

ОГЛАВЛЕНИЕ

1	СПЕЦИФИКАЦИЯ ПРОГРАММЫ «КУМИР»	5
1.1	Задание.....	5
1.2	Введение	5
1.2.1	Цель.....	5
1.2.2	Область применения	5
1.2.3	Определения, термины и сокращения.....	6
1.2.4	Ссылки.....	6
1.2.5	Обзор.....	6
1.3	Общее описание программного продукта «КуМир»	10
1.3.1	Перспективы продукта	10
1.3.1.1	Системный интерфейс	10
1.3.1.2	Пользовательский интерфейс	11
1.3.1.3	Аппаратный интерфейс	12
1.3.1.4	Программный интерфейс	13
1.3.1.5	Коммуникационные интерфейсы	14
1.3.1.6	Ограничения по памяти	15
1.3.1.7	Операции.....	16
1.3.1.8	Требования по адаптации.....	17
1.3.2	Функции продукта.....	18
1.3.3	Пользовательские характеристики.....	19
1.3.4	Ограничения	20
1.3.5	Предположения и зависимости.....	21
1.3.6	Распределение требований.....	22
1.4	Конкретные требования.....	24
1.5	Сопровождающая информация.....	27
2	ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПРОГРАММЫ «КУМИР»	29
2.1	Введение	29
2.1.1	Название проекта.....	29

2.1.2	Цель и задачи разработки	29
2.1.3	Основания для разработки	30
2.1.4	Термины и сокращения.....	30
2.2	Требования к продукту	32
2.2.1	Функциональные требования.....	32
2.2.2	Нефункциональные требования	33
2.3	Технические требования	35
2.3.1	Аппаратные требования	35
2.3.2	Программные требования.....	36
2.4	Требования к жизненным циклам и обслуживанию.....	37
2.5	Критерии приемки и оценки качества.....	38
2.5.1	Методология тестирования	38
2.5.2	Критерии оценки качества	39
2.6	Заключительные положения.....	40
2.6.1	Срок действия технического задания.....	40
2.6.2	Ответственность сторон	40

1 СПЕЦИФИКАЦИЯ ПРОГРАММЫ «КУМИР»

1.1 Задание

Спецификация — документ, подробно описывающий требования, характеристики и параметры какого-либо объекта, процесса или системы. Он определяет каким должен быть конечный продукт и служит основой для его создания, контроля и стандартизации.

В качестве программного продукта был выбран учебный язык программирования «КуМир».

1.2 Введение

1.2.1 Цель

«КуМир» — это язык и система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе. Основная цель «КуМира» — помочь обучающимся освоить основы алгоритмического подхода и процедурного программирования с использованием простого алголоподобного языка с русской лексикой.

1.2.2 Область применения

Область применения «КуМира» в первую очередь сосредоточена на сфере образования — это обучение основам алгоритмизации и программирования в средней и высшей школе. «КуМир» используется как средство для формирования у школьников алгоритмического мышления и базовых навыков программирования на простом языке с русской лексикой.

Система «КуМир» поддерживает начальные курсы информатики и позволяет создавать, отлаживать и выполнять программы с помощью встроенных графических исполнителей, что делает процесс обучения более наглядным и интерактивным. Она подходит для начального этапа обучения и

помогает подготовить учащихся к более сложным языкам программирования, таким как Python, Pascal, C или C++.

1.2.3 Определения, термины и сокращения

«КуМир» — комплект Учебных МИРов.

Иные особые определения, термины и сокращения отсутствуют.

1.2.4 Ссылки

- Кушниренко, А. Г. Основы информатики и вычислительной техники [Текст] : учеб. пособие для сред. учеб. заведений / А. Г. Кушниренко, Г. В. Лебедев, Р. А. Сворень. — Москва : Просвещение, 1991. — 224 с. — ISBN 5-09-001727-1.
- Официальный сайт системы КуМир [Электронный ресурс] / Разраб. : А. Г. Кушниренко, Г. В. Лебедев, А. Г. Леонов и др. — Электрон. дан. — Москва, 1990–2025. — URL: <https://www.niisi.ru/kumir/> (дата обращения: 23.09.2025).
- Поляков, К. Ю. Информатика. 7–9 классы [Текст] : учебник в 2 ч. / К. Ю. Поляков, Е. А. Еремин. — 3-е изд. — Москва : БИНОМ. Лаборатория знаний, 2023. — Ч. 1. — 304 с. — ISBN 978-5-9963-5802-7.
- Российская электронная школа (РЭШ). Информатика. 7–9 классы [Электронный ресурс] / Минпросвещения России. — Электрон. дан. — Москва, 2018–2025. — URL: <https://resh.edu.ru/subject/9/> (дата обращения: 23.09.2025).

1.2.5 Обзор

Запустив программу, пользователь увидит главное окно, которое разделено на несколько областей (Рисунок 1.1):

- Верхняя панель с меню и инструментами.
- Большое поле для редактирования программы (слева).

- Область построчных сообщений об ошибках и подсказках (справа).
- Нижняя область вывода результатов или взаимодействия с пользователем. Состояние системы отображается в строке состояния внизу окна.

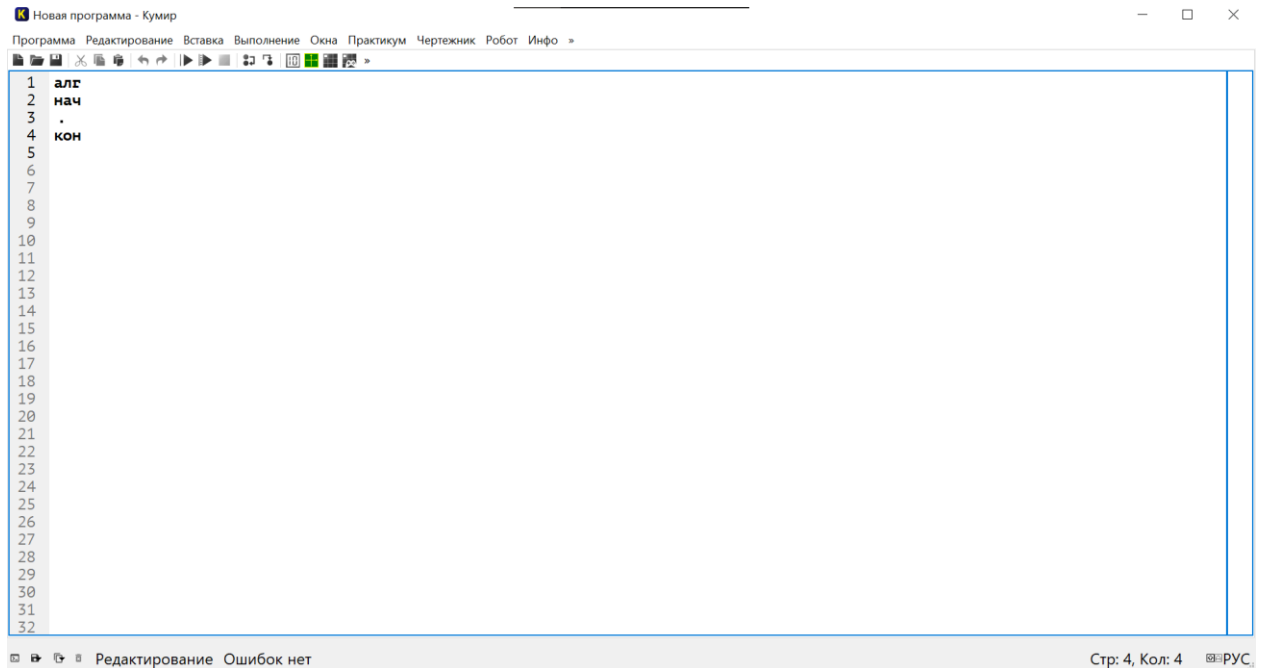


Рисунок 1.1 — Окно при входе

В главном меню есть пункты для создания, сохранения, открытия программ, выполнения программ, отладки и работы с исполнителями.

Основные состояния системы:

- Редактирование — написание и изменение программы, выполнение в этом режиме недоступно.
- Выполнение — программа запущена, редактирование блокируется.
- Анализ результатов — после выполнения можно просмотреть значения переменных и сообщения.
- Пауза — выполнение временно остановлено, можно продолжить или завершить.

Доступные функции и возможности:

- Написание алгоритмических программ на языке с русской лексикой, близком к Алголу.
- Автоматическая проверка синтаксиса, подсветка ошибок в коде.
- Поддержка процедур и функций для структурирования кода.
- Отладка программ с пошаговым выполнением и просмотром значений переменных.
- Встроенные графические исполнители (Робот, Чертежник, Водолей), которые работают на игровом поле из квадратных клеток:
 - Робот может передвигаться, рисовать, обнаруживать стены и цвета. Доступные команды: вверх, вниз, влево, вправо, закрасить.
 - Чертежник может передвигаться, оставляя след. Доступные команды: вверх, вниз, влево, вправо, опустить перо, поднять перо.
 - Водолей может работать с переменными. Доступные команды: ввод, вывод, арифметические операции.
- Взаимодействие с исполнителями через панель управления.
- Сохранение и загрузка программ.
- Возможность написать программы для решения учебных задач, освоения логического и алгоритмического мышления.

Что можно сделать в «КуМир»:

- Учить основы программирования и алгоритмизации.
- Создавать и запускать учебные программы с визуальной отладкой.
- Управлять виртуальными исполнителями и моделировать их поведение.
- Писать программы с использованием циклов, условий, функций и процедур.
- Разрабатывать простые игры и анимации.

- Изучать механику выполнения программ благодаря пошаговому исполнению и визуализации.

1.3 Общее описание программного продукта «КуМир»

1.3.1 Перспективы продукта

1.3.1.1 Системный интерфейс

Системные интерфейсы программы «КуМир» в текущий момент представляют собой комплексное многооконное графическое окружение с главным меню, панелью инструментов, рабочими окнами для редактирования, вывода результатов и сообщений, а также строкой состояния. Интерфейс построен с использованием библиотек Qt и Cocos2D, что обеспечивает мультиплатформенность и переносимость между Windows, Linux, macOS и мобильными платформами.

Основные особенности и компоненты системного интерфейса «КуМир»:

- Главное окно с разделением на несколько областей: редактор программ (ввод текста), окно вывода вычислений и системных сообщений, панель управления исполнителями.
- Визуализация изменений состояния данных и логических операций непосредственно на полях программы.
- Поддержка графических исполнителей с собственными пультами управления и визуализацией.
- Реализация многооконного режима с фиксированным расположением рабочих окон, что удобно для больших экранов, и с наложением окон с прозрачностью для устройств с малым экраном.
- Использование объектно-ориентированного подхода и мультиплатформенных библиотек (Qt, Cocos2D) для упрощения поддержки и развития интерфейса.

Перспективы развития системных интерфейсов «КуМир» включают:

- Расширение мультиплатформенности с поддержкой большего числа устройств и ОС, включая мобильные и планшеты.

- Усиление сетевого функционала с улучшенной синхронизацией и возможностью совместной работы с программами и исполнителями в сети.
- Более гибкая и адаптивная организация оконного интерфейса, учитывающая особенности разных экранов и пользовательских предпочтений.
- Добавление новых видов визуализации и эффектов (например, физика частиц, анимации дождя, снега, фейерверков) для повышения наглядности учебного процесса.
- Возможная интеграция современных элементов UX/UI дизайна для улучшения удобства и интуитивности работы с системой.

1.3.1.2 Пользовательский интерфейс

Пользовательские интерфейсы программы «КуМир» в настоящее время включают:

- Главный редактор кода с поддержкой написания программ на русском языке с автоматическим синтаксическим разбором и подсветкой ошибок.
- Окно вывода результатов вычисления и отображения системных сообщений.
- Панель управления графическими исполнителями, позволяющая управлять виртуальными объектами в интерактивном режиме.
- Многооконный интерфейс с разделением на области для кода, вывода, ввода данных и состояния системы.
- Меню с разделами для создания, сохранения, запуска, отладки и настройки программ.
- Встроенные средства отладки, включая пошаговое выполнение программ и просмотр значений переменных.

Ожидаемое развитие пользовательских интерфейсов «КуМир» связано с актуальными трендами UX/UI и включает:

- Повышение адаптивности интерфейса под различные устройства: ПК, планшеты, смартфоны.
- Интеграция голосового и жестового управления для расширения возможностей взаимодействия.
- Улучшение визуализации и анимаций для более понятного и увлекательного обучения.
- Персонализация пользовательского опыта с помощью ИИ, подстраивающего интерфейс под индивидуальные потребности.
- Повышение доступности и инклюзивности интерфейса для пользователей с ограниченными возможностями.
- Внедрение современных элементов микровзаимодействия для повышения отзывчивости и удобства.

1.3.1.3 Аппаратный интерфейс

Сейчас аппаратные интерфейсы программы «КуМир» включают классический оконный интерфейс с разделением на несколько рабочих областей:

- Главное окно с заголовком, строкой меню и полудинамической инструментальной панелью.
- Три рабочих окна: левое верхнее для ввода текста программы, правое для отображения результатов выполнения и сообщений об ошибках, нижнее для ввода/вывода данных и системных сообщений.
- Аппаратно-программное взаимодействие с виртуальными исполнителями через специальные пульты управления.
- Строка состояния и множество горячих клавиш для удобства управления.
- Интерфейс ориентирован на управление с помощью клавиатуры и мыши.

Перспективы развития аппаратных интерфейсов «КуМир»:

- Адаптация под сенсорные экраны и мультитач жесты для удобного использования на планшетах и мобильных устройствах.
- Введение голосового управления и поддержки жестового ввода для расширения возможностей взаимодействия.
- Использование современных технологий отслеживания взгляда и биометрических данных для адаптивного изменения интерфейса.
- Поддержка VR/AR-устройств для создания более иммерсивных и наглядных учебных сред.
- Интеграция с разнообразными образовательными гаджетами и IoT-устройствами.
- Улучшение эргономики и удобства управления, в том числе расширение поддержки горячих клавиш, панелей и настройки интерфейса под пользователя.

1.3.1.4 Программный интерфейс

Программные интерфейсы «КуМир» в настоящее время включают:

- Главное окно с разделением на несколько рабочих областей: редактор программ (ввод кода), окно вывода результатов и сообщений об ошибках, окно ввода/вывода данных.
- Меню и панель инструментов с множеством горячих клавиш для создания, редактирования, сохранения, запуска и отладки программ.
- Подсветка синтаксиса и автоматический синтаксический разбор с сообщениями об ошибках.
- Отладка программ с пошаговым выполнением и просмотром значений переменных.
- Встроенная поддержка графических исполнителей с панелями управления и визуализацией алгоритмов.

- Многооконный, интуитивный интерфейс, обеспечивающий удобство учебного процесса.

Перспективы развития программных интерфейсов «КуМир»:

- Адаптивность интерфейса под разные устройства: компьютеры, планшеты, мобильные устройства.
- Интеграция голосового и жестового управления для расширения способов взаимодействия.
- Использование искусственного интеллекта для персонализации и помощи в обучении.
- Более богатая визуализация с 3D-графикой и анимациями для повышения наглядности учебных материалов.
- Повышение доступности и инклюзивности для пользователей с различными возможностями.
- Реализация современных микровзаимодействий и элементов UX/UI для удобства и интуитивности.

1.3.1.5 Коммуникационные интерфейсы

Коммуникационные интерфейсы программы «КуМир» на сегодняшний день включают базовые средства взаимодействия пользователя с системой: ввод программного текста, управление графическими исполнителями, вывод результатов выполнения и сообщений, а также базовые функции сохранения, загрузки и обмена программами. Основной акцент сделан на локальном взаимодействии с программой.

Перспективы развития коммуникационных интерфейсов «КуМир» в будущем связаны с внедрением современных цифровых и AI-технологий, которые делают коммуникацию более гибкой, персонализированной и мультимедийной:

- Интеграция искусственного интеллекта для создания умных ассистентов, которые смогут помогать в обучении, отвечать на

вопросы и адаптировать учебный процесс под индивидуальные особенности пользователя.

- Расширение возможностей по обмену учебными материалами и проектами через облачные сервисы и сетевые платформы с поддержкой совместной работы в реальном времени.
- Введение голосовых, жестовых и биометрических интерфейсов для более естественного и многообразного взаимодействия.
- Генерация визуального, аудиального и текстового контента с помощью нейросетей для повышения наглядности и вовлеченности.
- Адаптация коммуникационных каналов под разные устройства с поддержкой мобильных платформ, планшетов и VR/AR технологий.
- Улучшение персонализации контента и интерфейса на основе анализа поведения и предпочтений пользователей.

1.3.1.6 Ограничения по памяти

В программе «КуМир» пока нет жестко фиксированного ограничения по памяти в явном виде, поскольку система ориентирована на учебные задачи с относительно небольшими программами и моделями. Однако используется стековое выполнение байткода, что ограничивает максимально возможный размер вызываемых процедур и глубину рекурсии.

В реальности ограничение по памяти и размерам программ вытекает из архитектуры виртуальной машины и объёмов выделяемой памяти в операционной системе. Это достаточно традиционный подход для языков с интерпретатором и учебных сред.

Перспектив развития не прогнозируется в следующих версиях.

1.3.1.7 Операции

В программе «КуМир» сейчас доступны следующие основные операции и команды на алгоритмическом языке:

- Арифметические операции: сложение (+), вычитание (-), умножение (*), деление (/), возведение в степень (**).
- Логические операции: и, или, не, а также логические константы "да" и "нет".
- Операции работы с текстом: конкатенация строк, взятие подстроки.
- Операции присваивания значений переменным.
- Ввод и вывод данных.
- Управляющие конструкции: условные операторы (если, то, иначе), циклы (пока, для), переходы.
- Возможность использовать процедуры и функции для структурирования кода.
- Поддержка работы с массивами и таблицами.
- Отдельные команды для управления графическими исполнителями (Робот, Черепаха) — движение, поворот, рисование и др.

Ожидается развитие операций в сторону:

- Расширения набора встроенных функций и операторов для работы с более сложными структурами данных.
- Улучшения поддержки процедур и функций, включая рекурсию и вспомогательные алгоритмы.
- Внедрения новых типов данных и расширения возможностей работы с текстом и графикой.
- Оптимизации и улучшения синтаксиса для повышения удобства и читаемости кода.

- Возможной интеграции с современными библиотеками и API для расширения функциональности.

1.3.1.8 Требования по адаптации

Сейчас программа «КуМир» имеет следующие основные требования по адаптации и системе:

- «КуМир» работает под операционными системами Windows и Linux, поддерживает установку на разные версии этих ОС.
- Для установки рекомендуется использовать папку по умолчанию в директории "Program Files" или создавать отдельную папку для «КуМир», чтобы избежать потери данных при удалении программы.
- Программа рассчитана на работу с оборудованием школьных и учебных компьютеров средней мощности — специфика учебного ПО не требует мощных ресурсов.
- Интерфейс и функционал ориентированы на удобство использования школьниками и преподавателями, с учётом эргономики и простоты.
- В системе предусмотрена поддержка графических исполнителей, работающих на уровнях "игры" и "алгоритм", что требует от устройства достаточной графической и вычислительной мощности.

Перспективы и развитие требований по адаптации «КуМир»:

- Расширение поддержки новых операционных систем, включая современные версии Windows, Linux-дистрибутивов и, возможно, macOS.
- Усиление кроссплатформенности с акцентом на мобильные и планшетные устройства.
- Оптимизация ресурсов для работы на более слабых или специализированных учебных компьютерах и устройствах.

- Улучшение адаптивности интерфейса под различные экраны и устройства ввода.
- Повышение возможностей интеграции с облачными сервисами и удалённым хранением данных.
- Актуализация системных требований в соответствии с современными образовательными стандартами и технологиями.

1.3.2 Функции продукта

Функции программного продукта «КуМир» включают следующие основные возможности:

- Создание, редактирование и выполнение программ на алгоритмическом языке с русской лексикой.
- Автоматический синтаксический разбор программ и подсветка ошибок в процессе написания кода.
- Поддержка структурного программирования: процедуры, функции, циклы, условия, массивы, таблицы.
- Интерактивное управление встроенными графическими исполнителями, действия которых визуализируются на игровом поле.
- Возможность отладки программ с покадровым выполнением и просмотром текущих значений переменных.
- Поддержка ввода и вывода данных в процессе выполнения программы.
- Создание учебных и практических упражнений для изучения алгоритмического мышления и основ программирования.
- Многооконный интерфейс для одновременного отображения исходного кода, результатов выполнения и взаимодействия с исполнителями.

- Кроссплатформенность — работа под Windows и Linux с открытым исходным кодом.
- Использование в образовательных учреждениях как средство обучения основам информатики и программирования.

1.3.3 Пользовательские характеристики

Пользовательские характеристики программы «КуМир» включают:

- Русскоязычный интерфейс и язык программирования с понятной и близкой к естественному языку лексикой, что облегчает освоение и использование.
- Интерактивный и визуальный подход к обучению: программа содержит встроенных графических исполнителей (Робот, Черепаха, Кузнечик, Водолей, Вертун, Чертёжник, Рисователь), которые визуализируют алгоритмы и позволяют работать как в программируемом, так и в интерактивном режиме.
- Система автоматического синтаксического разбора и диагностики ошибок во время написания кода, с подсветкой и комментариями прямо на полях редактора.
- Возможность пошагового выполнения программ с отслеживанием текущих значений переменных, что помогает лучше понять работу алгоритмов.
- Многооконный интерфейс с областями для ввода кода, вывода результатов и сообщений, а также панелью управления исполнителями.
- Кроссплатформенность — поддержка операционных систем Windows и Linux.
- Высокая адаптивность интерфейса к уровню пользователя: от новичков до учеников средних классов.
- Поддержка горячих клавиш и меню для удобства и быстрого доступа к функциям.

- Широкий набор учебных материалов, методических рекомендаций и программных примеров.
- Программа предназначена для использования в средних и высших школах, а также в академическом образовании для начального уровня программирования.

1.3.4 Ограничения

Основные ограничения программы «КуМир»:

- Низкая скорость выполнения команд и сложных программ, что связано с интерпретируемой природой языка.
- Отсутствие классической графики — визуализация ограничена встроенными графическими исполнителями.
- Недостаток методических материалов, справочников и описаний, затрудняющих самостоятельное освоение.
- Ограниченная информация о состоянии переменных при пошаговом выполнении — отображается только факт присвоения, а текущее значение не всегда очевидно.
- Язык программирования исключительно на русском языке, что ограничивает его применение вне учебных целей.
- Необходимость частого переключения раскладки клавиатуры при написании программ.
- Ограниченные возможности для углубленного изучения программирования и перехода к современным языкам.
- Программа ориентирована преимущественно на начальный уровень и не подходит для сложных проектов или профессионального программирования.
- Ограничения по типам параметров и их значениям в функциях и процедурах.

- Ограниченное применение в профильном обучении и вузах, где используются более мощные языки программирования.

1.3.5 Предположения и зависимости

Предположения:

- Основной целевой пользователь — школьник или начинающий программист, изучающий основы алгоритмического мышления и программирования.
- «КуМир» предназначен для работы в учебной среде и не рассчитан на профессиональное программирование.
- Пользователь знаком с принципами работы с компьютером, умеет вводить текст и взаимодействовать с графическими интерфейсами.
- Язык программирования «КуМир» использует русскоязычную лексику, предположительно для русскоязычных пользователей.
- Система предполагает использование базовых операционных систем: Windows, Linux, а также поддержку X11 для графических интерфейсов.
- Ожидается, что программная среда будет работать в рамках ограниченных учебных заданий, не требующих высокой производительности и больших ресурсов.

Зависимости:

- Зависимость от операционной системы (Windows, Linux) и наличия совместимой графической среды (X11).
- Использование внешних библиотек и фреймворков, таких как Qt и Cocos2D для интерфейсов и визуализации.
- Возможность использования различных графических исполнителей (Робот, Черепаха, Водолей, Чертёжник и др.) требует наличия функционала визуализации и управления виртуальными исполнителями.

- Зависимость от стандартов и требований образовательных учреждений и экзаменационных комиссий (например, ЕГЭ по информатике).
- Возможность интеграции с другими языками программирования, такими как Python, для расширения функционала и удобства обучения.

1.3.6 Распределение требований

Функциональные требования:

- Обеспечение возможности создавать, редактировать и запускать программы на русском алгоритмическом языке.
- Поддержка структурного и процедурного программирования с процедурами, функциями, циклами и условиями.
- Интерактивное управление графическими исполнителями с визуализацией результатов.
- Автоматическая проверка синтаксиса и подсветка ошибок в режиме реального времени.
- Поддержка отладки программ с пошаговым выполнением и просмотром значений переменных.
- Многооконный удобный интерфейс с разделением областей на ввод, вывод и управление.

Нефункциональные требования:

- Кроссплатформенность: поддержка Windows и Linux.
- Удобство и понятность интерфейса для целевой аудитории — школьников и начинающих.
- Стабильность и производительность, достаточные для учебных целей.
- Безопасность и устойчивость к ошибкам пользователя.
- Возможность масштабирования и расширения функционала.

- Легкость установки и обновления программы.

Приоритеты и уровни требований:

- Требования must-have (обязательные): базовые функции программирования, визуальные исполнители, базовая отладка и диагностика.
- Требования should-have (желательные): расширенные средства визуализации, улучшенная работа с большими проектами, голосовое управление.
- Требования could-have (возможные): интеграция с облачными сервисами, поддержка VR/AR, AI-ассистенты для обучения.
- Требования won't-have (незапланированные): профессиональные среды разработки или поддержка низкоуровневого программирования.

1.4 Конкретные требования

1. Создание и редактирование программ.

Функция: позволяет пользователю писать программы на русском алгоритмическом языке с использованием ключевых слов, процедур, циклов и условий.

Требования:

- Поддержка синтаксиса алгоритмического языка с русскими ключевыми словами.
- Автоматическая подсветка синтаксиса.
- Выделение и сообщение об ошибках в процессе редактирования.
- Возможность вставки комментариев.
- Удобство и простота интерфейса для начинающих пользователей.

2. Запуск и выполнение программ.

Функция: интерпретация и выполнение написанных программ, включая последовательное выполнение команд, циклов и условий.

Требования:

- Корректное выполнение алгоритмов согласно заданной логике.
- Обработка нестандартных ситуаций (деление на ноль, выход за границы массива).
- Вывод результатов выполнения в отдельном окне или консоли.
- Визуальная индикация этапов выполнения программы.

3. Отладка программ.

Функция: пошаговое выполнение кода с возможностью просмотра текущих значений переменных, установки точек останова.

Требования:

- Возможность запуска программы покомандно.
- Отображение значений переменных в реальном времени.
- Удобный интерфейс для переключения между режимами выполнения и редактирования.

4. Управление графическими исполнителями.

Функция: Управление встроенными виртуальными исполнителями (Робот, Черепаха, Кузнечик и др.) для визуализации алгоритмов на игровом поле.

Требования:

- Поддержка команд передвижения, рисования и проверки условий для исполнителей.
- Реализация интерактивного управления через программы.
- Анимация и визуализация действий исполнителей в режиме реального времени.

5. Работа с процедурами и функциями.

Функция: Возможность создавать процедуры и функции с аргументами и типами возвращаемых значений для организации кода.

Требования:

- Поддержка определения процедур и функций с параметрами.
- Корректная обработка вызовов и возвратов из функций.
- Возможность рекурсии.
- Обеспечение типовой безопасности и контроля параметров.

6. Работа с переменными и типами данных.

Функция: Определение и использование различных типов данных (целые, вещественные, логические, символьные, литерные) и массивов.

Требования:

- Поддержка объявления переменных и таблиц (массивов).
- Типизация данных с возможностью проверки во время компиляции/интерпретации.
- Поддержка операций с базовыми типами и таблицами.

7. Ввод и вывод данных.

Функция: Организация ввода данных пользователем и вывода результатов программы на экран.

Требования:

- Поддержка команд ввода и вывода с возможностью работы с переменными и выражениями.
- Обработка ошибок ввода.
- Удобство отображения результатов.

8. Системные функции и управление программой.

Функция: Управление процессом работы программы: создание новых файлов, сохранение, загрузка, импорт/экспорт.

Требования:

- Безопасное сохранение и загрузка программ.
- Поддержка форматов файлов, используемых в «КуМир».
- Удобный пользовательский интерфейс для доступа к этим функциям.

1.5 Сопровождающая информация

Основные компоненты сопровождающей информации:

- **Официальная документация:**
 - В комплекте поставки «КуМир» входит справочники и инструкции, охватывающие процесс установки, использования среды, написания программ, а также описание языка программирования на русском алгоритмическом языке.
 - Документация подробно объясняет синтаксис, особенности команд, работу с графическими исполнителями (Робот, Черепаха и др.).
 - В ней есть разделы с описанием методов отладки, структурирования программ, а также разбором примеров.
- **Методические материалы:**
 - Учебные пособия и методические рекомендации для учителей и учеников, где содержатся пошаговые инструкции и задания для практики.
 - Включают задачи разного уровня сложности, от простейших упражнений до комплексных проектов, позволяющих освоить алгоритмическое мышление и программирование.
- **Видеоуроки и обучающие курсы:**
 - Видео с лекциями и практическими занятиями знакомят пользователей с интерфейсом «КуМир», способами управления исполнителями и основами программирования.
 - Позволяют визуально воспринять процесс обучения, что особенно полезно новичкам.
- **Справочный раздел в программе:**

- Внутри самой среды «КуМир» есть раздел помощи или информационное меню, где можно быстро получить ответы на основные вопросы по использованию.
- Интернет-ресурсы и сообщества:
 - Сайт проекта «КуМир» (НИИСИ РАН) предлагает скачать актуальные версии программы и руководств.
 - Пользователи могут обращаться к форумам, образовательным порталам и поддержке для обмена опытом и получения помощи.

2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПРОГРАММЫ «КУМИР»

2.1 Введение

2.1.1 Название проекта

Разработка и совершенствование учебного языка программирования «КуМир».

2.1.2 Цель и задачи разработки

Цели разработки:

- Создание и поддержка современной, удобной и доступной учебной среды для обучения основам программирования и алгоритмизации на русском языке, ориентированной на школьников и начинающих пользователей.
- Обеспечение развития педагогических и технических возможностей среды для повышения качества и эффективности обучения информатике и программированию.
- Создание платформы, позволяющей плавно перейти к более сложным языкам программирования, таким как Python, Pascal, C/C++.

Задачи разработки:

- Обновление и совершенствование языка программирования с сохранением простоты, наглядности и русскоязычной лексики для легкости восприятия.
- Разработка функционала визуальных исполнителей (Робот, Черепаха, Кузнечик, Водолей и др.) с расширенными возможностями интерактивного управления и визуализации алгоритмов.
- Улучшение пользовательского интерфейса с учетом современных требований UX/UI, адаптивности под различные устройства и ОС.

- Разработка встроенных средств отладки, визуального анализа и обучения с пошаговым выполнением и инспекцией данных.
- Модернизация документации и обучающих материалов, включая видеоуроки и методические пособия для преподавателей и учащихся.
- Расширение кроссплатформенной поддержки с акцентом на Windows и Linux, а также возможной поддержкой мобильных платформ.
- Обеспечение лицензирования и открытости исходного кода для поддержки образовательных учреждений и сообществ.

2.1.3 Основания для разработки

Необходимость поддержки и развития проверенной практикой образовательной среды, адаптации ее под современные требования и технологии, а также обеспечение доступности и эффективности обучения в рамках школьного и начального вузовского образования.

2.1.4 Термины и сокращения

Термины и сокращения представлены в таблице 2.1.

Таблица 2.1 — Термины и сокращения.

Термин/Сокращение	Расшифровка	Описание
«КуМир»	Комплект Учебных МИРов	Учебный алгоритмический язык и среда программирования
Алгоритм	-	Конкретная последовательность действий для решения задачи
Исполнитель	-	Виртуальный объект (например, Робот), выполняющий команды

Робот	Встроенный исполнитель «КуМир»	Один из графических исполнителей в среде «КуМир»
Процедура	-	Блок программного кода, выполняющий определённую задачу без возвращения значения
ФУНКЦИЯ	-	Блок программного кода, возвращающий некоторое значение
МАССИВ	-	Структура данных для хранения набора однотипных значений
UI	User Interface	Пользовательский интерфейс
ОС	Операционная система	Программное обеспечение, управляющее аппаратными ресурсами
IDE	Integrated Development Environment	Интегрированная среда разработки
ОГЭ	Основной государственный экзамен	Обязательный экзамен для школьников
RFC	Request For Comments	Документ стандартизации или технического предложения
API	Application Programming Interface	Интерфейс программирования приложений
GNU GPL	GNU General Public License	Свободная лицензия на программное обеспечение

2.2 Требования к продукту

2.2.1 Функциональные требования

- Поддержка алгоритмического языка с русской лексикой.
 - Обеспечение возможности написания программ с использованием понятного русскоязычного синтаксиса, что облегчает освоение основ программирования школьниками.
 - Поддержка ключевых команд: присваивание, условные операторы (если, то, иначе), циклы (пока, для), вызовы процедур и функций.
- Встроенные графические исполнители.
 - Реализация визуальных объектов (Робот, Черепаха, Кузнечик, Водолей, Вертун, Чертёжник, Рисователь), которые выполняют команды и визуализируют алгоритмы в интерактивной среде.
 - Поддержка управления исполнителями средствами самого языка.
- Работа с процедурами и алгоритмами-функциями.
 - Возможность объявления и вызова процедур (блоков кода без возвращаемого значения) и функций (блоков кода с возвращаемым значением через служебную переменную "знач").
 - Поддержка передачи параметров и возврата результатов.
- Типизация и работа с переменными.
 - Обеспечение поддержки различных типов данных: целые, вещественные, логические, символьные, литерные.
 - Возможность определения массивов (таблиц) и работы с ними.
- Система ввода и вывода данных.

- Поддержка команд пользовательского ввода данных в программу и вывода результатов работы.
- Редактор и проверка синтаксиса.
 - Подсветка синтаксиса при написании кода.
 - Автоматическая проверка и сообщение об ошибках в режиме реального времени.
- Отладка и исполнение.
 - Возможность покадрового (пошагового) исполнения программы с просмотром текущих значений переменных.
 - Обработка исключений и сообщений об ошибках времени выполнения.
- Работа с управляющими конструкциями.
 - Реализация условных операторов, циклов и операторов перехода.
 - Поддержка вложенности и логических выражений.
- Кроссплатформенность и интеграция.
 - Обеспечение стабильной работы под Windows и Linux.
 - Возможность расширения функционала через модули и интеграцию с внешними языками (перспективно).
- Обучающие и методические возможности.
 - Встроенная поддержка учебных материалов и заданий.
 - Просмотр пошагового выполнения алгоритмов для лучшего понимания.

2.2.2 Нефункциональные требования

- Удобство использования:
 - Интерфейс должен быть интуитивно понятным и адаптированным для школьников и начинающих пользователей.

- Обеспечение быстрого доступа к основным функциям без лишних сложностей.
- Производительность:
 - Среда должна корректно и быстро выполнять учебные программы, обеспечивая плавное взаимодействие с пользователем.
 - Быстрая реакция редактора при наборе кода и подсветке синтаксиса.
- Стабильность и отказоустойчивость:
 - Программа должна устойчиво работать без сбоев и потерь данных при различных сценариях использования и ошибках пользователя.
 - Корректная обработка ошибок и информирование пользователя о них.
- Кроссплатформенность:
 - Поддержка основных операционных систем: Windows и Linux.
 - Возможность установки и запуска на различном аппаратном обеспечении, включая слабые и старые компьютеры.
- Безопасность:
 - Ограничение доступа к критичным ресурсам операционной системы.
 - Минимизация риска повреждения данных пользователя и системных файлов.
- Масштабируемость и расширяемость:
 - Возможность добавления новых функциональных модулей и расширений в будущем.
 - Архитектура должна позволять простую интеграцию новых исполнителей и команд.
- Локализация и поддержка языка:

- Русскоязычный интерфейс и документация с возможностью расширения на другие языки (по необходимости).
- Поддержка русскоязычного учебного контента и справочных материалов.
- Документированность и методическая поддержка:
 - Наличие подробной документации для пользователей и преподавателей.
 - Обновляемые методические материалы и обучающие пособия.
- Совместимость с образовательными стандартами:
 - Поддержка требований Федерального государственного образовательного стандарта (ФГОС).
 - Соответствие критериям подготовки к ОГЭ и ЕГЭ по информатике.

2.3 Технические требования

2.3.1 Аппаратные требования

- Процессор:
 - Совместимость с процессорами архитектуры x86 и x86-64, а также с ARM (например, Raspberry Pi).
 - Минимальная частота процессора — от 1 ГГц, рекомендовано — более современные процессоры для плавной работы.
- Оперативная память (RAM):
 - Минимум 512 МБ оперативной памяти.
 - Рекомендуются не менее 1 ГБ для комфортной работы с интерфейсом и выполнения учебных программ.
- Место на жестком диске или SSD:

- Минимум 200 МБ свободного пространства для установки программы и хранения учебных проектов.
- Рекомендуется иметь запас дополнительного места при работе с большими проектами.
- Операционная система:
 - Поддержка основных настольных ОС: Windows (версии от Windows 7 и выше) и Linux (рекомендуемые дистрибутивы с поддержкой X11).
 - Возможность работы на некоторых мобильных платформах с доработкой (например, Android через NDK).
- Видео и графика:
 - Поддержка стандартных графических режимов для отображения визуальных исполнителей.
 - Минимум — видеокарта с поддержкой разрешения 1024x768 и выше.
- Ввод-вывод:
 - Стандартная клавиатура и мышь для взаимодействия пользователя с интерфейсом программы.
- Дополнительные требования:
 - Поддержка Unicode и кириллических символов для корректного отображения русскоязычного текста.
 - Отсутствие необходимости в сторонних библиотеках (например, Qt), что снижает системные требования и упрощает портирование.

2.3.2 Программные требования

- Операционные системы:
 - Поддержка современных версий Windows (не ниже Windows 7) и основных Linux-дистрибутивов с графической средой (X11).

- Возможность установки и запуска без дополнительных конфликтов и требований к нестандартному ПО.
- Нет строгих требований к дополнительным программам.

2.4 Требования к жизненным циклам и обслуживанию

- Установка программы:
 - Среда программирования «КуМир» должна предоставлять удобный и понятный процесс установки, адаптированный для Windows (версии 7 и выше) и Linux.
 - В Windows процесс установки должен включать стандартный мастер с выбором компонентов, папки установки (по умолчанию в "Program Files"), создания ярлыков и подтверждения лицензии.
 - В Linux установка должна поддерживать установку из системных репозиториях (например, пакет kumir2), через Центр приложений или Synaptic, а также возможность установки из исходных кодов.
 - Установка должна предполагать проверку и необходимые предупреждения о правах доступа к папкам для предотвращения потери данных при удалении программы.
- Настройка средовых параметров:
 - После установки должна быть возможность базовой настройки интерфейса: выбор языкового пакета, оформление, окна сообщений и инструментов.
 - Конфигурация параметров исполнения программ, таких как выбор исполняющих модулей, настройка путей к учебным материалам, и управление обновлениями.
 - Возможность интеграции с дополнительными ресурсами (учебными курсами, справочными материалами).
- Обновление и поддержка:

- Система должна поддерживать обновления с возможностью автоматической загрузки и установки новых версий.
- Обеспечение обратной совместимости с проектами, созданными в предыдущих версиях среды.
- Удаление (деинсталляция):
 - Деинсталляция должна производиться через стандартные средства ОС, при этом гарантируя корректное удаление файлов «КуМир», но сохраняя пользовательские данные, если они хранятся вне папки установки.

2.5 Критерии приемки и оценки качества

2.5.1 Методология тестирования

- Автоматизированное тестирование:
 - Использование встроенных команд для проверки правильности выполнения алгоритмов с контрольными значениями.
 - Тестирование проводится с использованием подготовленных файлов с условиями, которые ученик не видит, изменяя только разрешенные области.
 - Запуск тестов из меню среды или горячими клавишами; успешное прохождение фиксируется сообщением "Тестирование завершено успешно".
 - При ошибках тест прерывается с соответствующим сообщением.
- Пошаговое выполнение и отладка:
 - Режим пошагового выполнения с отображением текущих значений переменных и проверок условий.
 - Визуализация каждой операции в окне результатов.

- Возможность отследить порядок исполнения алгоритма и выявить логические ошибки.
- Проверка синтаксиса и ошибок времени выполнения:
 - Среда автоматически проверяет синтаксис на этапе редактирования, предупреждая о некорректных конструкциях.
 - Во время исполнения фиксируются ошибки неопределенных переменных, выхода за границы массивов и др.
- Педагогический эксперимент и анализ:
 - Использование педагогических экспериментов для оценки эффективности среды при обучении.
 - Количественный анализ освоения навыков программирования с помощью «КуМир».

2.5.2 Критерии оценки качества

- Корректность исполнения программ:
 - Все корректно написанные алгоритмы должны выполняться без ошибок и с ожидаемым результатом.
- Удобство и информативность интерфейса:
 - Пользователь должен легко находить необходимые функции и видеть четкую обратную связь по ошибкам и состоянию выполнения.
- Надежность и стабильность:
 - Отсутствие сбоев и зависаний при работе с учебными проектами.
 - Корректная работа на рекомендованных конфигурациях.
- Точность диагностики ошибок:
 - Четкие и своевременные сообщения о синтаксических и логических ошибках в коде.

- Помощь пользователю в выявлении и исправлении ошибок.
- Поддержка учебных задач и тестов:
 - Возможность полноценного использования встроенных средств контроля выполнения и автоматического тестирования.
 - Соответствие требованиям образовательных стандартов и экзаменов.
- Производительность:
 - Быстрое выполнение программ и реакция среды на действия пользователя.

2.6 Заключительные положения

2.6.1 Срок действия технического задания

Действительно до момента полной разработки и вывода продукта на рынок, далее подлежит постоянному обновлению.

2.6.2 Ответственность сторон

Разработчик несет ответственность за соответствие требованиям, заказчик обязан предоставить необходимое оборудование и инфраструктуру для тестирования.