МИНИСТЕРСТВО ОБРАЗОВАНИЯ НОВГОРОДСКОЙ ОБЛАСТИ

Областное государственное автономное профессиональное образовательное учреждение «Боровичский техникум строительной индустрии и экономики»

Специальность: 09.02.07 «Информационные системы и программирование»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА ПО**

**КУРСОВОМУ ПРОЕКТУ**

**Разработка программы для построения графиков математических функций**

# КП 01 МДК.01.03. 02 2021 ПЗ

Разработал: студентка группы П-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.М. Бубнова

Руководитель курсового проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ю.А. Архипова

Боровичи

2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ НОВГОРОДСКОЙ ОБЛАСТИ

Областное государственное автономное профессиональное образовательное учреждение «Боровичский техникум строительной индустрии и экономики»

«Утверждаю»

Зам. директора по учебной работе (зав. отделением) М.Ю. Кутузова

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на курсовое проектирование**

**МДК 01.03 Разработка мобильных приложений**

**«Разработка программы для построения графиков математических функций “Just math Building”»**

Студентке группы П-31 Бубновой Ксении Михайловне

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Введение

1. Общая часть
   1. Постановка задачи
      1. Назначение разработки
      2. Функциональные требования к программе
   2. Обзор существующих решений
   3. Технические и программные средства
      1. Требования к составу и параметрам технических средств
      2. Требования к программной и информационной совместимости
2. Специальная часть
   1. Описание алгоритмов
   2. Технология разработки программного изделия
   3. Результаты работы программы и их оценка
3. Технологическая часть
   1. Технология нисходящего проектирования программных изделий
   2. Технология документирования программных изделий
4. Экология и безопасность
   1. Вредные и опасные факторы при работе с компьютером
   2. Оптимальное рабочее место программиста

Заключение

Приложение А. Текст программы

Приложение Б. Видовые экраны работы программы

Приложение В. Алгоритм решения задачи

Дата выдачи «19» мая 2021 г.

Срок окончания «16» июня 2021 г.

Преподаватель-руководитель курсового проектирования Ю.А. Архипова

Председатель цикловой комиссии С. В. Андреева

# Аннотация

Данный курсовой проект разрабатывался на платформe Android Studio. Для работы с данным проектом необходимо было изучить тонкости языка программирования и выбранной платформы для разработки.

Работа проводилась с реальными источниками и информацией для более реальной и удобной работы программы. Все прописанные в коде функции и операции являются необходимыми для корректной работы программы.

Данная работа состоит из четырёх глав и подпунктов в них. Каждая глава раскрывает работу над проектом и работу самой программы. Основной объём составляют главы, а их составляют таблицы, приложения, рисунки.

Всего в курсовой присутствует две таблицы. В это число входят таблицы вариантов использования. Данные в них помогают правильно распределить нужную информацию внутри программы.

Количество рисунков в данном курсовом проекте составляет пять. Они отражают видовые окна программы и показывают, как выглядит интерфейс. Присутствуют рисунки, которые показывают код программы, вызов определённых обработчиков событий, в данном случае печати, остальные рисунки демонстрируют алгоритм работы программы и диаграмму вариантов использования. Все рисунки находятся в приложении.

Объём работы составляет 44 страницы.

Содержание

[Аннотация 4](#_Toc74174748)

[Содержание 5](#_Toc74174749)

[Введение 3](#_Toc74174750)

[1. Общая часть 5](#_Toc74174751)

[2. Специальная часть 11](#_Toc74174752)

[3. Технологическая часть 14](#_Toc74174753)

[3.1 Технологии визуального проектирования приложений 14](#_Toc74174754)

[3.2Технология проектирования баз данных 16](#_Toc74174755)

[4. Экология и безопасность 19](#_Toc74174756)

[4.1Эргономическая безопасность дисплеев 19](#_Toc74174757)

[4.2Гигиеническая оценка рабочих мест по условиям труда 21](#_Toc74174758)

[Заключение 24](#_Toc74174759)

[Использованная литература 26](#_Toc74174760)

[Приложение А. Текст программы 27](#_Toc74174761)

[Приложение Б. Видовые экраны работы программы 36](#_Toc74174762)

[Приложение В. Алгоритм решения задачи 39](#_Toc74174763)

# Введение

Актуальность темы разработки состоит в том, что в современном мире не всегда предоставляется решить или построить график математической функции и даже проверить. разрабатываемая программа предназначена для решения уравнений, построения графиков и основным пользователем будут являются студент, школьник и преподаватель.

Использование графических калькуляторов облегчит работу с математическими выражениями и их проверкой. Такие программы должны быть в арсенале каждого педагога, учащегося.

Цель – разработать программу для построения графиков математических функций.

Задачи:

1. Изучить предметную область

2. Ознакомится с оформлением мобильных приложений

3. Ознакомится и углубиться в язык программирования Java

4. Освоить платформу Android Studio

Создание программы для построения графиков математических функций.

Объектом курсовой работы будет являться создание программы для построения графиков математических функций. Под объектом курсовой понимается создание программы и ознакомление с языком Java.

В данном курсовом проекте предметом рассмотрения и изучения будут являться математические функции, графики, уравнения и математические операции.

Методом работы с данной программой будет являться метод анализа. Для создания качественного программного продукта следует проанализировать предметную область и понять алгоритмы работы.

Гипотеза. Основной гипотезой курсовой работы является возможное снижение времени решения задач, более наглядное и простое объяснение способов и методов решения. Также, возможность быстрой проверки задания.

Практическая значимость. Данная программа предоставляет возможность модернизации и ускорения процесса обучения. Сократит возможность ошибки, недочётов или неправильного построения графика.

Разрабатываемая программа предназначена для использования в целях подсчёта, проверки и построения графиков математических функций. Работает по прописанным математическими операциями, такие как расчёт по простым алгебраическим выражениям, определение х и у, а также многие другие.

Программный продукт должен использоваться исключительно на смартфонах, в данном случае только с программным обеспечением Android, ведь разработка программы происходит на платформе ПО Android Studio, где возможно разрабатывать только под вышеуказанный тип обеспечения.

Работа приложения будет заключаться в вычислении простых примеров, функций и их построения графиков. Интерфейс интуитивно понятен и имеет приятное оформление, теплые глазу цвета и цветовую гамму в целом. При включении пользователь может видеть экран загрузки, где представлен логотип и название «Just math Building».

# Общая часть

* 1. **Постановка задачи**
     1. **Назначение разработки**

Требуется создать мобильное приложение для построения графиков математических функций, которое позволить упростить работу с математикой, подсчётами, построением и определением графиков, проверку решений и ответов в решённых выражениях.

«Just math Building»­ ­ – программный продукт разработанный на платформе Android Studio и языке программирования Java, предназначенный для упрощения работы с математическими функциями и их графиками. Курсовая работа с программой «Just math Building» предназначена для автоматизации вычислений, проверки и построения функций. Данный курсовой проект предназначен для создания программного продукта, который позволит более быстро строить и вычислять графики функций «Just math Building», которая предназначена для автоматизации различных работы школьников, студентов, также включая преподавателей.

На данный момент обучение математике и работа с вычислениями занимают много времени и сил ученика, а также иногда невозможность проверить на правильность полученные результаты или непонимание того или иного алгоритма построения. При построении графика функций необходимо чётко знать, что к чему относиться, внимательно расписывать весь ход решения и определение получившегося выражения, то есть ответа. Весь график функции строится по изначально выданному выражению, проверяется и строится по решению, но не все могут сразу определить вид графика или как его строить, данная программа поможет разобраться.

«Just math Building» решает все введенные данные, анализирует какой график требуется и строит его, тем самым помогая или делая работу за пользователя.

* + 1. **Функциональные требования к программе**

«Just math Building» – программа для математических вычислений. Это приложение предназначено для пользования в личных целях, такого как вычисления и построения функций.

«Just math Building» обеспечивает решение таких задач, как, решение простых операций с числами, вычислениями; вычислений функций; построение графиков; проверка на правильность вычислений пользователя с результатами программы.

Данный курсовой проект включает в себя создание программы «Just math Building», которая будет состоять из нескольких модулей:

Простой калькулятор;

График.

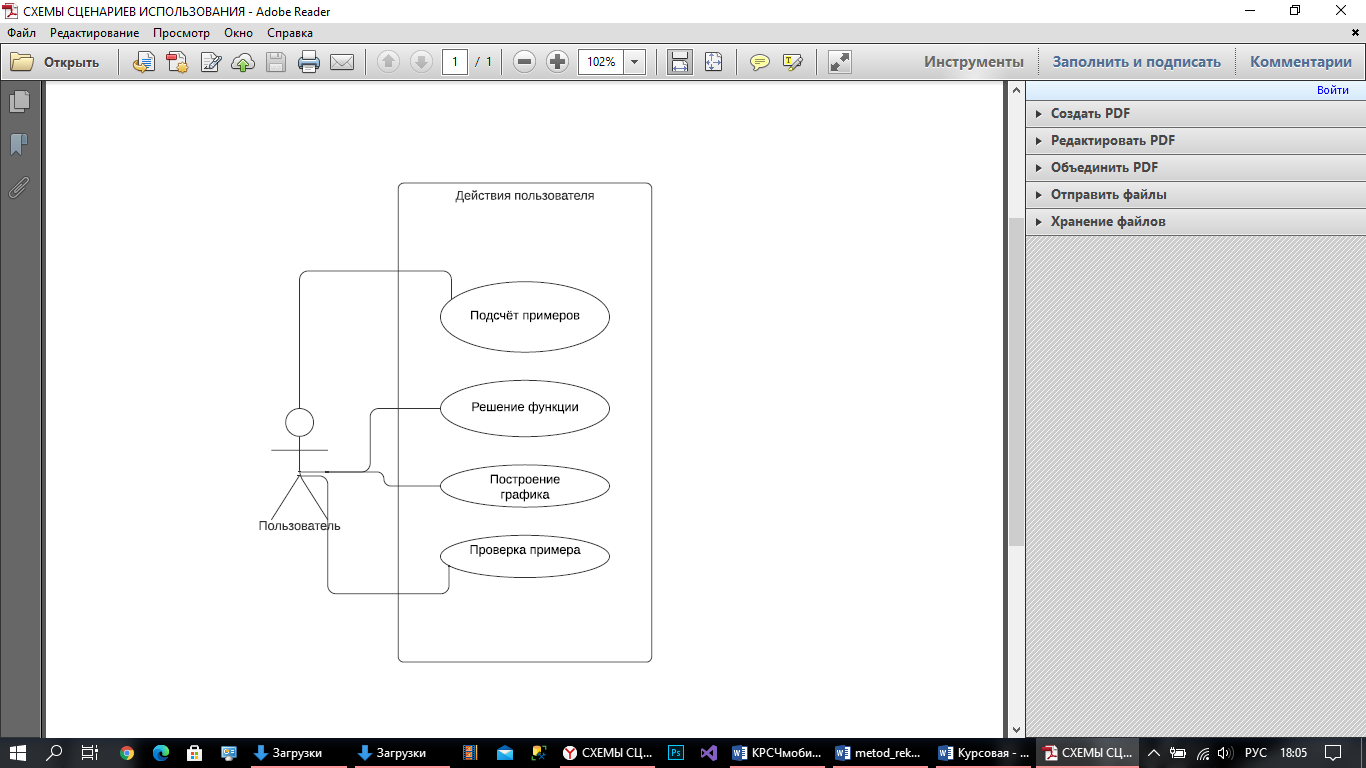
Функционал программы будет состоять в обычном калькуляторе и построении графиков. Пользователь будет вводить данные в программу, а она будет рассчитывать все необходимые и требуемые данные.

рис.1 Диаграмма вариантов использования

Таблица 1 – Варианты использования пользователем

|  |  |
| --- | --- |
| Раздел 1. Описание варианта использования | |
| Наименование варианта использования | Подсчёт |
| Актеры | Пользователь |
| Краткое описание | Пользователь вводит необходимые числа и проделывает операции над ними |
| Цель варианта использования | Вычисление входящих значений |
| Тип варианта использования | Базовый |
| Ссылки на другие варианты использования | Нет |
| Раздел 2. Последовательность действий актеров | |
| Действия актера | Отклик системы |
| 1. Пользователь вносит данные | 1.1 Записывание информации |
| 2. Получение ответа | 2.1 Демонстрация вычислений программой |
| Раздел 3. Последовательность действий актеров при возникновении исключительных ситуаций | |
| Действия актеров | Отклик системы |
| Исключение 1. Данные внесены некорректно | |
| Действие 1. Пользователь перепроверят данные и вносит заново | Отклик 1. Указать ошибки  Отклик 2. Исправление ошибок |

Таблица 2 – Варианты использования пользователем

|  |  |
| --- | --- |
| Раздел 1. Описание варианта использования | |
| Наименование варианта использования | Построение графика функции |
| Актеры | Пользователь |
| Краткое описание | Пользователь вводит необходимые числа и проделывает операции над ними |
| Цель варианта использования | Вычисление входящих значений |
| Тип варианта использования | Базовый |
| Ссылки на другие варианты использования | Вычисления |
| Раздел 2. Последовательность действий актеров | |
| Действия актера | Отклик системы |
| 1. Пользователь вносит данные | * 1. Записывание информации   2. Вычисление   3. Подсчёт |
| 2. Получение ответа | 2.1 Демонстрация вычислений программой  2.2 Постройка графика |
| Раздел 3. Последовательность действий актеров при возникновении исключительных ситуаций | |
| Действия актеров | Отклик системы |
| Исключение 1. Данные внесены некорректно | |
| Действие 1. Пользователь перепроверят данные и вносит заново | Отклик 1. Указать ошибки  Отклик 2. Исправление ошибок |

* 1. **Обзор существующих решений**

Калькулятор++ — это очень простое и приятное на вид приложение, но при этом набор функций в нём также поражает. Во-первых, при первом запуске пользователь сразу может настроить режим (обычный/инженерный), тему (светлая/тёмная/другие) и другие параметры отображения информации. Выглядит этот калькулятор очень просто и аккуратно, без нагромождений и лишних элементов. Во-вторых, помимо возможности производить всевозможные математические операции, копировать и сохранять результаты в памяти, просматривать историю вычислений, работать с переменными и константами, Калькулятор++ позволяет строить простые графики, создавать свои константы, переводить некоторые величины (время, длину, массу и т.д.). Кроме того, у этого калькулятора есть виджет, что даёт возможность пользователю производить быстрые расчёты прямо на рабочем столе, даже не запуская приложение.

Microsoft Math Solver — это математический помощник, основанный на искусственном интеллекте. Основной фишкой данного приложения является возможность считывания примеров из тетради или экрана монитора при помощи камеры смартфона, а также возможность рукописного ввода данных. Math Solver довольно быстро распознаёт различные задачи (связанные с арифметикой, алгеброй, тригонометрией, статистикой и некоторыми другими направлениями) и показывает решение с пошаговыми объяснениями. Задачу не обязательно фотографировать «в живую», можно импортировать изображение из галереи. Также Math Solver умеет строить графики. Ещё одной интересной особенностью этого калькулятора является то, что он предлагает пользователю для закрепления пройденного материала решить задачи, подобные ранее введённым. При этом приложение может работать и как стандартный калькулятор.

* 1. **Технические и программные средства**
     1. **Требования к составу и параметрам технических средств**

Для полноценной работы приложения необходим смартфон с операционной системой Android с версией не младше 9.

* + 1. **Требования к программной и информационной совместимости**

«Just math Building» работает только с платформой Android. Пользователь должен иметь смартфон Android любой марки и модели для работы с программой, также должен иметь базовые навыки работы с данным видом гаджетов.

Для контроля работы приложения, пользователь должен иметь хотя бы минимальную математическую базу, чтобы отслеживать работу приложения и свою, также анализировать свои ошибки.

# Специальная часть

* 1. **Описание алгоритмов**

Алгоритмы решения основных задач представлены в приложении В.

Пользователь может как решать простые математические операции (см. Приложение В рис1. Общий алгоритм решения задач), все необходимые решения задач пользователь может наблюдать в работе приложения. Программа имеет свой алгоритм расчёта, который написан по классическим математическим понятиям, поэтому любой может проверить приложение на корректность.

Построение графиков идёт по определённому алгоритму (см. Приложение В рис2. Алгоритм решения построения графиков), в котором всё идёт как в обычных математических расчётах на бумаге. Приложение само решает и определяет тип графика по введённым пользователем данным, далее выводит ответ на экран, где мы можем видеть сам график и проверить себя на правильность выполнения расчётов и знания математических функций.

* 1. **Организация данных и диалога**

Организация данных и диалог с пользователем осуществлён таким образом,

для того чтобы ввести данные пользователь должен вбить их при помощи специальной клавиатуры, но вводимые данные не хранятся в программе.

Диалог происходит посредством подсчёта данных, вывода ответа и вывода графика.

* 1. **Технология разработки программного изделия**

При разработке данного приложения использовались такие элементы как, экран калькулятора, графический экран, кнопки и текстовые поля все они отвечают за внос данных и вывод на экран. Для полноценной работы необходимо было создать несколько классов, где расписываются основные команды и функции программы.

Эмулятором выбран Pixel 4, на данном смартфоне отображаются все основные компоненты приложения. Данная программа не требует работы с сетью Интернет или с библиотеками, ведь все функции являются основными и их можно прописать, зная формулы и математические функции.

Технология разработки заключалась в изначальном создании макета приложения в AdobeXD, переносе графического интерфейса, составлении переходов между видовыми окнами и дальнейшего кодирования программы.

* 1. **Результаты работы программы и их оценка**

Работа программы начинается с её запуска и демонстрации экрана загрузки калькулятора (см. Приложение Б рис.1 Экран калькулятора). На нём показан логотип приложения и его название.

Далее перед пользователем открывается главное окно приложения, где пользователь может выбрать определённый раздел простым нажатием кнопки и сразу попасть на выбранный им пункт программы. Данное окно оформлено интуитивно просто, окрашен в приятные пользователю цвета.

При выборе обычного калькулятора перед пользователем открывается окно простого калькулятора (см. Приложение Б рис. 1 Экран калькулятора), где можно вводить цифры с специальной клавиатуры приложения. Все кнопки удобного размера, поэтому нажать не ту будет не так и легко, если совершается ошибка ввода, то можно очистить поле ввода специальной кнопкой.

На окне с обычным калькулятором можно найти кнопку для открытия графиков (см. Приложение Б рис.2 Экран графиков). Здесь можно вводить как цифры, так и буквы, выражения и т.д. (см. Приложение Б рис.3 Экран ввода выражений)

После внесения всех данных, пользователь нажимает на специальную кнопку для получения ответа или графика, если получение ответа, то далее приложение никуда не переходит.

После выбора получения графика, пользователю открывается новое окно (см. Приложение Б рис.4 Экран выбора выражения для построения) , где видно начерченный программой график математической функции.

Также можно выбрать сетку для более удобного рассмотрения графиков, введено всего два вида (см. Приложение Б рис.5 Экран видов сетки).

Результатами работы программы является успешное решение математических функций и построение графиков. Приложение хорошо справляется с поставленными перед ним задачами и является удобной для пользования, благодаря интуитивно понятному интерфейсу.

# Технологическая часть

# 3.1 Технологии визуального проектирования приложений

Визуальное программирование (от лат. visualis – зрительный) – это технология программирования, предусматривающая создание приложений с помощью наглядных средств.

К визуальному программированию можно отнести также Rapid Application Development (RAD) – быструю разработку программ. RAD – технология программирования, обеспечивающая ускоренную разработку и модификацию приложений за счет использования объектно-ориентированного и визуального программирования.

Средствами визуального программирования обычно решают задачи построения пользовательского интерфейса и упрощения разработки приложения путем замены метода "написания программы" на метод конструирования.

Визуальное программирование, бесспорно, обладает достоинством наглядного представления информации и гораздо лучше соответствует природе человеческого восприятия, чем методы традиционного, текстового программирования. Однако практически все визуальные средства нуждаются в дополнении функциями, которые не могут быть представлены в виде графических конструкций и требуют текстового выражения. Визуальные средства дополняются специальными программами – "скриптами", написанными на различных языках программирования.

Концепция визуального программирования реализована во многих современных средах разработки программных систем. Все ведущие фирмы, создающие средства для программирования и конструирования имеют системы, поддерживающие технологию визуального программирования.

Начальные шаги технологии визуального программирования определяются оболочкой самой среды визуального программирования. Сначала создаются экранные формы простейшей буксировкой мыши. В инспекторе объектов производится настройка их свойств путем заполнения отдельных полей. На главную форму помимо визуальных компонент наносятся не визуальные компоненты. Формы объединяются в единый проект. Далее в соответствии со сценарием диалога программируются методы события основной и подчиненных форм. Программы «пустых» методов событий появляются в окне редактора после нажатия соответствующих клавиш или действий мыши. «Пустые» методы дополняются определенными операторами активации и дезактивации форм. По окончании начальных шагов получается работающий «скелет» программы с источниками данных из файловых баз данных и со сгенерированными формами документов, выводимых на печать. Исследователь (Browser) обеспечивает визуализацию схемы иерархии классов полученного «скелета» программы. Другими словами, технический проект реализованной части программы формируется автоматически.

Дальнейшая разработка программы ведется по технологии объектно-ориентированного программирования. Можно часть программы реализовать по технологии структурного программирования. Некоторые недостающие визуальные и невизуальные компоненты получаются модификацией исходных текстов наиболее близких прототипов имеющихся компонент. Рекомендуется новые компоненты помещать в палитру компонент. Это облегчит их повторное использование в данной или последующих разработках. Код, относящийся только к данной разработке, набирается по тексту программы.

Выводы:

• Визуальное программирование во многом автоматизирует труд программиста по написанию программ.

• Визуальное программирование — одна из самых популярных парадигм программирования на данный момент. Оно базируется на технологии ООП.

• Среда визуального программирования поддерживает работу браузеров (Browser), при помощи которых можно автоматически получить документацию по структуре программы.

• Основным элементом в средствах визуального программирования является компонент. Компоненты бывают визуальными и не визуальными.

• Технология визуального программирования состоит в следующем: создание экранных форм, нанесение визуальных и не визуальных компонент, программирование событий и методов оконных форм.

# 3.2Технология проектирования баз данных

Список основных задач проектирования:

* Обеспечение хранения в БД всей необходимой информации.
* Обеспечение возможности получения данных по всем необходимым запросам.
* Сокращение избыточности и дублирования данных.

Обеспечение целостности базы данных.

Проектирование базы данных заключается в многоступенчатом описании будущей БД с различной степенью детализации и формализации, в ходе которого производится уточнение и оптимизация ее структуры.

Проектирование включает описание предметной области и задач информационной системы, далее идет к логическому описанию данных и затем – к физической модели БД. Различают три этапа детализации описания объектов БД и их взаимосвязей по трем основным уровням моделирования системы – концептуальному, логическому и физическому.

На концептуальном уровне проектирования производится смысловое (семантическое) описание информационного содержания предметной области, определяются границы предметной области, производится абстрагирование от несущественных для данной информационной системы деталей. В результате определяются моделируемые объекты, их свойства и связи. Выполняется структуризация знаний о предметной области, стандартизируется терминология. Затем строится концептуальная модель, описываемая на естественном языке. Для описания свойств и связей объектов применяют различные диаграммы.

На следующем шаге принимается решение о том, в какой СУБД будет реализована БД. Определяющими параметрами являются вид программного продукта и категория пользователей (профессиональные программисты или конечные пользователи, или и то, и другое). Другими показателями, влияющими на выбор СУБД, являются: удобство и простота использования; качество средств разработки, защиты и контроля БД; уровень коммуникационных средств (применение в сетях); фирма-разработчик; стоимость. Каждая конкретная СУБД работает с определенной моделью данных.

На логическом уровне производится отображение данных концептуальной модели в логическую модель, поддерживаемую выбранной СУБД. Здесь объектом работы выступают сами данные, их структура и правила построения. Логическая модель не зависит от конкретной СУБД – построенная на основе таблиц логическая модель может быть реализована на любой СУБД реляционного типа.

На физическом уровне производится выбор рациональной структуры хранения данных и методов доступа к ним, решаются вопросы эффективного выполнения запросов к БД, строятся дополнительные структуры, например, индексы. В физической модели содержится информация обо всех объектах БД (таблицах, индексах, процедурах и др.) и используемых типах данных. Физическая модель зависит от конкретной СУБД. Одной и той же логической модели может соответствовать несколько разных физических моделей. Физическое проектирование является начальным этапом реализации БД.

В данном курсовом проекте использовалось проектирование и построение баз данных на основе платформы Android Studio. Для данного проекта требовалось построить блок-схемы для понимания работы приложения и его расчётов, разработка диаграммы вариантов использования. Всё это необходимо для правильной разработки программы и её корректной работы.

# Экология и безопасность

# 4.1Эргономическая безопасность дисплеев

Эргономическая безопасность характеризуется такими условиям работы, когда ее удобно выполнять с максимальным эффектом. Это достигается правильным размещением рабочего стола и применением технических средств.

Общие эргономические требования и требования безопасности. Важным условием безопасности человека перед экраном является правильный выбор визуальных параметров дисплея и светотехнических условий рабочего места. В ГОСТ Р 50948-96 и ГОСТ Р 50949-96 и в Санитарных правилах и нормах (СанПиН) установлены требования к двум группам визуальных параметров:

1) Яркость, освещенность, угловой размер знака и угол наблюдения.

2) Неравномерность яркости, блики, мелькания, расстояние между

знаками, словами, строками, геометрические и нелинейные искажения, дрожание изображения и т.д.

Вторая группа требований обеспечения эргономической безопасности - нормы на излучения.

Визуальные эргономические параметры ПК являются параметрами безопасности, и их неправильный выбор приводит к ухудшению здоровья пользователей. Опасности, которые вас подстерегают, могут быть связаны с обострением хронических заболеваний глаз, проявлением наследственных предрасположенностей. Поэтому так важен режим работы с ПК, профилактические мероприятия, ну и, конечно, самое главное — эргономические параметры видеомониторов.

Одним из основных параметров является частота вертикальной или кадровой развертки (частота обновления), которая (что признается подавляющим числом международных и национальных стандартов) должна быть не менее 85 Гц, желательно в режиме максимального разрешения.

Особенно это важно при работе с графическими пакетами. В практической эргономике для определения усталостных характеристик человека-оператора применяется такой психофизиологический показатель, как критическая частота слияния мельканий (КЧСМ). Так вот КЧСМ зависит от яркости изображения, спектра излучения, местоположения изображения на сетчатке глаза, размеров наблюдаемого объекта, от возраста оператора и ряда других факторов, в том числе от времени работы человека с информационной моделью, вызывающей усталость. Но в любом случае она не может превысить 30-35 Гц в центральной области зрения. Этими цифрами, как вы уже догадались, определяется нижняя граница допустимой частоты вертикальной развертки монитора. Особенно высокой чувствительностью к изменению яркости изображения обладают окраинные области сетчатки ПК.

В настоящее время все мониторы должны соответствовать стандарту MPRII, ограничивающему излучение мониторов в диапазоне крайне низких частот (некоторые основные параметры, определенные этим и другими стандартами.

Требования эргономики и технической эстетики к рабочему месту программиста:

Для обеспечения требований эргономики и технической эстетики конструкция рабочего места, расположение и конструкция органов управления должны соответствовать анатомическим и психофизиологическим характеристикам человека. Вместе с этим все оборудование, приборы и инструменты не должны вызывать психологических раздражений.

Основными элементами рабочего места оператора являются: рабочий стол, рабочий стул (кресло), дисплей, клавиатура.

# 

# 4.2Гигиеническая оценка рабочих мест по условиям труда

Гигиеническая оценка условий труда заключается в определении фактических значений опасных и вредных производственных факторов на рабочих местах.

При аттестации рабочего места по условиям труда оценке подлежат все имеющиеся на рабочем месте опасные и вредные производственные факторы (физические, химические, биологические), тяжесть и напряженность труда.

Прежде всего рабочие места проверяются по гигиеническим критериям. Они показывают, насколько условия труда отличаются от действующих гигиенических нормативов.

При проверке гигиенических условий следует опираться на:

- Руководство по гигиенической оценке факторов рабочей среды и трудового процесса № Р 2.2.2006-05;

- Межотраслевые правила по охране труда в общественном питании ПОТ РМ-011-2000;

- Межотраслевые правила по охране труда в розничной торговле ПОТ РМ-014-2000.

При аттестации рабочих мест по условиям труда оценке подлежат все имеющиеся на рабочем месте вредные и (или) опасные производственные факторы.

Вредными производственным факторам могут быть:

- физические – такие как температура, влажность, скорость движения воздуха, тепловое излучение, производственный шум, освещение и т.п.;

- химические – химические вещества, смеси, получаемые химическим синтезом и (или) для контроля которых используют методы химического анализа;

- биологические факторы- микроорганизмы-продуценты, живые клетки и споры, содержащиеся в бактериальных препаратах, патогенные микроорганизмы - возбудители инфекционных заболеваний;

- тяжесть – преимущественно нагрузка на опорно–двигательный аппарат и функциональные системы организма (сердечно-сосудистую, дыхательную и др.), обеспечивающие его деятельность;

- напряженность;

- нагрузка преимущественно на центральную нервную систему, органы чувств, эмоциональную сферу работника, например, интеллектуальные, сенсорные, эмоциональные нагрузки, степень монотонности нагрузок, режим работы.

Опасные факторы рабочей среды

- факторы среды и трудового процесса, которые могут стать причиной острого заболевания или внезапного резкого ухудшения здоровья, смерти. В зависимости от количественной характеристики и продолжительности действия отдельные вредные факторы рабочей среды могут стать опасными.

Исходя из степени отклонения фактических уровней факторов рабочей среды и трудового процесса от гигиенических нормативов условия труда по степени вредности и опасности условно подразделяются на 4 класса: оптимальные, допустимые, вредные и опасные.

Оптимальные условия труда (1 класс) – условия, при которых сохраняется здоровье работника и создаются предпосылки для поддержания высокого уровня работоспособности. Оптимальные нормативы факторов рабочей среды установлены для микроклиматических параметров и факторов трудовой нагрузки. Для других факторов за оптимальные условно принимают такие условия труда, при которых вредные факторы отсутствуют либо не превышают уровни, принятые в качестве безопасных для населения.

Допустимые условия труда (2 класс) характеризуются такими уровнями факторов среды и трудового процесса, которые не превышают установленных гигиенических нормативов для рабочих мест, а возможные изменения функционального состояния организма восстанавливаются во время регламентированного отдыха или к началу следующей смены и не оказывают неблагоприятного действия в ближайшем и отдаленном периоде на состояние здоровья работников и их потомство. Допустимые условия труда условно относят к безопасным.

Вредные условия труда (3 класс) характеризуются наличием вредных факторов, уровни которых превышают гигиенические нормативы и оказывают неблагоприятное действие на организм работника и/или его потомство.

# Заключение

Основной целью данного курсового проекта является разработка программного обеспечения для построения графиков математических функций. При разработке необходимо было узнать предметную область работы приложения, основную аудиторию и понять главные задачи работы.

Сначала нужно было определить цель и функции курсового проекта и максимально их осуществить. В процессе разработки функции программы менялись несколько раз, пока не выяснились основные функции, а остальные нужно было удалить, так как они были бесполезны и трудноосуществимы.

Далее следовало составление таблиц и диаграмм. Разработанное приложение не имеет хранимых данных, большое количество пользователей с разными функциями и задачами, поэтому были составлены: диаграмма прецедентов и таблица вариантов использования.

На следующем этапе надо было решить алгоритмы решения задач, так как программа имеет работу с математикой, то надо было составить блок-схемы для решения простых вычислений и построения. Все данные в блок-схемах имеют общий вид для всех операций и функций, ведь они работают по одному алгоритму.

В данной курсовой работе рассмотрены другие программы, которые являются аналогами приложения. Разработанная программа является уникальной, из-за её доступности на Android и абсолютная бесплатность, своя цветовая гамма и свой набор функций, расположение клавиш и полей ответа. Другие решения являются программами крупных компаний - производителей ПО и они доступны на всех платформах.

Неотъемлемой частью данного курсового проекта является описание функционала и технических требований, последнее включает умение работы с описанными характеристиками. Функционал расписан достаточно полно и ясно. Функциональные требования к пользователю являются одними из важных, ведь они включают навыки работы с девайсами, а также аккаунтов, при помощи которых можно установить данное ПО.

В технологической части проекта описана и раскрыта тема визуального программирования, а точнее технология визуального проектирования приложений и технология проектирования баз данных. В данном разделе раскрыты данные темы и также связаны с практикой, так как это основа разработки и проектирования программы. Описывается теоретическая часть основ визуального проектирования и проектирования баз данных.

В разделе Экология и безопасность, разбирались такие темы как, эргономическая безопасность дисплеев и гигиеническая оценка условий труда. Данные пункты были написаны на основе общепринятых норм и САНПИНов.

Пункт «Результаты» включает в себя итоги работы всех функций и их оценку. Здесь подробно расписаны действия пользователя и его контакт с программой. Данный пункт связан с технологией разработки и организации данных и диалога с пользователем, ведь идёт оценка разработанного приложения и диалога программы с пользователем.

Курсовой проект позволил посмотреть на работу других схожих ПО и понять их работу, а после разработать своё. Цели и задачи, поставленные вначале выполнены на максимум и в дальнейшем имеет место для модернизации самой программы, возможным добавлением новых функций.

# Использованная литература

1. Интернет-учебник по программированию под Android [Электронный ресурс] Режим доступа: https:/ /metanit. com/ java/ android/.

2. Интернет-учебник по программированию под Android [Электронный ресурс] Режим доступа: http:// developer. alexanderklimov.ru/android/.

3. Документация по языку программирования Java [Электронный ресурс] Режим доступа: https://www.oracle.com.

4. Сайт вопросов и ответов для программистов [Электронный ресурс] Режим доступа: https://ru.stackoverflow.com

5.Установка Android Studio [Электронный ресурс] Режим доступа: https://startandroid.ru/ru/uroki/vse-uroki-spiskom/9-urok-2-ustanovka-i-nastrojka-sredy-razrabotki.html

6.Оценка условий труда по гигиеническим критериям [Электронный ресурс] Режим доступа: https://studfile.net/preview/6381613/page:50/

7.Требования безопасности к излучению от дисплея [Электронный ресурс] Режим доступа: https://studbooks.net/2429196/informatika/trebovaniya\_bezopasnosti\_izlucheniyu\_displeya

8.Визуальное программирование [Электронный ресурс] Режим доступа: https://studopedia.ru/3\_115492\_vizualnoe-programmirovanie.html

9.Технология визуального программирования [Электронный ресурс] Режим доступа: https://it.wikireading.ru/58399

10. Топ-5 лучших калькуляторов на телефоны Android: обычные и продвинутые варианты [Электронный ресурс] Режим доступа: https://trashbox.ru/link/top-android-calculators

11. Филлипс Б., Стюарт К., Марсикано К.Android. Программирование для профессионалов. 3-е издание / пер. с англ. — СПб.: Издательский дом «Питер», 2017 год, 688 стр.

# Приложение А. Текст программы

**Класс CalcListener**

public interface CalcListener {

void digitEvent(ActionEvent evt);

void operationEvent(ActionEvent evt);

void deletingEvent(ActionEvent evt);

void functionEvent(ActionEvent evt);

void bracketEvenet(ActionEvent evt);

}

public

class Main

extends JFrame {

public static void main(String[] args) {

SwingUtilities.invokeLater(

()->{

new Main();

}

);

}

**Класс Main**

public Main(){

JPanel jp0 = new JPanel();

jp0.setLayout(new BoxLayout(jp0,BoxLayout.Y\_AXIS));

jp0.add(Box.createRigidArea(new Dimension(0,3)));

JPanel jp = new JPanel();

this.setTitle("Just math Building");

jp.setPreferredSize(new Dimension(640,240));

MySystemChanger mySystemChanger = new MySystemChanger();

MyDisplay myDisplay = new MyDisplay();

MyKeybord myKeybord = new MyKeybord();

mySystemChanger.setMaximumSize(new Dimension(900,60));

myDisplay.setMaximumSize(new Dimension(900,200));

// myKeybord.setMinimumSize(new Dimension(640,290));

jp.add(mySystemChanger);

jp.add(myDisplay);

myDisplay.setPreferredSize(new Dimension(620,200));

myKeybord.addCalcListener(myDisplay);

mySystemChanger.addSystemListener(myDisplay);

mySystemChanger.addSystemListener(myKeybord);

jp0.add(mySystemChanger);

jp0.add(myDisplay);

jp0.add(myKeybord);

//jp.add(myDisplay, BorderLayout.PAGE\_START);

//jp.add(myKeybord, BorderLayout.CENTER);

JTabbedPane jtp = new JTabbedPane();

//jp0.add(mySystemChanger, BorderLayout.PAGE\_START);

//jp0.add(myDisplay, BorderLayout.CENTER);

//jp0.add(myKeybord, BorderLayout.PAGE\_END);

jtp.addTab("Калькулятор", jp0);

jtp.addTab("График", new GraphicTab());

this.add(jtp);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

//this.add(jp, BorderLayout.PAGE\_START);

//this.add(myKeybord, BorderLayout.CENTER);

//this.add(jtp);

//this.add(md1,BorderLayout.PAGE\_END);

//setSize( 640, 480);

pack();

setVisible(true);

}

}

**Класс MyCalculator**

public MyCalculator(String equation) {

this.equation=addBrackets(equation,2);

//ystem.out.println("To jest wynik: " +brackets(this.equation));

//System.out.println(solve(this.equation));

}

public double solveEquation() {

return brackets(this.equation);

}

public String addBrackets (String equation,int lvl) {

String eq1="";

String eq2="";

int brackL=0, brackR=0;

String operation="";

ArrayList<String> downOperations = new ArrayList<>();

for(int i=0; i<equation.length(); i++) {

if(((equation.charAt(i)=='\*'||equation.charAt(i)=='/')&&lvl==1)||((equation.charAt(i)=='^'||equation.charAt(i)=='p')&&lvl==2)) {

switch(equation.charAt(i)) {

case '\*':operation=operation+equation.charAt(i);

downOperations.add("+");

downOperations.add("-");

// downOperations.add("(");

// downOperations.add(")");

break;

case '/':operation=operation+equation.charAt(i);

downOperations.add("+");

downOperations.add("-");

// downOperations.add("(");

// downOperations.add(")");

//

break;

case '^':operation=operation+equation.charAt(i);

downOperations.add("+");

downOperations.add("-");

downOperations.add("\*");

downOperations.add("/");

break;

case 'p':operation=operation+equation.charAt(i);

downOperations.add("+");

downOperations.add("-");

downOperations.add("\*");

downOperations.add("/");

;

break;

}

int beginLeft= i;

int beginRight= i+1;

for(int j=beginLeft+1; j>=0 ; j--) {

boolean stat=false;

for(String x : downOperations) {

if(equation.charAt(j)==x.charAt(0) || j==0) {

eq1=equation.substring(0, j!=0 ? j+1 : j)+"("+equation.substring(j!=0 ? j+1 : j, beginLeft);

System.out.println("to jest 1: "+eq1);

brackL=j;

stat=true;

break;

}

}

if(stat)break;

}

for(int z=beginRight; z<equation.length() ; z++) {

boolean stat=false;

for(String x : downOperations) {

if(equation.charAt(z)==x.charAt(0) ||z==equation.length()-1) {

eq2=equation.substring(beginRight, z!=equation.length()-1 ? z : z+1)+")"+equation.substring(z!=equation.length()-1 ? z : z+1, z+1);

brackR=z+1;

stat=true;

System.out.println("to jest 2: "+eq2);

break;

}

}

if(stat)break;

}

equation=eq1+operation+eq2 +addBrackets(equation.substring(brackR,equation.length()),2);

break;

}

}

if(--lvl>0) {

equation =addBrackets(equation,lvl);

}

return equation;

}

public double brackets (String equation) {

System.out.println("this is "+ equation);

int rounds=0;

for(int i=0; i<equation.length(); i++)

if(equation.charAt(i)=='(')

rounds++;

int beg=0;

int end=0;

do

for(int i=0; i<equation.length(); i++) {

if(equation.charAt(i)=='(') {

beg=i ;

// rounds++;

}

if(equation.charAt(i)==')') {

end=i;

System.out.println(equation.substring(end+1,equation.length()));

//beg>0 ? "" : equation.substring(end+1,equation.length()-1);

equation = (beg==0 ? "" : equation.substring(0,beg)) +

Double.toString(solve(equation.substring(beg+1, end)))+

(end==0 ? "": equation.substring(end+1,equation.length()));

rounds--;

break;

}

}

while(rounds>0);

System.out.println(equation+" ------------------tu jest juz po nawiasach");

return solve(equation);

}

public double solve(String equation) {

System.out.println(equation + " co to tutaj mamy");

for(int i=equation.length()-1; i>=0; i--) {

if(equation.charAt(i)=='+') {

System.out.println(i);

return add(equation.substring(0, i),equation.substring(i+1, equation.length()));

}

if(equation.charAt(i)=='-') {

return substract(equation.substring(0, i),equation.substring(i+1, equation.length()));

}

if(equation.charAt(i)=='\*') {

return multiply(equation.substring(0, i),equation.substring(i+1, equation.length()));

}

if(equation.charAt(i)=='/') {

return divide(equation.substring(0, i),equation.substring(i+1, equation.length()));

}

}

System.out.println("------------ "+equation);

return Double.parseDouble(equation);

}

//Double.parseDouble(equation1)-solve(equation2)

public double add(String equation1, String equation2) {

return solve(equation1)+Double.parseDouble(equation2);

}

public double substract(String equation1, String equation2) {

return solve(equation1)-Double.parseDouble(equation2);

}

public double multiply(String equation1, String equation2) {

return solve(equation1)\*Double.parseDouble(equation2);

}

public double divide(String equation1, String equation2) {

return solve(equation1)/Double.parseDouble(equation2);

}

}

# Приложение Б. Видовые экраны работы программы

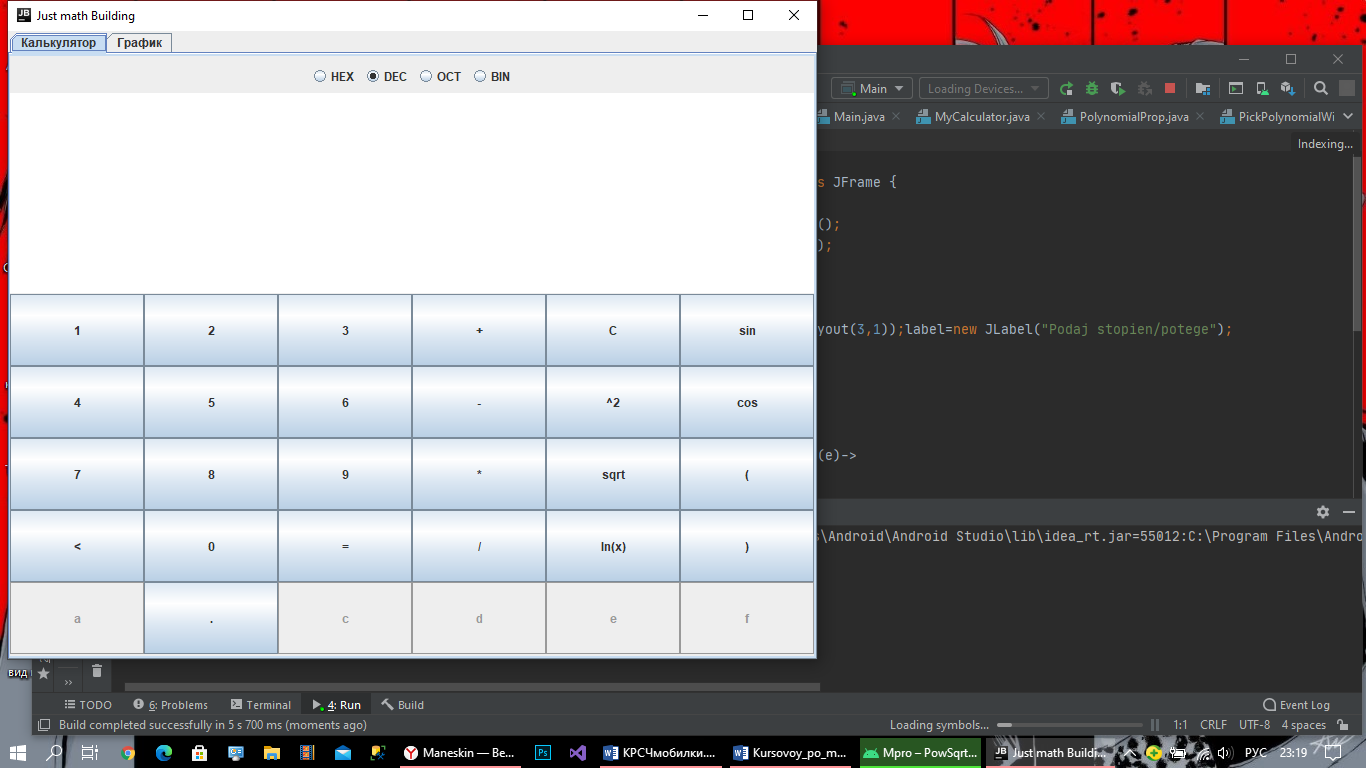


