

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет компьютерных наук
Образовательная программа «Программная инженерия»**

СОГЛАСОВАНО

Научный руководитель, старший
преподаватель департамента больших
данных и информационного поиска

_____ В. В. Куренков

«__» _____ 2025 г.

УТВЕРЖДЕНО

Академический руководитель
образовательной программы
«Программная инженерия», старший
преподаватель департамента
программной инженерии

_____ Н. А. Павлочев

«__» _____ 2025 г.

**СИСТЕМА ПОСТРОЕНИЕ ГЕОМЕТРИЧЕСКИХ ЧЕРТЕЖЕЙ СО
ВСТРОЕННЫМ ЯЗЫКОМ ПРОГРАММИРОВАНИЯ И ВОЗМОЖНОСТЬЮ
УДАЛЕННОГО ПРОГРАММНОГО УПРАВЛЕНИЯ**

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.12.17-01 34 01-1-ЛУ

Исполнитель:

Студент группы БПИ233

_____ / С. А. Чубий /

«__» _____ 2025 г.

Подп. и дата	
Инв.№ дубл.	
Взам. инв.№	
Подп. и дата	
Инв.№ подл.	

УТВЕРЖДЕН

RU.17701729.12.17-01 34 01-1-ЛУ

**СИСТЕМА ПОСТРОЕНИЕ ГЕОМЕТРИЧЕСКИХ ЧЕРТЕЖЕЙ СО
ВСТРОЕННЫМ ЯЗЫКОМ ПРОГРАММИРОВАНИЯ И ВОЗМОЖНОСТЬЮ
УДАЛЕННОГО ПРОГРАММНОГО УПРАВЛЕНИЯ**

Руководство оператора

RU.17701729.12.17-01 34 01-1

Листов 10

Инов.№ подп	Подп. и дата	Взам. инв.№	Инов.№ дубл.	Подп. и дата

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММЫ	3
1.1. Функциональное назначение	3
1.2. Эксплуатационное назначение	3
2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ	4
3. ВЫПОЛНЕНИЕ ПРОГРАММЫ	5
3.1. Клиент командной строки (CLI)	5
3.2. Графический клиент (GUI)	7
3.3. Клиент-библиотека (lib)	9
3.4. Сервер	10
4. СООБЩЕНИЯ ОПЕРАТОРУ	11

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1. Функциональное назначение

Программный продукт позволяет пользователю строить и автоматически перестраивать геометрические чертежи. Производить расчеты на основе построенного чертежа.

1.2. Эксплуатационное назначение

Продукт состоит из трех исполняемых файлов для ОС Linux и одной библиотеки для языка программирования Rust:

- Сервера;
- Графического (GUI) клиента;
- Клиента командной строки (CLI);
- Клиента-библиотеки (lib).

Целевой аудиторией являются:

- Школьники, изучающие геометрию (5–11 классы);
- Школьные учителя, преподающие геометрию (5–11 класс);
- Студенты ВУЗов, изучающие вычислительную геометрию;
- Преподаватели ВУЗов, преподающие вычислительную геометрию.

Продукт будет полезен как для проведения занятий, так и для индивидуальной работы.

CLI- и lib-клиенты в первую очередь нацелены на студентов и преподавателей ВУЗов. GUI клиент будет интересен всем представителям целевой аудитории.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Для максимально качественной работы системы установлены следующие требования:

- Операционная система Linux¹
- 4Гб оперативной памяти
- 128Гб памяти на HDD или SSD
- Мышь, клавиатура, монитор

¹Разработка и тестирование проводились на NixOS Unstable (rev: 42a1c96).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1. Клиент командной строки (CLI)

Запустить установленный клиент командной строки можно, введя соответствующую команду в терминале. CLI-клиент может работать в нескольких режимах, которые описаны в последующих секциях. Более подробную информацию о аргументах командной строки можно получить, запустив клиент с флагом `-h`.

Сразу же после запуска клиент попытается подключиться к серверу, если подключение не удастся и адресом сервера будет является `localhost`, то клиент попытается запустить сервер и повторит попытку подключения. В остальных случаях, клиент завершится с сообщением об ошибке. Это поведение можно изменить, передав соответствующие аргументы командной строки.

3.1.1. Скриптовый режим

Скриптовый режим будет запущен, если передать имя файла, в качестве аргумента командной строки. Тогда этот файл будет обработан, как скрипт на встроенном языке программирования. Результат выполнения скрипта будет напечатан на стандартный вывод.

