к новым условиям. Внесение изменений в МЭПО и соответствующую документацию, вызванных потребностью модификаций. |

6 Порядок разработки автоматизированной системы.

а) Порядок организации разработки системы: проектирование архитектуры системы, разработка прототипов интерфейса, программирование модулей системы, установка системы на рабочие компьютеры, настройка оборудования и программного обеспечения, обучение сотрудников работе с системой, техническая поддержка;

б) Требования к гарантийным обязательствам разработчика: гарантийный срок составляет шесть месяцевс момента внедрения системы, устранение всех выявленных дефектов и ошибок в работе системы, бесплатное устранение проблем, возникших по вине разработчика, реакция на обращения заказчика в течение трёх часов, после окончания гарантийного срока разработчик предоставляет платную техническую поддержку;

7 Порядок контроля и приёмки автоматизированной системы.

а) Виды, состав и методы испытаний системы:

* Виды: Функциональное, интерфейсное;
* Состав: тестирование каждого модуля, тестирование взаимодействия модулей;
* Ручное тестирование;

б) Общие требования приёмки работ:

* Критерии приёмки: соответствие заданиям ТЗ, успешное прохождение тестирований;
* Документация: предоставление полного комплекта документации, включая руководство пользователя и результаты испытаний;
* Процедура приёмки: проведение приёмочного тестирования совместно с заказчиком, передача системы в эксплуатацию;

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

* Установка МЭПО на персональном компьютере заказчика;
* Проведение практических занятий по работе с МЭПО.
* Оптимизация производительности.

9 Требования к документированию.

* Техническое задание;
* Руководство пользователя;
* Результаты тестирования;
* Рекомендации по внедрению и сопровождению МЭПО.

10 Источники разработки.

* Протокол закрепления тем и выдачи заданий курсовых работ по междисциплинарному курсу «Технология разработки программного обеспечения» профессионального модуля «Осуществления интеграции программных модулей» специальности 09.02.07 Информационные системы и программирование от 26.02.2025 г.;
* СТО 02069024.101 – 2015 Работы студенческие. Общие требования и правила оформления;
* ГОСТ 34.602 – 2020 Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
* ГОСТ 34.601 –­ 90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

## 1.3 Функциональное проектирование программной системы

Функциональное проектирование представляет собой подход, при котором программное обеспечение декомпозируется на подсистемы, которые в свою очередь делятся на подфункции и так далее до конкретных задач. При этом автоматизированная система сохраняет целостность. Определение интеграции (Integration DEFinition IDEF0) – это методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Она позволяет наглядно представить функции, входные и выходные данные, управляющие элементы и механизмы системы в виде блоков и связей между ними.

Для реализации таких моделей могут применяться специализированные инструменты, например Ramus Educational – это CASE-средство, предназначенное для поддержки процесса разработки программного обеспечения, обучения и моделирования. Оно используется в образовательных целях, а также может применяться профессионалами для создания диаграмм, моделей и документации, необходимых на различных этапах разработки программного обеспечения.

Исходя из функций МЭПО построены функциональные модели IDEF0, изображённые на рисунке 1.

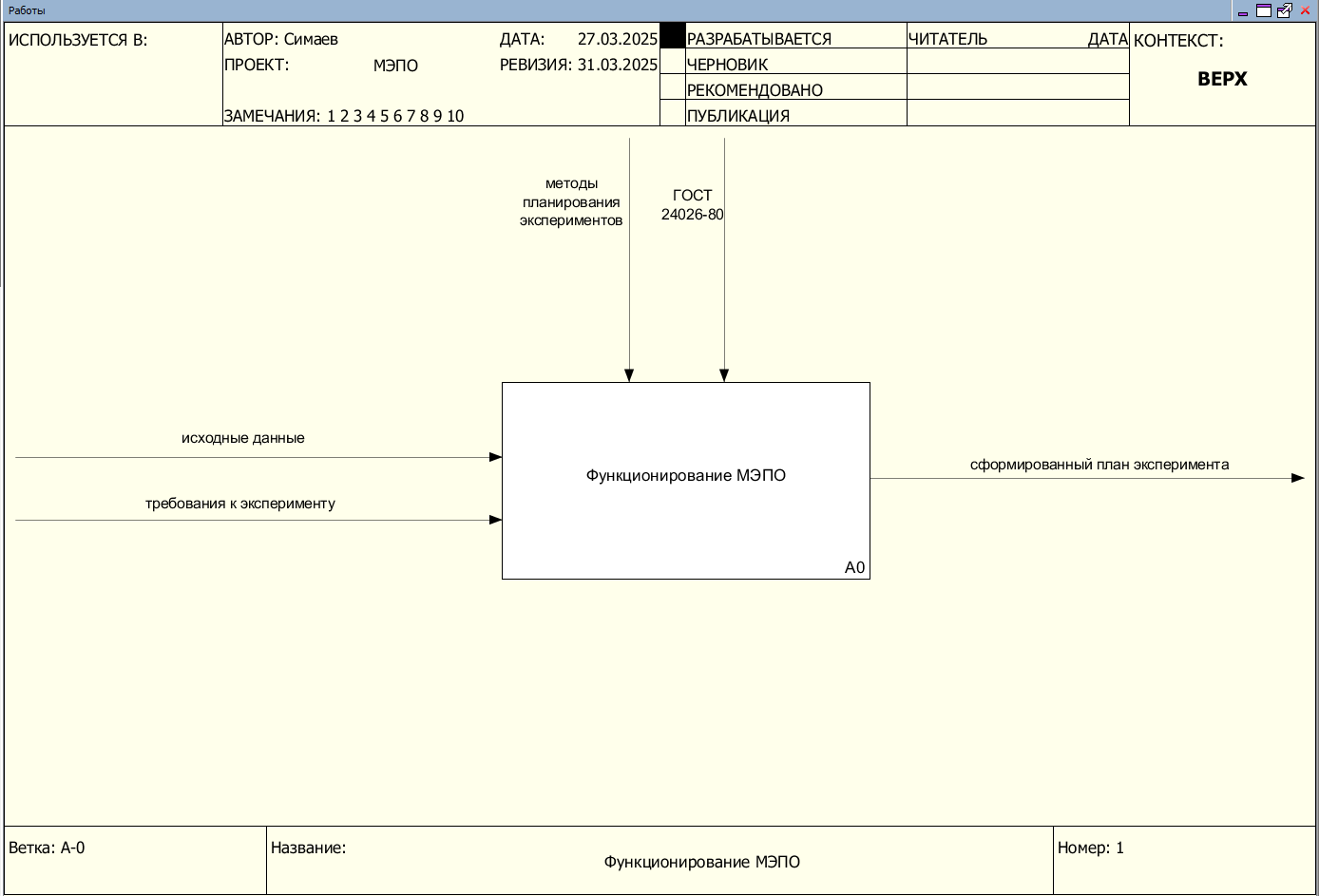


Рисунок 1 – Контекстная диаграмма модели IDEF0

Блок A0 представляет собой процесс функционирования МЭПО:

* Вход: исходные данные, требования к эксперименту;
* Управление: методы планирования экспериментов, ГОСТ 24026-08;
* Выход: сформированный план эксперимента.