рис.1 Экран калькулятора

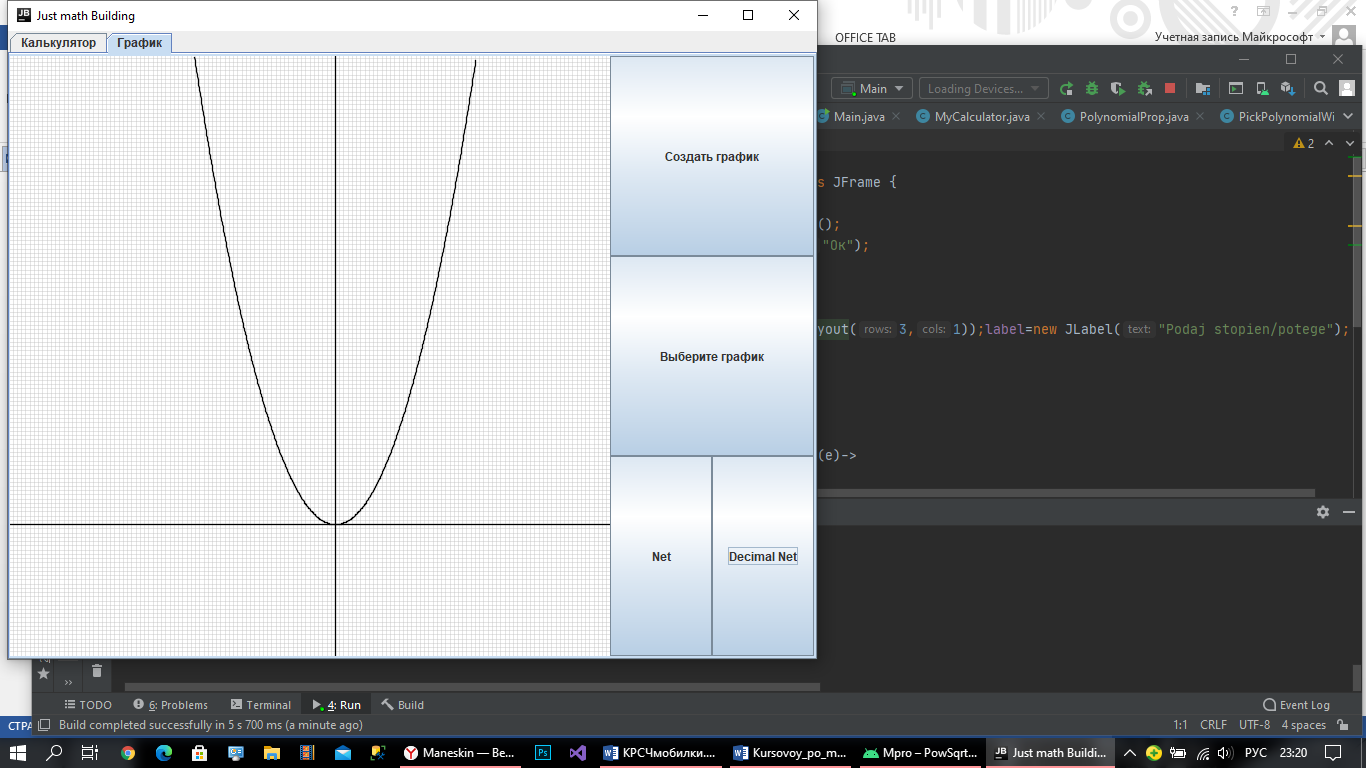


рис.2 Экран графиков

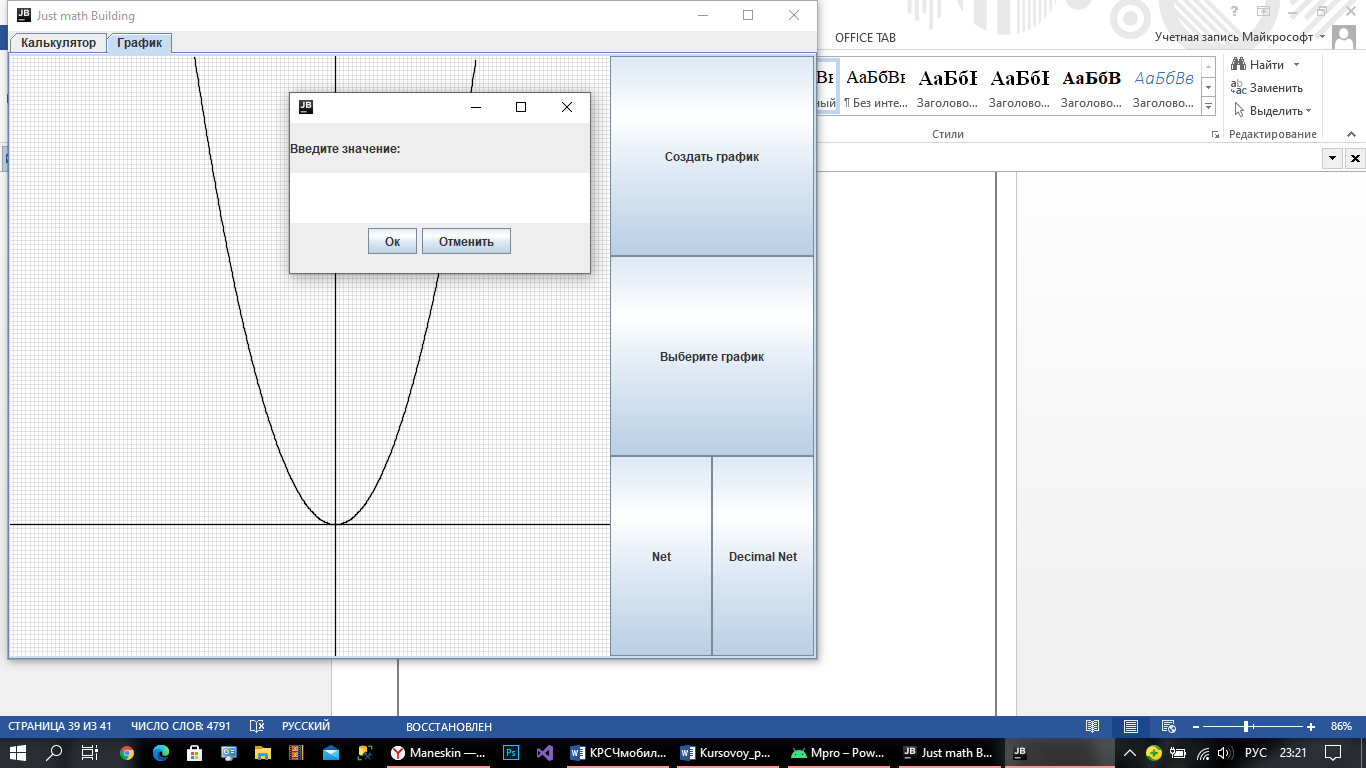


рис.3 Экран ввода выражений

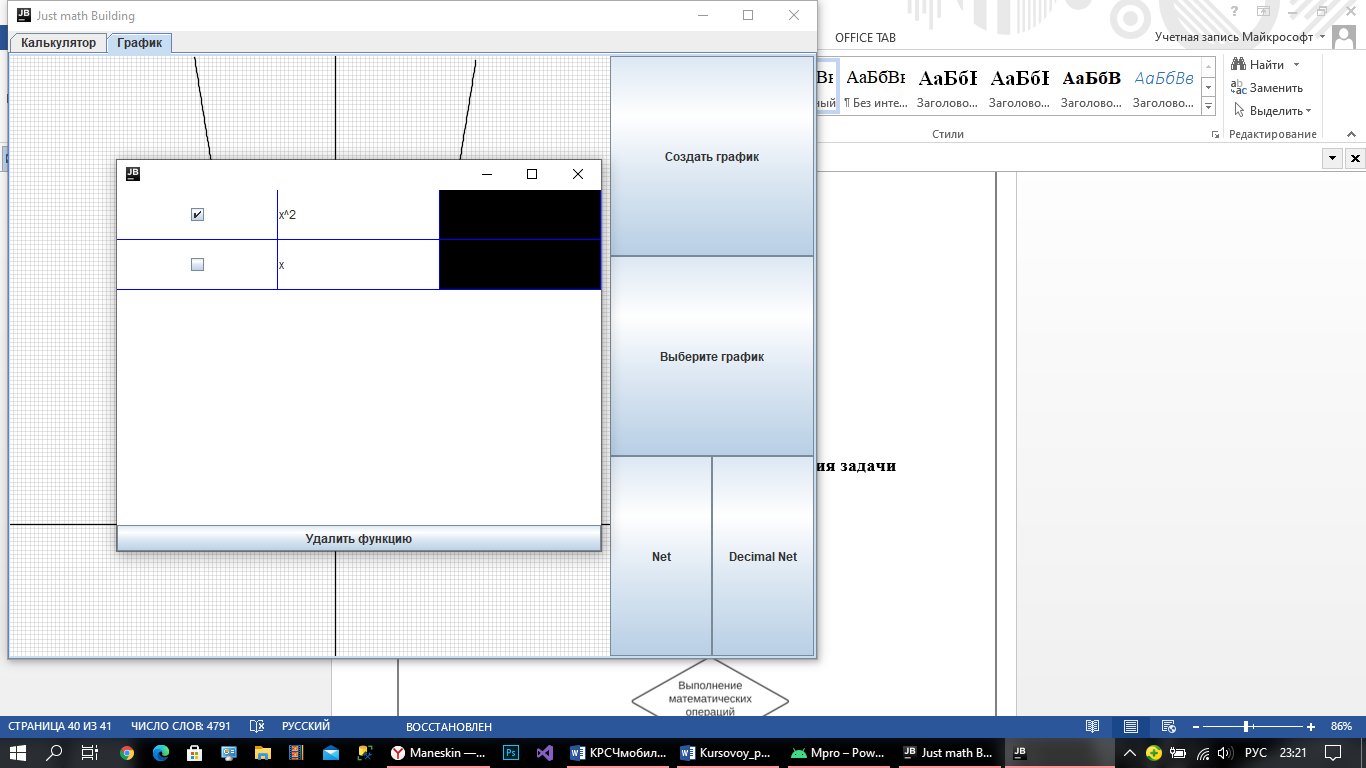


рис.4 Экран выбора выражения для построения

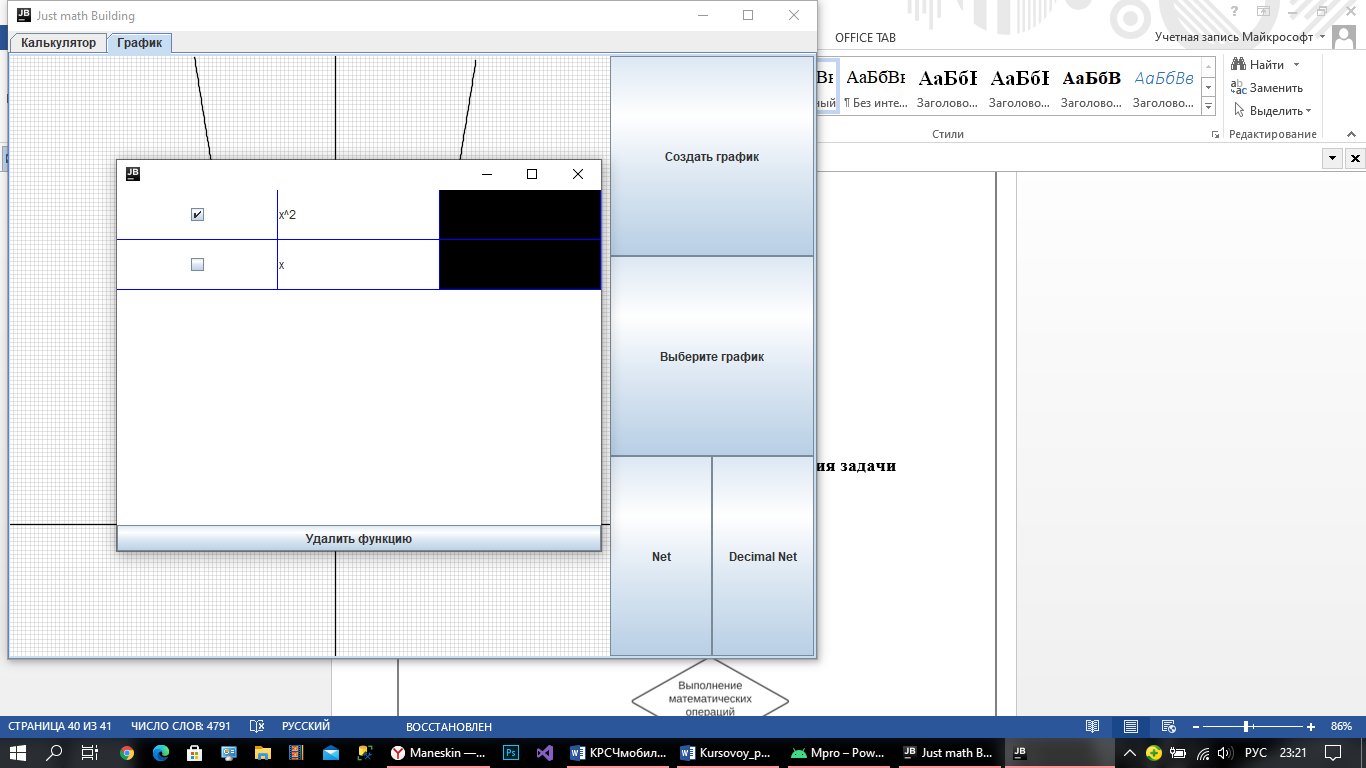


рис.5 Экран видов сетки

# Приложение В. Алгоритм решения задачи

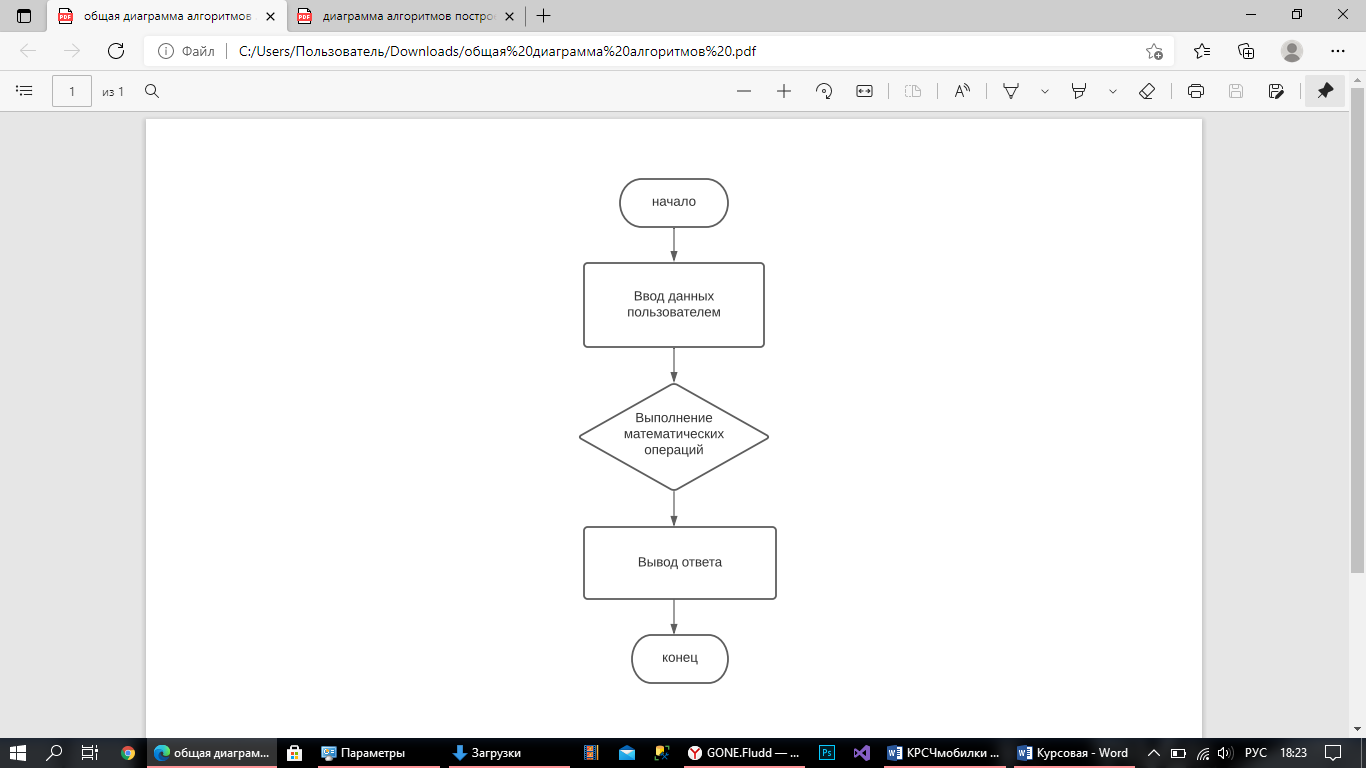


рис.1 Общий алгоритм решения задач

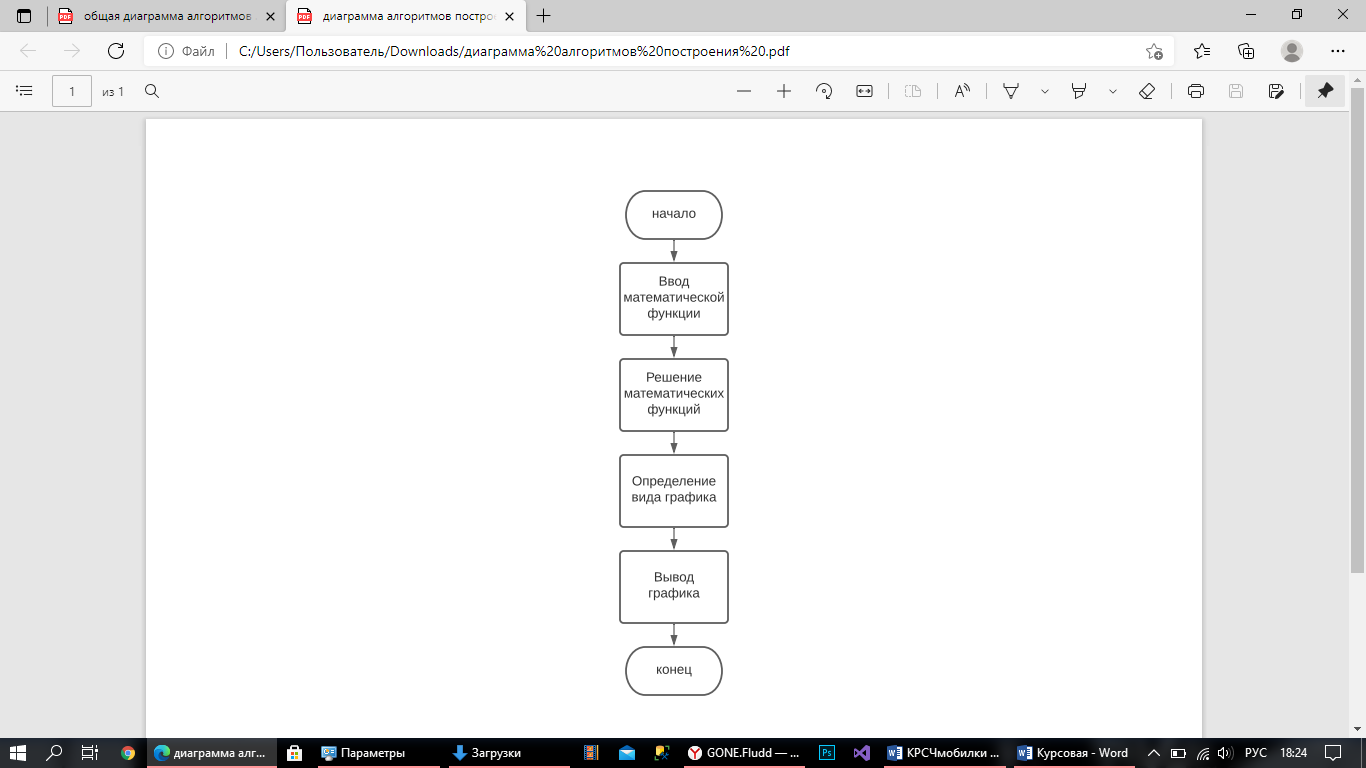


рис.2 Алгоритм решения построения графиков