3.1.2. Режим стандартного ввода

Режим стандартного ввода будет запущен, если передавать данные на стандартный ввод через `pipe`. Тогда входные данные будут обработаны, как код на встроенном языке программирования. Результат выполнения скрипта будет напечатан на стандартный вывод.

3.1.3. Интерактивный режим

Интерактивный режим будет запущен в остальных случаях (то есть, если не передавать аргументов командной строки и не использовать `pipe`). В этом режиме пользователь может интерактивно вводить код на встроенном языке программирования и сразу получать результат его выполнения.

Для ввода многострочного кода нужно заключить его в `;` и `;`. Для завершения работы с клиентом, нужно сообщить ему о конце ввода (End of File (EOF)), для этого нужно нажать сочетание клавиш `Ctrl-D`.

Таблица 1 показывает пример интерактивной сессии.

Пользователь (ввод)	Система (вывод)	Комментарий
p1 = pt 100.0 200.0		Объявляем точку p1.
p2 = pt 400.0 600.0		Объявляем точку p2.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Пользователь (ввод)	Система (вывод)	Комментарий			
p2 = pt 400.0 600.0		Пытаемся переопределить точку p2.			
	error: failed to execute script Caused by: 0: define failed 1: got error from server 2: variable redefinition: p2	Получаем ошибку, так как переопределение запрещено.			
<pre>;; dist lhs:pt rhs:pt -> real = let delta = lhs - rhs, x = delta.x, y = delta.y, in (x^2 + y^2)^0.5 ;;</pre>		Объявляем новую функцию dist, которая измеряет расстояние между двумя точками. Используем ;; для ввода многострочного кода.			
d = dist p1 p2		Объявляем новую переменную d, равную расстоянию между точками p1 и p2.			
get! d		Получаем значение d.			
	<table><tr><td>Name</td><td>Value</td></tr><tr><td>d</td><td>500.000</td></tr></table>		Name	Value	d
Name	Value				
d	500.000				
set! p2 (pt 600.0 1400.0)		Обновляем значение p2.			
get! d					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Пользователь (ввод)	Система (вывод)	Комментарий							
	<table><thead><tr><th>Name</th><th>Value</th></tr></thead><tbody><tr><td>d</td><td>1300.000</td></tr></tbody></table>	Name	Value	d	1300.000	Получаем значение d. Видим, что оно изменилось вслед за p2.			
Name	Value								
d	1300.000								
get_all!		Выводим значения всех переменных.							
	<table><thead><tr><th>Name</th><th>Value</th></tr></thead><tbody><tr><td>d</td><td>1300.000</td></tr><tr><td>p2</td><td>pt 600 1400</td></tr><tr><td>p1</td><td>pt 100 200</td></tr></tbody></table>		Name	Value	d	1300.000	p2	pt 600 1400	p1
Name	Value								
d	1300.000								
p2	pt 600 1400								
p1	pt 100 200								
Нажато сочетание клавиш Ctrl-D.		Завершаем интерактивную сессию.							

Таблица 1. Пример интерактивной сессии

3.2. Графический клиент (GUI)

Запустить установленный графический клиент можно средствами вашей операционной системы. Например, можно ввести соответствующую команду в терминале.

3.2.1. Подключение к серверу

Сразу после запуска, клиент предложит пользователю подключиться к серверу. Если подключение не удастся и адресом сервера будет является localhost, то клиент попытается запустить сервер и повторит попытку подключения. В остальных случаях, клиент сообщит об ошибке и предложит пользователю повторить попытку подключения.

3.2.2. Работа с командной строкой

В GUI клиент встроена командная строка. Работа с ней аналогична работе с CLI клиентом.

3.2.3. Создание новых свободных² объектов

- Пользователь выбирает инструмент создания новых объектов в боковой панели.
- Пользователь выбирает тип создаваемого объекта.
- Пользователь вводит имя создаваемого объекта.

²Термины «свободный объект» и «зависимый объект» описаны в Техническом Задании.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- Пользователь вводит значение создаваемого объекта.
- Система создает новый объект, если это возможно. Иначе сообщает пользователю об ошибке.

3.2.4. Создание новых зависимых объектов

- Пользователь выбирает, на основании какой функции создать новый зависимый объект, в боковой панели.
- Пользователь выбирает, какие существующие объекты передать в функцию в качестве аргументов, кликая по ним мышью.
- Система создает новый объект, если это возможно. Иначе сообщает пользователю об ошибке.

3.2.5. Изменение свободных объектов

- Пользователь выбирает инструмент изменения свободных объектов в боковой панели.
- Пользователь выбирает свободный объект, кликая по нему мышью.
- Пользователь вводит новое значение объекта.
- Система обновляет значение объекта и пересчитывает дерево чертежа.