Декомпозиция контекстной диаграммы «Функционирование МЭПО» представлена на рисунке 2.

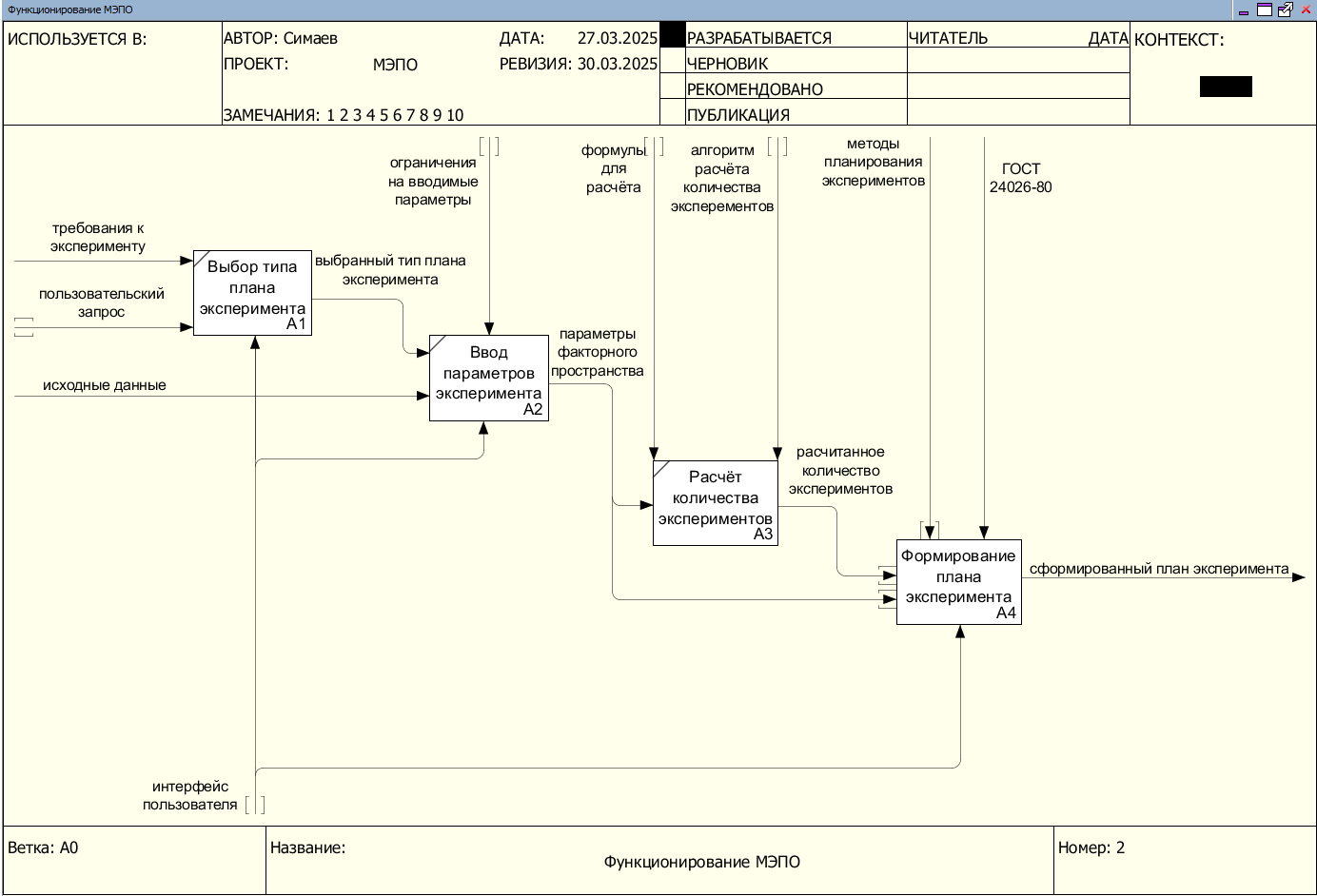


Рисунок 2 – Первый уровень модели IDEF0

Контекстной диаграмма «Функционирование МЭПО» декомпозирована на блоки:

Блок А1 представляет собой процесс выбора типа план эксперимента:

* Вход: требования к эксперименту, пользовательский запрос;
* Механизм: интерфейс пользователя;
* Выход: выбранный тип плана эксперимента.

Блок A2 представляет собой процесс ввода параметров эксперимента:

* Вход: исходные данные, выбранный тип плана эксперимента;
* Механизм: интерфейс пользователя;
* Управление: ограничение на вводимые параметры;
* Выход: параметры факторного пространства.

Блок A3 представляет собой процесс расчёта количества экспериментов:

* Вход: параметры факторного пространства;
* Управление: формулы для расчёта, алгоритм расчёта количества экспериментов;
* Выход: рассчитанное количество экспериментов.

Блок A4 представляет собой процесс формирования плана эксперимента:

* Вход: рассчитанное количество экспериментов, параметры факторного пространства;
* Управление: методы планирования экспериментов, ГОСТ24026-80;
* Механизм: интерфейс пользователя;
* Выход: сформированный план эксперимента.

Декомпозиция блока А4 на примере полного факторного эксперимента показана на рисунке 3.

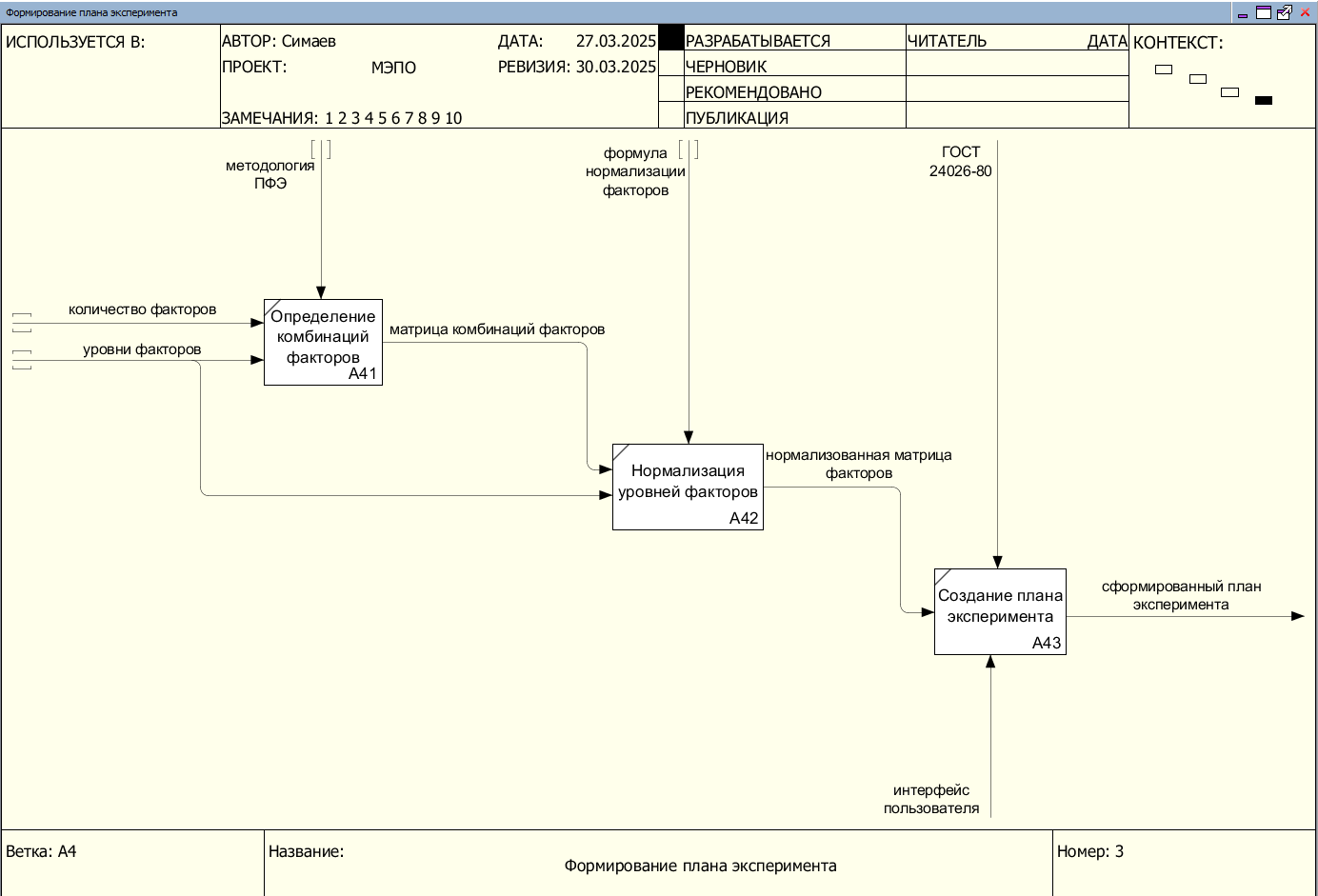


Рисунок 3 – Второй уровень декомпозиции блока А4 модели IDEF0

Блок А41 представляет собой процесс определения комбинаций факторов:

* Вход: количество факторов, уровни факторов;
* Управление: методология ПФЭ;
* Выход: матрица комбинаций факторов.

Блок А42 представляет собой процесс нормализации уровней факторов:

* Вход: матрица комбинаций факторов, уровни факторов;
* Управление: формула нормализации факторов;
* Выход: нормализованная матрица факторов.

Блок А43 представляет собой процесс создания плана эксперимента:

* Вход: нормализованная матрица факторов;
* Механизм: интерфейс пользователя;
* Управление: ГОСТ 24026-80;
* Выход: сформированный план эксперимента.

## 1.4 Проектирование пользовательского интерфейса

Пользовательский интерфейс – это средство взаимодействия между пользователем и системой, которое обеспечивает передачу информации между человеком и программно-аппаратными компонентами компьютерной системы. Это совокупность программных средств, позволяющих пользователю управлять программами с помощью голоса, нажатий, жестов или командной строки. Основная цель пользовательского интерфейса – предоставить инструменты для комфортного и эффективного взаимодействия человека с системой, будь то компьютер, мобильное приложение или веб-сайт.

Основные требования к программному интерфейсу:

* Интерфейс должен быть интуитивно понятным, чтобы пользователь мог легко освоить его без дополнительного обучения. Необходимо избегать сложных элементов и предоставлять только те функции, которые действительно нужны.
* Важно учитывать потребности и ожидания целевой аудитории. Каждый элемент интерфейса должен быть разработан с учётом того, как пользователи будут его использовать. Например, предоставление информации и функций, которые действительно необходимы, делает интерфейс более эффективным.
* Элементы интерфейса должны быть согласованы между собой, чтобы пользователь не испытывал путаницы при переходе от одной части системы к другой.
* Все сообщения и команды должны быть предельно ясными. Пользователь должен получать немедленную обратную связь на свои действия – например, визуальные или звуковые сигналы, подтверждающие успешное выполнение операции.
* В интерфейсе не должно быть лишней или ненужной информации, которая может отвлекать пользователя. Каждый элемент должен иметь свою логику и цель, избегая перегруженности деталями.
* Интерфейс должен быть спроектирован таким образом, чтобы все важные функции были легко доступны, без необходимости углублённого поиска нужных элементов управления.

Интерфейс программного проекта является многооконным и состоит из трёх окон. Главное окно на рисунке (скриншот из конструктора) дальше описание (какие элементы windows forms представлены, для чего используются)

Интерфейс МЭПО является многооконным, состоит из трёх окон. Главное окно изображено на рисунке 4.

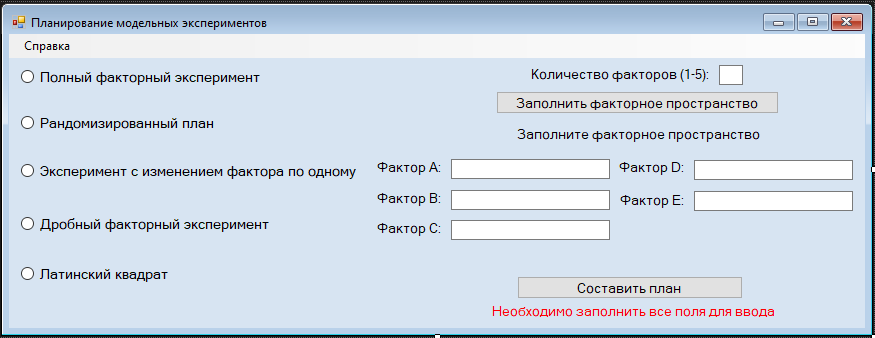


Рисунок 4 – Главное окно МЭПО в конструкторе

Главное окно состоит из двух частей, списка доступных планов и зоны для ввода первичных значений, которая динамически меняется в зависимости от выбранного плана. Для выбора типа плана эксперимента используется элемент radioButton. Для ввода значений применяются элементы textbox и Button. Пояснительные слова на форме размещены с помощью элемента label. Также на форме находится кнопка для перехода к следующему окну приложения.

Окно для вывода всех планов эксперимента, кроме плана «Латинский квадрат» будет единым, за исключением названия самого окна и представлено на рисунке 5.

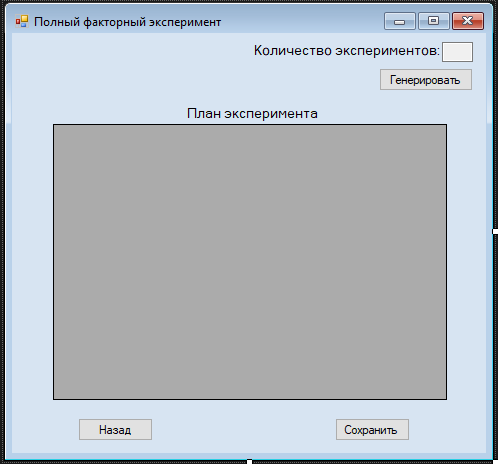


Рисунок 5 – Окно для вывода плана эксперимента в конструкторе

Окно для вывода экспериментов состоит из dataGridView для отображения плана, textbox для отображения общего количества экспериментов, а также кнопок (button) «Назад» для возвращения к главной форме и «Сохранить» для записи плана эксперимента в файл. На форме есть кнопка «Создать план», которая отображается только при выборе «Рандомизированного плана».

Нетипичным является окно для вывода плана «Латинский квадрат», оно изображено на рисунке 6.

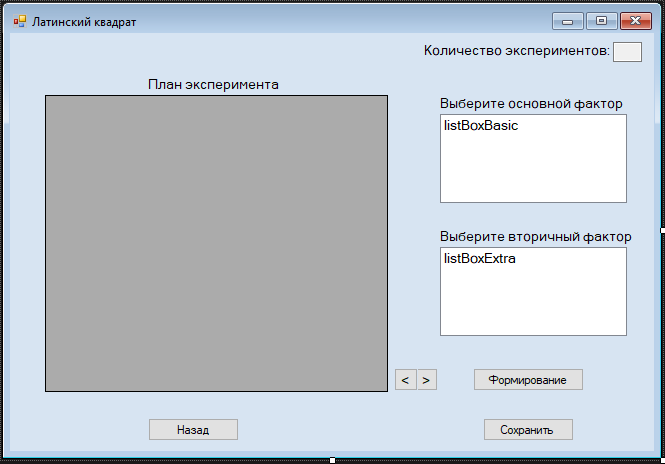


Рисунок 6 – Окно плана «Латинский квадрат» в конструкторе

Окно плана «Латинский квадрат» в конструкторе состоит из dataGridView для вывода плана эксперимента, listBox для выбора основного и вторично фактора, кнопки представлены элементом button. Кнопки «Назад» и «Сохранить» повторяют функционал таких же кнопок из окна для вывода плана остальных экспериментов, а кнопка «Генерация» принимает то, какой из факторов пользователь выбрал в качестве основного и создаёт план эксперимента.

# 2 Программная реализация и тестирование системы для планирования модельных экспериментов

## 2.1 Обоснование выбора инструментальных средств разработки

Согласно ТЗ, создание МЭПО должно осуществляться на языке программирования C# в интегрированной среде разработки Visual Studio. Выбор языка C# обусловлен следующими его преимуществами:

* C# обладает продвинутым синтаксисом, который является улучшенной версией многих других языков программирования, таких как Java и C++. Язык был разработан позже этих языков, что позволило внедрить более современные подходы к программированию.
* C# является строго типизированным языком, что означает, что компилятор проверяет код на ошибки, связанные с типами данных, ещё до его выполнения. Это помогает избежать многих распространённых ошибок на этапе разработки.
* C# полностью поддерживает парадигму объектно-ориентированного программирования (ООП), включая такие концепции, как классы, наследование, полиморфизм и инкапсуляция. Эти возможности позволяют создавать масштабируемые и легко поддерживаемые программы. Благодаря ООП-подходу разработчики могут организовать код в виде модульных компонентов, что упрощает его повторное использование и тестирование.
* Отсутствие необходимости следить за расходом памяти. В C# автоматически происходит очистка памяти от ненужных объектов.
* Выбор Visual Studio как интегрированной среды разработки обоснован следующими её преимуществами:
* Visual Studio – это одна из самых популярных интегрированных сред разработки (integrated development environment IDE), которая предоставляет разработчикам множество инструментов для написания, отладки и тестирования кода. Она поддерживает множество языков программирования, включая C#, VB.NET, F# и другие, что делает её универсальным решением для разработки. Встроенные функции, такие как IntelliSense (автодополнение кода), анализаторы кода и отладчики, значительно ускоряют процесс разработки и помогают избежать ошибок.
* Одним из ключевых преимуществ Visual Studio является наличие мощных инструментов для графического дизайна интерфейсов. Например, при разработке десктопных приложений с использованием Windows Forms можно использовать визуальный редактор для создания пользовательских интерфейсов «перетаскиванием» элементов управления. Это значительно упрощает процесс проектирования и экономит время.
* Visual Studio поддерживает огромное количество плагинов и расширений, которые позволяют адаптировать среду под конкретные нужды разработчика.

## 2.2 Алгоритмизация и программирование

Согласно четвёртому пункту технического задания основными функциями МЭПО являются:

* Возможность выбора типа плана эксперимента;
* Ввод параметров выбранного эксперимента (факторного пространства);
* Расчёт количества экспериментов;
* Формирование плана полного факторного эксперимента;
* Формирование плана рандомизированного факторного эксперимента;
* Формирование плана эксперимента с изменением факторов по одному;
* Формирование плана дробного факторного эксперимента;
* Формирование плана факторного эксперимента «Латинский квадрат»;
* Вывод плана эксперимента на экран монитора;
* Сохранение плана эксперимента.

Вышеперечисленные функции реализованы с помощью методов, представленных в таблицах 2 - 4.

Таблица 2 – Список методов формы «Планирование модельных экспериментов»

|  |  |  |
| --- | --- | --- |
| № метода | Название метода | Назначение метода |
| 1 | visibleOff | Отключение отображения элементов интерфейса для ввода данных |
| 2 | collectData | Сбор данных, введённых пользователем в поля ввода |
| 3 | areaIsFill | Проверка заполненности активных полей ввода |
| 4 | textBox1\_KeyPress | Ограничение данных, доступных для ввода в поле |
| 5 | button1\_Click | Отображение полей ввода в соответствии с выбором пользователя |
| 6 | radioButtonFull\_CheckedChanged | Отслеживание изменения состояния элемента radioButtonFull |
| 7 | radioButtonRand\_CheckedChanged | Отслеживание изменения состояния элемента radioButtonRand |

|  |  |  |
| --- | --- | --- |
| № метода | Название метода | Назначение метода |
| 8 | radioButtonOne\_CheckedChanged | Отслеживание изменения состояния элемента radioButtonOne |
| 9 | radioButtonFractional\_CheckedChanged | Отслеживание изменения состояния элемента radioButtonFractional |
| 10 | radioButtonLat\_CheckedChanged | Отслеживание изменения состояния элемента radioButton |
| 11 | buttonPlan\_Click | Перенаправление пользователя к окну построенным планом |
| 12 | textBoxA\_KeyPress | Защита поля от ввода некорректных значений |
| 13 | textBoxB\_KeyPress | Защита поля от ввода некорректных значений |
| 14 | textBoxC\_KeyPress | Защита поля от ввода некорректных значений |
| 15 | textBoxD\_KeyPress | Защита поля от ввода некорректных значений |
| 16 | textBoxE\_KeyPress | Защита поля от ввода некорректных значений |

Таблица 3 – Список методов формы «Факторный эксперимент»

|  |  |  |
| --- | --- | --- |
| № метода | Название метода | Назначение метода |
| 1 | GenerateFractionalFactorialPlan | Формирование дробного факторного плана. Вызов метода для вывода плана эксперимента |
| 2 | GenerateAllCombinations | Запуск формирования всех возможных комбинаций в соответствии с введёнными от пользователя данными |
| 3 | GenerateCombinationsRecursive | Формирование всех возможных комбинаций в соответствии с введёнными от пользователя данными |
| 4 | addDataPlan | Отображение сформированных данных в виде таблицы с помощью элемента dataGridView |
| 5 | buttonBack\_Click | Закрытие окна с планом и возвращение к основному окну приложения |
| 6 | textBoxCount\_KeyPress | Защита поля от ввода некорректных данных |

|  |  |  |
| --- | --- | --- |
| № метода | Название метода | Назначение метода |
| 7 | generateRandomizedPlan | Случайная выборка комбинаций для составления рандомизированного плана |
| 9 | buttonGen\_Click | Запуск генерации рандомизированного плана |
| 10 | GenerateOnePlan | Формирование плана факторного эксперимента с изменением факторов по одному |
| 11 | buttonSave\_Click | Сохранение сформированного плана факторного эксперимента |

Таблица 4 – Список методов формы «Латинский квадрат»

|  |  |  |
| --- | --- | --- |
| № метода | Название метода | Назначение метода |
| 1 | HideAllGrids | Скрытие таблиц с планом |
| 2 | ShowGrid | Отображение номера и таблицы с планом |
| 3 | buttonBack\_Click | Закрытие окна с планом и возвращение к основному окну приложения |
| 4 | buttonRight\_Click | Перелистывание отображаемый таблицы вправо |
| 5 | buttonLeft\_Click | Перелистывание отображаемый таблицы влево |
| 6 | buttonGen\_Click | Подсчёт общего количества экспериментов и запуск генерации плана «Латинский квадрат» |
| 7 | GenerateLatinSquare | Формирование плана «Латинский квадрат» |
| 8 | GetDataGridView | Получение номера dataGridView |
| 9 | listBoxBasic\_SelectedIndexChanged | Выбор базового фактора для построения плана эксперимента |
| 10 | ClearDataGridViews | Очистка dataGridView при изменении пользователем выбранных факторов |
| 11 | buttonSave\_Click | Сохранение плана эксперимента в файл |

На рисунке 7 представлена блок-схема алгоритма формирования плана с изменением факторов по одному.

factor in factors

start

factors[]

factorIndex = 0; factorIndex < factorIndex.Length; factorIndex++

level in factors[factor].Values

Combination = fixedValues

combination[FactorIndex] = level

addDataPlan(combination)

End

Добавить fator в fixedValues

Преобразовать factor в строку

factor = factor.Values[0]

Рисунок 7 – Блок-схема алгоритма формирования плана с изменением факторов по одному

## 2.3 Тестирование МЭПО

Тестирование – это проверка программного обеспечения, которая показывает, соответствует ли оно ожиданиям разработчиков и правильно ли работает. Оно помогает установить надёжность, стабильность и качество программы, а также повысить уровень удовлетворённости пользователей её работой. Тестирование позволяет ускорить разработку программного обеспечения, сократить затраты на исправление ошибок после выпуска и снизить риски убытков и потерь, связанных с неправильной работой программы.

Для МЭПО будут использованы следующие виды тестирования:

1. Функциональное тестирование – проверка того, насколько верно были реализованы заявленные функции ПО:

* Корректность выводимого результата;
* Работоспособность всех интерактивных элементов интерфейса;
* Защита от некорректных данных.

1. Тестирование пользовательского интерфейса ­– проверка удобство того, насколько удобно пользователю взаимодействовать с элементами интерфейса системы, при этом тестировании должны быть проверены следующие элементы:

* Читаемость текста;
* Удобство цветовой схемы;
* Понятность подписей элементов интерфейса.

Результаты тестирования МЭПО показаны на рисунках 8 – 15.

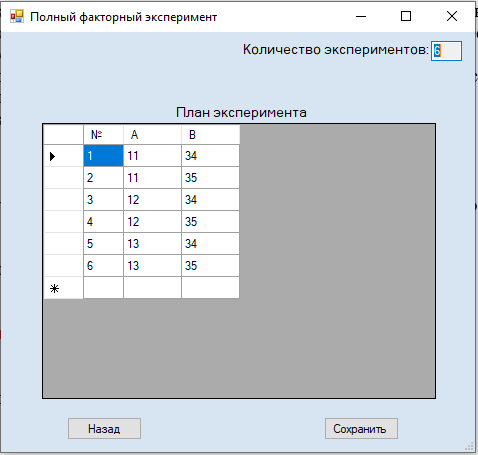


Рисунок 8 – Результат работы МЭПО при заполнении двух факторов и выборе полного факторного эксперимента

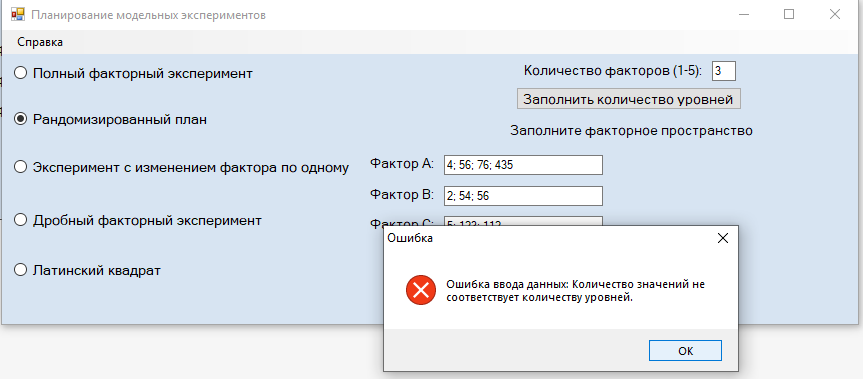


Рисунок 9 – Ошибка при неверном заполнении значений уровней факторного пространства

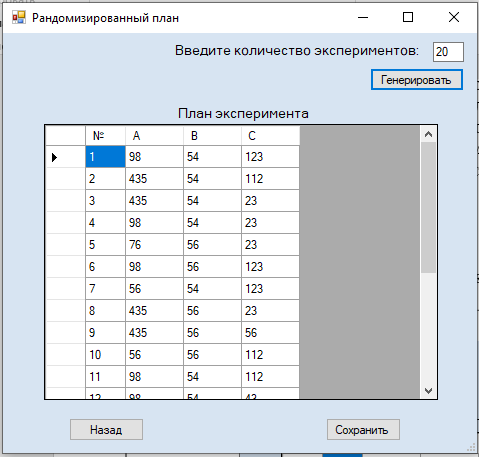


Рисунок 10 – Результат работы МЭПО при заполнении двух факторов и выборе рандомизированного плана

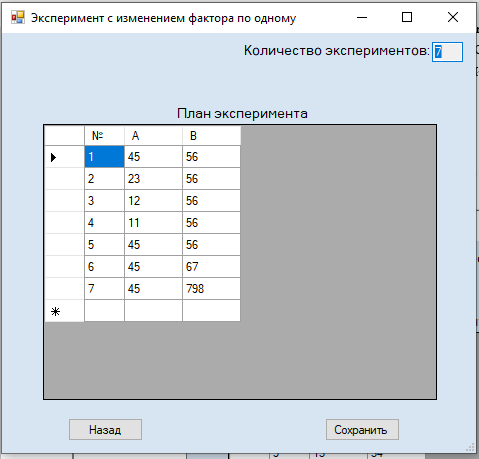


Рисунок 11 – Результат работы МЭПО при заполнении двух факторов и выборе эксперимента с изменением факторов по одному

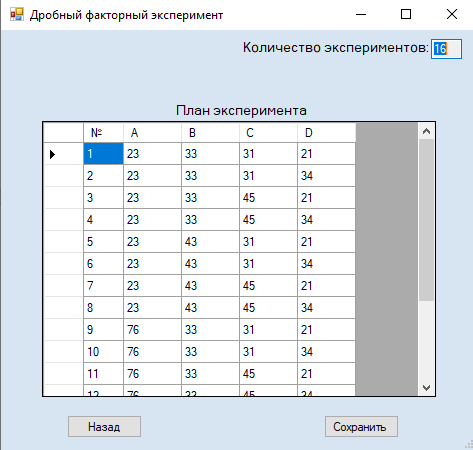


Рисунок 12 – Результат работы МЭПО при заполнении четырёх факторов и выборе дробного факторного эксперимента

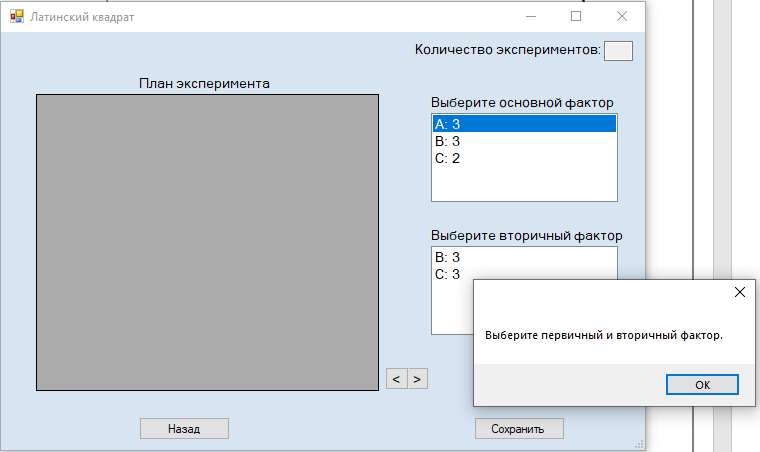


Рисунок 13 – Ошибка при невыбранном вторичном факторе в эксперименте «Латинский квадрат»

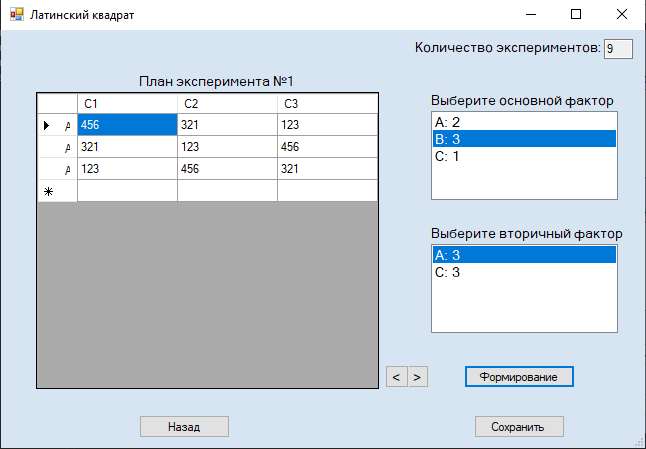


Рисунок 14 – Результат работы МЭПО при заполнении трёх факторов и выборе эксперимента «Латинский квадрат»

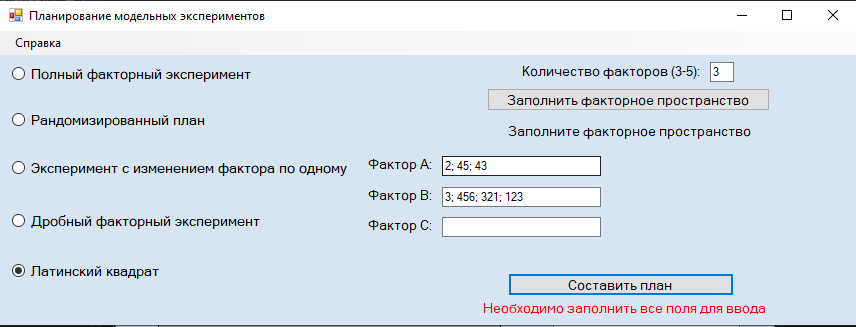


Рисунок 15 – Ошибка при нажатии на кнопку «Составить план», когда заполнены не все поля для ввода

Кроме того, все поля защищены от ввода некорректных данных, то есть:

* пользователь не может ввести количество факторов больше пяти или меньше одного, а при выборе типа плана «Латинский квадрат» меньше трёх;
* в поля для заполнения факторного пространства пользователь не может указать менее одного и более пяти уровней для каждого фактора;
* в окнах, предназначенных для отображения планов пользователю запрещено менять число экспериментов.

## 2.4 Руководство пользователя

[1 Начало работы 29](#_Toc200467747)

[1.1 Основные понятия и термины 29](#_Toc200467748)

[2 Системные требования 30](#_Toc200467749)

[3 Пользовательский интерфейс 30](#_Toc200467750)

[3.1 Главное окно программы 30](#_Toc200467751)

[3.2 Окно «Факторный эксперимент» 31](#_Toc200467752)

[3.3 Окно «Латинский квадрат» 32](#_Toc200467753)

[3.4 Режимы работы 33](#_Toc200467754)

[4 Лучшие практики 33](#_Toc200467755)

[4.1 Начало работы 34](#_Toc200467756)

[4.2 Выбор типа эксперимента 34](#_Toc200467757)

[5 Устранение типовых проблем 34](#_Toc200467758)

[6 Частые вопросы (FAQ) 34](#_Toc200467759)

[7 Контактная информация 35](#_Toc200467760)

1 Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с МЭПО. Перед началом работы, пожалуйста, ознакомьтесь с [системными требованиями](#a9eb003e-8704-4d8f-8d92-8f59944ba6d6) и [лицензионным соглашением](#0343637b-d224-4986-a470-e45222a3137f).

1.1 Основные понятия и термины

Перед началом работы в МЭПО рекомендуем ознакомиться с основными понятиями и терминами:

* Факторное пространство – множество внешних и внутренних параметров модели, значения которых исследователь может контролировать в ходе подготовки и проведения модельного эксперимента;
* Уровень – значение определённого фактора;
* Полный факторный эксперимент представляет из себя все возможные сочетания всех факторов;
* Рандомизированный план строится на основе случайного выбора сочетаний уровней для каждого эксперимента;
* Латинский план («Латинский квадрат») используется в том случае, когда проводится эксперимент с одним первичным фактором и несколькими вторичными. Суть такого планирования состоит в следующем. Если первичный фактор A имеет m уровней, то для каждого вторичного фактора также выбирается m уровней;
* Эксперимент с изменением фактора по одному. Его суть состоит в том, что один из факторов «пробегает» все уровни, а остальные факторы поддерживаются постоянными. Такой план обеспечивает исследование эффектов каждого фактора в отдельности;
* Дробный факторный эксперимент – каждый фактор имеет два уровня – верхний и нижний, из которых строятся сочетания;

1.2 Установка

Для установки МЭПО, пожалуйста, загрузите дистрибутива последней версии 1.0, доступный по адресу <https://github.com/Cy-Nec/ExperimentalPlans>. Для работы программы дополнительно необходимо дополнительно установить среду Microsoft Visual Studio. Перед установкой ознакомьтесь с [системными требованиями](#a9eb003e-8704-4d8f-8d92-8f59944ba6d6) и [лицензионным соглашением](#0343637b-d224-4986-a470-e45222a3137f).

2 Системные требования

Для стабильной и эффективной работы программного обеспечения рекомендуется использовать следующую конфигурацию:

* Частота процессора: 1.2 GHz;
* Количество ядер процессора: от 2 и более;
* Объём оперативной памяти: 2 GB;
* Объём свободного дискового пространства: 0,2 GB;
* Операционная система: Windows 7, 10, 11.

3 Пользовательский интерфейс

3.1 Главное окно программы

Основным окном МЭПО является «Планирование модельных экспериментов», с его помощью пользователь может выбрать тип желаемого эксперимента и заполнить факторное пространство для дальнейшего построение плана.

Окно «Планирование модельных экспериментов» предоставляет возможность выбирать тип формируемого плана и заполнять факторное пространство

Общая структура интерфейса:

* Область выбора типа плана;
* Область ввода факторного пространства:
  1. Поле для ввода количества уровней факторного пространства (от одного до пяти);
  2. Поля для ввода значений уровней каждого фактора. Ввод осуществляется по следующему шаблону: <количество уровней>; <значение уровня>; (Значения уровней вводятся через «; »).

Главное окно МЭПО представлено на рисунке 16.

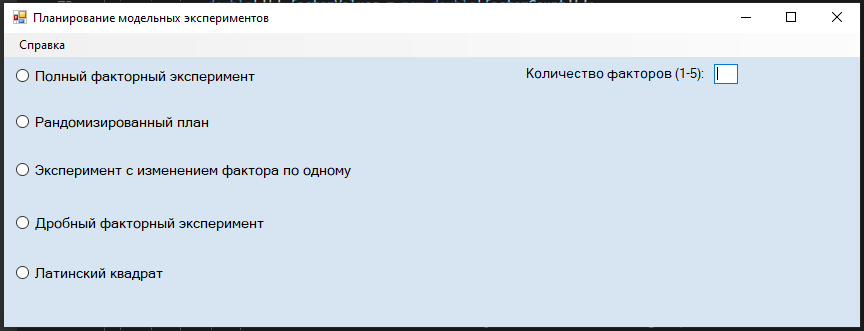


Рисунок 16 – Главное окно МЭПО

3.2 Окно «Факторный эксперимент»

Окно «Факторный эксперимент» предоставляет составленный план во всех режимах работы помимо «Латинский квадрат».

Общая структура интерфейса:

* Область отображения сформированного плана;
* Элементы управления:
* Кнопка для возвращения к главному окну;
* Кнопка для сохранения сформированного плана.

Окно «Факторный эксперимент» представлено на рисунке 17.

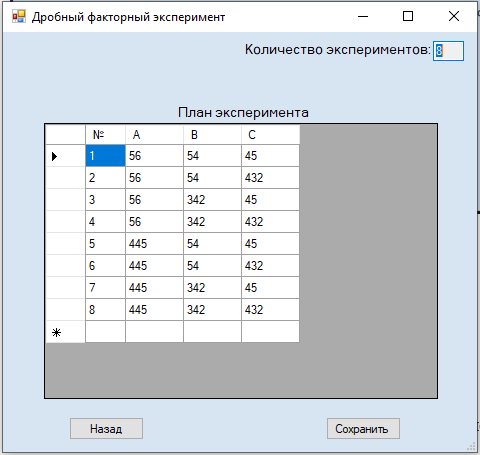


Рисунок 17 ­– Окно «Факторный эксперимент» МЭПО

3.3 Окно «Латинский квадрат»

Окно «Латинский квадрат» предоставляет возможность выбрать первичный и вторичный фактор для формирования плана «Латинский квадрат».

Общая структура интерфейса:

* Область для выбора первичного и вторичного фактора эксперимента;
* Область отображения планов эксперимента;
* Элементы управления:

1. Кнопка для возвращения к главному окну;
2. Кнопка для сохранения сформированных планов;
3. Кнопки для просмотра сформированных планов.

Окно «Латинский эксперимент» представлено на рисунке 18.

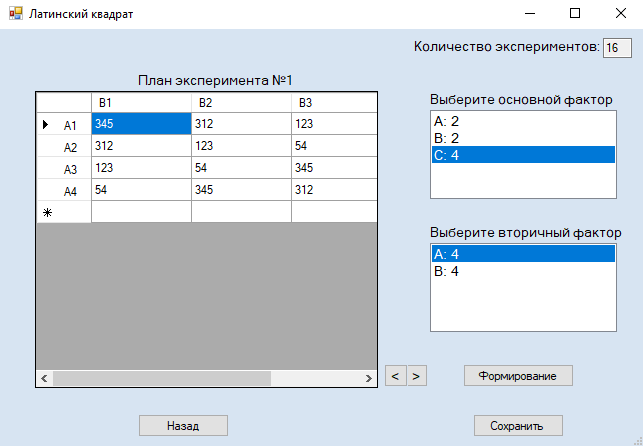


Рисунок 18 – Окно «Латинский эксперимент» МЭПО

3.4 Режимы работы

Пользовательский интерфейс МЭПО обеспечивает работу в нескольких режимах:

3.4.1 Режим 1 «Полный факторный эксперимент»

Режим предоставляет возможность формирования максимально полного плана эксперимента с комбинированием значений всех уровней.

3.4.2 Режим 2 «Рандомизированный факторный эксперимент»

Режим предоставляет возможность формирования плана из случайно подобранных комбинаций значений уровней.

3.4.3 Режим 3 «Эксперимент с изменением факторов по одному»

Режим предоставляет возможность формирования плана, в котором только один фактор меняется за раз, а остальные остаются фиксированными.

3.4.4 Режим 4 «Дробный факторный эксперимент»

Режим предоставляет возможность формирования плана из минимальных и максимальных значений уровня факторного пространства. В этом случае ввод уровней для каждого фактора осуществляется по следующему шаблону: <Минимальное значение>; <Максимальное значение>.

3.4.5 Режим 5 «Латинский квадрат»

Формирование плана, минимизируя влияние сторонних факторов.

4 Лучшие практики

Данный раздел посвящён наиболее эффективным способам применения МЭПО для решения типовых задач.

4.1 Начало работы

* Перед началом работы важно чётко определить:
* Цели эксперимента;
* Значимые факторы;
* Наиболее подходящий тип плана для ваших целей.

4.2 Выбор типа эксперимента

Каждый тип эксперимента имеет свои преимущества и ограничения. Рекомендуется следовать этим рекомендациям:

1. Полный факторный эксперимент:

* Используйте, если количество факторов и уровней невелико.
* Подходит для детального анализа взаимодействий между факторами.
* Лучшая практика: проверьте возможность сокращения количества уровней для уменьшения объёма данных.

1. Рандомизированный факторный эксперимент:

* Применяйте, если необходимо минимизировать влияние внешних факторов.
* Лучшая практика: используйте рандомизацию для всех испытаний, но сохраняйте контроль над базовыми условиями.

1. Эксперимент с изменением факторов по одному:

* Выбирайте этот метод, если требуется простой и быстрый анализ.
* Лучшая практика: убедитесь, что влияние других факторов минимально, чтобы избежать ошибок интерпретации.

1. Дробный факторный эксперимент:

* Используйте, если количество факторов велико, а ресурсы ограничены.

1. Факторный эксперимент «Латинский квадрат»:

* Применяйте, если необходимо учесть два сторонних фактора.

5 Устранение типовых проблем

1. «Неверный результат формирования плана»:

* Причина: ошибка при вводе данных;
* Решение: перепроверить соответствие введённых данных условиям эксперимента.

1. «Программа не запускается»:

* Причина: конфликт с операционной системой или отсутствие зависимостей;
* Решение: убедитесь, что ваша операционная система совместима с программой, а все зависимости установлены.

Если вы не нашли ответа на свой вопрос, пожалуйста, [свяжитесь с нами](file:///C:\Users\Cy_Nec\Downloads\Simaev_23ISP-1.docx#7f77a01d-3465-466f-ba5d-ec49887442b5).

6 Частые вопросы (FAQ)

1. Что такое МЭПО?

МЭПО – программа для планирования и анализа факторных экспериментов. Она помогает создавать планы экспериментов различных типов.

1. Какие типы экспериментов поддерживает программа?

* Полный факторный эксперимент.
* Рандомизированный факторный эксперимент.
* Эксперимент с изменением факторов по одному.
* Дробный факторный эксперимент.
* Факторный эксперимент «Латинский квадрат».

1. Можно ли использовать программу на мобильных устройствах?

На данный момент программа доступна только для компьютеров с операционной системой Windows. Версия для мобильных устройств не планируется.

Если вы не нашли ответа на свой вопрос, пожалуйста, [свяжитесь с нами](file:///C:\Users\Cy_Nec\Downloads\Simaev_23ISP-1.docx#7f77a01d-3465-466f-ba5d-ec49887442b5).

7 Контактная информация

Вы можете направить вопросы по функциональности программы МЭПО следующими способами:

* Email: bron.009@mail.ru;
* Телефон: +7 905-844-51-68;
* Мессенджеры: Telegram.

## 2.5 Рекомендации по внедрению и сопровождению

Внедрение программного обеспечения (ПО) — это процесс интеграции нового ПО в существующие бизнес-процессы и системы работы организации: установка нового ПО или различных автоматизированных систем и адаптация существующей системы под нужды компании. Цель внедрения — обеспечить максимальную производительность и эффективность работы, улучшить качество предоставляемых услуг и оптимизировать бизнес-процессы.

Сопровождение ПО — это процесс улучшения, оптимизации и устранения дефектов ПО после сдачи его в эксплуатацию. В ходе сопровождения в программу вносятся изменения, с тем, чтобы исправить обнаруженные в процессе использования дефекты и недоработки, а также для добавления новой функциональности, с целью повысить удобство использования и применимость ПО.

Элементами сопровождения ПО являются:

* Корректировка ошибок, обнаруженных в работе ПО.
* Обновление и модификация ПО. Улучшение производительности и безопасности программного обеспечения.
* Помощь пользователям в работе с программным обеспечением.
* Мониторинг и анализ работы ПО.

План внедрения МЭПО представлен в таблице 2.

Таблица 2 – План внедрения системы планирования модельных экспериментов

|  |  |  |
| --- | --- | --- |
| Номер этапа | Название этапа | Описание работ по внедрению |
| 1 | Анализ технического и программного обеспечения | Сбор информации о технических и программных средствах персонального компьютера (ПК) пользователя |
| 2 | Установка системы | Проведение установки МЭПО с учётом окружения ПК пользователя |
| 3 | Приёмочное тестирование | Проведение тестирования всех функций установленной МЭПО с целью выявления ошибок в среде пользователя |
| 4 | Обучение пользователя | Проведение консультаций для пользователя |

План сопровождения МЭПО представлен в таблице 3.

Таблица 3 – План сопровождения системы планирования модельных экспериментов

|  |  |  |
| --- | --- | --- |
| Номер этапа | Название этапа | Описание работ по сопровождению |
| 1 | Адаптация к пользователю | Анализ обращений пользователя, выявление и исправление дефектов в работе МЭПО, изменения интерфейса по требованию пользователя. Внесение изменений в документацию пользователя при внесении значимых изменений в интерфейс |
| 2 | Мониторинг работы | Отслеживание времени выполнения функций программы, опираясь на обратную связь от пользователя |
| 3 | Оптимизация | Разработка обновлений для повышения производительности |
| 4 | Помощь пользователю | Предоставление технической поддержки пользователю. Консультирование по вопросам, касающихся работы МЭПО |
| 5 | Окончание сопровождения | Оценка результатов сопровождения по истечению шести месяцев после внедрения |

# Заключение

# Список использованных источников

<https://studfile.net/preview/724637/page:4/>

<https://studfile.net/preview/3073744/page:3/>

<https://infourok.ru/lekciya-metodi-nauchnogo-issledovaniya-621362.html>

<https://docs.cntd.ru/document/1200009493>

<https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F>

<https://cyberleninka.ru/article/n/trebovaniya-k-sovremennym-polzovatelskim-interfeysam/viewer>

<https://www.internet-technologies.ru/blog/solutions/10-pravil-horoshego-polzovatelskogo-interfeysa.html>

<https://visualstudio.microsoft.com/ru/#vs-section>

<https://otus.ru/journal/c-oblasti-primeneniya-pljusy-i-minusy/>

<https://sky.pro/wiki/profession/sravnenie-visual-studio-i-visual-studio-code/>

<https://ru.hexlet.io/blog/posts/vidy-testirovaniya#chto-takoe-testirovanie>

Внедрение [Внедрение программного продукта: основные этапы и потенциальные риски (simbirsoft.com)](https://www.simbirsoft.com/blog/vnedrenie-programmnogo-produkta/)

Сопровождение - [Чем сопровождение программного обеспечения полезно для бизнеса? Ответ от LeanTech.](https://leantech.ai/soprovozhdenie-programmnogo-obespecheniya-dlya-biznesa)

[Сопровождение программного обеспечения — Рувики: Интернет-энциклопедия (ruwiki.ru)](https://ru.ruwiki.ru/wiki/%D0%A1%D0%BE%D0%BF%D1%80%D0%BE%D0%B2%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)

Стандарт оформления, гост на тз, гост на стадии разработки

# Приложение

(обязательное)

**Программный код**