3.2.6. Получение значений объектов

- Пользователь смотрит на боковую панель. На этой панели выведена таблица с именами объектов и их значениями.

Name	Value
l	line (pt 100.0 100.0) (pt 200.0 200.0)
p1	pt 100.0 100.0
p2	pt 200.0 200.0

Таблица 2. Пример таблицы значений объектов

3.2.7. Удаление объектов

- Пользователь выбирает инструмент удаление объектов в боковой панели.
- Пользователь выбирает объект, кликая по нему мышью.
- Система удаляет выбранный объект и все объекты, зависящие от него (рекурсивно).

3.2.8. Вычисление произвольного выражения

- Пользователь вводит выражение.
- Пользователь нажимает кнопку вычисления выражения.
- Система вычисляет значение выражения и выводит результат на экран, если это возможно. Иначе, сообщает пользователю об ошибке.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.2.9. Сохранение чертежа в файл

- Пользователь нажимает кнопку сохранения в файл.
- Пользователь вводит имя файла.
- Система сохраняет чертеж в файл, если это возможно. Иначе, сообщает пользователю об ошибке.

3.2.10. Загрузка чертежа из файла

- Пользователь нажимает кнопку загрузки из файла.
- Пользователь вводит имя файла.
- Система загружает чертеж из файла, если это возможно. Иначе, сообщает пользователю об ошибке.

3.2.11. Изменение трансформации чертежа

- Пользователь выбирает инструмент изменения трансформации в боковой панели.
- Для изменения сдвига: удерживая левую кнопку мыши в нажатом положении, пользователь перемещает курсор мыши.
- Для изменения приближения: пользователь использует колесико мыши.
- Система меняет параметры трансформации и перерисовывает чертеж, учитывая новые параметры.

3.3. Клиент-библиотека (lib)

Для работы с библиотекой, нужно подключить её к вашему проекту на ЯП Rust любым доступным способом. Например, можно воспользоваться пакетным менеджером Cargo и командой `cargo add` (подробнее смотри «The Cargo Book», гл. 4.3.1 `cargo add`³).

Основной структурой клиента-библиотеки является `Client`. В первую очередь его нужно создать, используя либо метод `new`, либо метод `from`. Процесс подключения к серверу такой же, как и у CLI-клиента. Параметры подключения можно указать в структуре `ClientSettings` в случае создания `Client` через метод `from`.

Для выполнения кода на встроенном языке программирования использовать методы `exec` и `exec_one`. Для выполнения произвольных команды использовать метод `command`. Для выполнения остальных операций использовать методы с соответствующим названием: например, `set` для установки значения, `eval` для вычисления значения произвольного выражения и т.д.

³<https://doc.rust-lang.org/cargo/commands/cargo-add.html>

⁴<https://doc.rust-lang.org/rustdoc/>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Более подробную документацию сгенерировать на основе исходного кода стандартным для ЯП Rust способом⁴. Например, это можно сделать, выполнив команду `cargo doc --no-deps --open`, находясь в директории и исходным кодом клиента-библиотеки (`crates/client`).

3.4. Сервер

Запустить сервер можно либо вручную (это можно сделать средствами вашей операционной системы; например, можно ввести соответствующую команду в терминале), либо автоматически, при запуске одного из клиентов.

При ручном запуске серверу можно передать аргументы командной строки, информацию о которых можно получить, запустив сервер с аргументом `-h`.

Для успешного автоматического запуска убедитесь, что путь к серверу содержится в переменной окружения `$PATH`.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4. СООБЩЕНИЯ ОПЕРАТОРУ

Формат сообщений:

- При невозможности инициализации (запуска) CLI-клиент и сервер завершатся и выведут сообщение об ошибке на стандартный вывод. В следующих пунктах описываются только ошибки времени исполнения (**не** времени инициализации).
- При работе пользователя в CLI-клиенте, сообщения об ошибках выводятся на стандартный вывод.
- При работе пользователя в командной строке GUI-клиента, сообщения об ошибках выводятся в командную строку GUI-клиента. При работе пользователя с остальными частями GUI-клиента, сообщения об ошибках выводятся графически.
- При обработке REST API запросов сервером, сообщения об ошибках передаются в ответах на эти запросы. Такой ответ будет иметь код (HTTP Status Code) равный 500 (INTERNAL SERVER ERROR) и содержать сообщение об ошибке.

В сообщениях об ошибках подробно описаны причины их возникновения, из которых очевидным образом следуют способы их разрешения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.12.17-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата