

Міністерство освіти і науки України

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНИЙ  
УНІВЕРСИТЕТ

### **МЕТОДИЧНІ ВКАЗІВКИ**

по виконанню курсового проекту (курсової роботи)  
з дисципліни «Об'єктно-орієнтоване програмування»  
для студентів спеціальності «Інформаційні управляючі системи та технології на  
автомобільному транспорті»  
напряму підготовки 6.050101 «Комп'ютерні науки»  
галузі знань «Інформатика та обчислювальна техніка»

Харків 2015

Укладач: кандидат технічних наук,  
старший науковий співробітник,  
доцент кафедри Тімонін В.А.

Кафедра інформаційних технологій і мехатроніки

## **1. Цель и задачи курсового проекта (работы)**

Курсовой проект (работа) – это одна из форм учебно-исследовательской работы, выполнение которой представляет собой самостоятельное решение студентом под руководством преподавателя частной задачи или проведение исследования по одному из вопросов, изучаемых в дисциплине «Объектно-ориентированное программирование».

Курсовой проект (работа) является завершающим этапом изучения дисциплины.

Целью курсового проекта (работы) является совершенствование теоретических знаний и практических навыков, полученных студентами в процессе изучения дисциплины, в овладении современными технологиями разработки программного обеспечения для решения прикладных задач на ПК.

Задачами курсового проекта (работы) являются:

- обобщение и систематизация научных знаний, овладение современными методами поиска, обработки и использования информации;
- углубление уровня и расширение объема профессионально значимых знаний, умений и навыков;
- формирование умений и навыков самостоятельной организации научно-исследовательской работы;
- закрепление и развитие практических навыков по созданию программного обеспечения и соответствующей документации, поясняющей основные особенности работы созданного программного обеспечения.

## **2. Требования к выполнению курсового проекта (работы)**

Курсовой проект (работа) выполняется под руководством преподавателя, который проводит занятие со студентами по этой дисциплине.

Курсовой проект (работа) должен соответствовать следующим требованиям:

- быть выполненным на достаточном теоретическом уровне;
- включать анализ теоретического и экспериментального материала;
- основываться на результатах самостоятельного исследования;
- иметь обязательные самостоятельные выводы в заключение работы;
- иметь необходимый объем;
- быть оформленной по стандарту и выполненной в указанные сроки.

Для выполнения курсового проекта (работы) каждому студенту выдается индивидуальное задание, в котором:

- формулируется:
  - назначение, сущность и цель решения задачи;
  - требования к задаче, необходимые ограничения и допущения;
  - структура, содержание и формы представления исходных данных и результатов решения;
- возможность решения задачи по различным вариантам;

- форма представления курсового проекта (работы) и соответствующая документация;

- срок сдачи курсового проекта (работы);

- список рекомендованных источников и литературы.

Пример индивидуального задания курсового проекта (работы) приведен в приложении 1.

Курсовой проект (работа) оформляется в виде пояснительной записки с прилагаемым к ней на CD или DVD-диске разработанным программным обеспечением, работу которого необходимо продемонстрировать непосредственно на компьютере.

Пояснительная записка – это доклад студента в письменной форме о работе, которую он выполнил при выполнении курсового проекта (работы). Пример пояснительной записки курсового проекта (работы) приведен в приложении 2.

В пояснительной записке должно быть приведено руководство пользователя, которое оформляется в виде приложения.

Пояснительные записки (электронные копии курсовых работ) и разработанные базы данных на CD или DVD-дисках сохраняются на кафедре в течение 2-х лет.

Курсовые проекты (работы) студентов, что имеют теоретическую новизну и практическую ценность, рекомендуются кафедрой для включения в библиотечный электронный фонд кафедры или факультета.

### 3. Структура пояснительной записки

Пояснительная записка имеет следующую структуру:

- титульный лист;
- индивидуальное задание на курсовую работу;
- содержание;
- введение;
- основной текст;
- заключение;
- список литературы;
- приложения.

**Титульный лист** должен содержать следующую информацию:

- названия министерства, высшего учебного заведения, кафедры;
- название дисциплины;
- название темы учебной практики;
- данные о студенте (фамилия, имя, отчество, номер учебной группы);
- название специальности и направления подготовки;
- данные о научном руководителе (фамилия, имя, отчество, должность, ученая степень, ученое звание);
- дату сдачи и защиты, оценку с подписью и фамилией руководителя;

- название города, в котором находится учебное заведение;
- год написания отчета.

**Содержание.** После титульного листа следует содержание (лист не нумеруется). В нем содержится название разделов и подразделов с указанием страниц. Оно размещается на первой странице и печатается через 1,5 интервала.

**Введение.** Во введении (1-2 страницы) обосновывается актуальность темы исследования и её практическая значимость, определяется логическая последовательность изложения материала с учетом того, что раскрытие теоретических вопросов темы предшествует анализу практических проблем объекта исследования, по материалам которого пишется курсовая работа, определяется объект и предмет исследования, формулируются цели и задачи.

Оно включает в себя краткий обзор литературы и экспериментальных данных, оценку степени теоретической разработанности проблемы и анализ противоречий практики, обоснование темы исследования и необходимости ее дальнейшего научного изучения.

**Основной текст.** В основной части работы (общий объем 20-30 страниц) значительное место отводится современному аспекту рассматриваемой темы, в которой выявляются современные особенности осуществления анализируемых закономерностей, объясняются их причины, определяются тенденции дальнейшего развития.

Кроме того, желательно использовать и специфический материал, учитывающий специфические особенности предмета исследования или условий использования, по выполняемой теме. Анализ специфического материала не только существенно обогащает теоретические выводы, но и позволяет выявить особенности осуществления общих экономических закономерностей.

Основной текст содержит описание выполненной работы по разработке программного обеспечения и состоит из разделов и подразделов, в которых размещается как теоретический (постановка задачи и математическое описание задачи), так и практический (разработка программного обеспечения и документации, результаты решения поставленной задачи) материал. Каждый раздел должен заканчиваться короткими выводами, которые должны быть обоснованными и грамотно изложенными.

Описание работы состоит из следующих разделов и подразделов:

- постановка задачи и математическое описание задачи;
- алгоритм решения задачи;
- описание приложения, в которое входит:
  - спецификация приложения;
  - исходный код приложения;
  - описание интерфейса;
  - руководство пользователя (размещается в приложении);
- результаты решения задачи.

**Постановка и математическое описание задачи.** Постановка задачи – это словесная формулировка задачи, которая берется из индивидуального задания. Здесь указываются требования к задаче, необходимые ограничения и допущения, при которых решается задача.

Математическое описание задачи – это перечень математических выражений, формул, уравнений, констант, с помощью которых могут быть получены результаты решения задачи. Если математические соотношения приведены в индивидуальном задании, то они переписываются из него. Если в индивидуальном задании математических соотношений нет, то исполнитель должен найти такие соотношения в рекомендованной литературе.


**Алгоритм решения задачи.** В этом подразделе отчета приводятся схема работы приложения, определяющая последовательность действий и этапы решения задачи, блок-схемы методов приложения.

При разработке алгоритма следует использовать пошаговый способ построения алгоритма. На первом шаге разрабатывается общая схема решения задачи, то есть формулируется ответ на вопрос «Что сделать», чтобы поставленная задача была решена. На втором шаге переходят к детализации каждого блока общей схемы, то есть выясняют и решают «Как сделать» каждый блок общей схемы, чтобы задача была решена. Результатом пошаговой разработки будет детальный алгоритм (один блок - одно действие) решения задачи.

После этого разрабатываются схема работы приложения и блок-схемы методов приложения с использованием символов, определенных ГОСТом 19.701-90.

**Описание приложения.** В этом подразделе отчета размещается спецификация приложения, приводится исходный код приложения, описание интерфейса приложения и руководство пользователя.

При описании приложения приводится его спецификация, которая включает в себя перечень сведений об объектах и типах данных, используемых приложением. В нем должно быть осуществлено присвоение всем объектам задачи имен согласно правилам алгоритмического языка С#.

 Согласно определению, приведенному в Единой системе конструкторской документации, спецификация — основной конструкторский документ, определяющий состав сборочной единицы, комплекса, комплекта. В спецификации содержится подробное перечисление узлов и деталей какого-либо изделия, конструкции, установки и т. п., входящих в состав сборочного или рабочего чертежа. Согласно определению, приведённому в Политехническом словаре, спецификация — выполненный в форме таблицы документ, определяющий состав какого-либо изделия.

При описании разработанных классов необходимо указать назначение и состав класса и каждого элемента в классе, определить, какие данные целесообразно использовать в качестве переменных, а какие – в качестве

констант. Как правило, в качестве константы используются физические, математические, логические данные. После этого следует перейти к спецификации каждого метода, то есть описанию переменных метода. Пример спецификации метода приведен в таблице 1.

Таблица 1 – Спецификация метода Obrabotka

Переменная	Математическое обозначение	Имя в программе	Структура данных	Тип данных	Назначение переменной
$\alpha$	$\alpha$	Alfa	простая	double	Угол поворота
$\pi$	$\pi$	PI	простая	double	Константа
...					
i		i	простая	int	Параметр цикла
t	t	time	структура	int	Текущее время
...					
P	P	VerSyst	простая	double	Вероятность безотказной работы системы
R	R	masR	массив	double	Соппротивление в цепи

Исходный код разработанного приложения приводится в виде листингов файлов с расширением \*.cs.

Описание интерфейса, в котором содержится описание всех компонентов, используемых при разработке приложения, должно быть проиллюстрировано внешним видом формы приложения и поясняющими надписями.

Руководство пользователя, в котором содержится порядок работы с приложением, должно быть проиллюстрировано последовательностью скриншотов на каждом этапе выполнения приложения (скриншот – это мгновенный снимок экрана монитора (изображение), которое показывает в точности то, что отображено на мониторе в конкретный момент).

**Результаты решения.** В этом подразделе курсовой работы размещаются результаты решения задачи при различных начальных данных.

**Заключение.** В заключение (объем 2-3 страницы) в сжатой, лаконичной форме подводятся итоги проведенного исследования, обобщаются основные теоретические положения и практические результаты, делаются выводы и предложения.

**Список использованных источников и литературы.** Список использованных источников и литературы содержит весь перечень источников,

использованных студентом при подготовке и выполнении курсовой работы. Оформление списка осуществляется согласно государственному стандарту Украины ДСТУ 7.1:2006 "Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления".

**Приложения.** В приложениях размещается материал, дополняющий текст работы и носящий справочный или рекомендательный характер. Приложением может быть графический материал, таблицы большого формата, расчеты, описания алгоритмов и программ задач, решаемых на персональных компьютерах и т.д.

#### **4. Методические указания по выполнению курсового проекта (работы)**

Курсовой проект (работа) выполняется в соответствии с календарным планом:

- первый месяц семестра:
  - выбор темы курсовой работы и ее предварительное обсуждение с руководителем;
  - поиск, подбор, систематизация и анализ материалов по теме курсовой работы;
  - составление плана курсовой работы и обсуждение его с руководителем;
- второй месяц семестра:
  - формализация задачи, т.е. представление ее в виде математической модели или составление строгого формального описания процедуры ее решения, входных и выходных данных;
  - разработка алгоритма решения задачи на основе её математического описания.
- третий месяц семестра:
  - разработка программного обеспечения;
  - отладка и тестирование программного обеспечения на ПК;
  - разработка документации программного обеспечения;
  - написание чернового варианта курсовой работы;
- четвертый месяц семестра:
  - предварительное обсуждение результатов выполнения курсовой работы;
  - оформление пояснительной записке;
  - представление курсовой работы на отзыв научному руководителю и ее защита.

Разработка программного обеспечения осуществляется в несколько этапов:

- постановка задачи;
- формализация задачи;



- программирование;
- сдача в эксплуатацию.

На этапе постановки задачи осуществляется разработка общего описания задачи с соответствующими обоснованиями. При постановке задачи необходимо обращать внимание на четкость и полноту изложения, однозначность и отсутствие произвольного токования описываемого процесса, а также на анализ всех существующих методов решения задачи в целях выбора наиболее целесообразного.

При постановке задачи четко формулируются:

- назначение, сущность и цель решения задачи;
- требования к задаче, необходимые ограничения и допущения;
- основные критерии решения;
- структура, содержание и формы представления исходных данных и результатов решения;
- ожидаемая периодичность и допустимая продолжительность решения;
- возможность решения задачи по различным вариантам.

На этапе формализации осуществляется описание условия задачи с помощью математических зависимостей (уравнений, последовательности формул), которые учитывают заданное число факторов и обеспечивают решение поставленной задачи с необходимой точностью.

В результате формализации создается математическая модель предметной области, которая записывается средствами различных видов математических моделей, либо просто формируется строгое описание задачи (не всякая предметная область может быть описана средствами какой-либо математической модели), определяются входные и выходные данные для задачи (или комплекса задач).

На этапе программирования на основании математической модели или другого формального описания поставленной задачи разрабатывается программное обеспечение. Технология программирования задач разбивается на два подэтапа:

- разработка алгоритма;
- разработка и отладка программы.

При разработке алгоритма следует использовать пошаговый способ построения алгоритма. На первом шаге разрабатывается общая схема решения задачи, то есть формулируется ответ на вопрос «Что сделать», чтобы поставленная задача была решена. На втором шаге переходят к детализации каждого блока общей схемы, то есть выясняют и решают «Как сделать» каждый блок общей схемы, чтобы задача была решена. Результатом пошаговой разработки будет детальный алгоритм (один блок - одно действие) решения задачи.

После этого разрабатываются схема работы приложения и блок-схемы методов приложения с использованием символов, определенных ГОСТом 19.701-90.

На этапе сдачи программного обеспечения в эксплуатацию результаты разработки программного обеспечения передаются заказчику для определения того, удовлетворен ли он решением. В этом случае оформляется документация на эксплуатацию программного продукта.

## **5. Требования по оформлению курсового проекта (работы)**

Оформление курсового проекта (работы) должно соответствовать требованиям ГОСТа Украины ДСТУ 3008-95 «Документация. Отчеты в сфере науки и техники: структура и правила оформления».

Курсовой проект (работа) в виде пояснительной записки должен быть напечатан на стандартных листах бумаги в формате А4 как структурированный документ с соблюдением следующих требований:

- рукопись должна быть подготовлена в текстовом редакторе MS Word шрифтом Times New Roman, размер шрифта для основного текста – 14;
- поля: левое – 30, правое – 20, верхнее – 20, нижнее – 20 мм;
- межстрочный интервал – одинарный;
- отступ красной строки – 1,25;
- выравнивание текста – по левому краю.

Каждый структурный элемент содержания курсового проекта (работы) начинается с новой страницы. Наименование структурных элементов следует располагать по центру строки без точки в конце, без подчеркивания.

Иллюстрированный материал следует располагать в курсовой работе непосредственно после текста, в котором они упоминаются впервые. На все иллюстрации должны быть ссылки в работе. Иллюстрации (чертежи, графики, схемы, рисунки, снимки) должны быть пронумерованы и иметь названия под иллюстрацией. Номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой, например, Рисунок 3.2 – График функции  $y=\sin(x)$ .

Таблицы в курсовой работе располагаются непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. На все таблицы должны быть ссылки в тексте. Заголовок таблицы размещается над таблицей, точка в конце заголовка не ставится. Таблицы следует нумеровать арабскими цифрами порядковой нумерацией в пределах раздела, за исключением таблиц, приводимых в приложениях. Номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой, например, Таблица 2.1 – Перечень методов класса Form1.

Формулы приводятся сначала в буквенном выражении, затем дается расшифровка входящих в них индексов, величин, в той же последовательности, в которой они даны в формуле. Уравнения и формулы следует выделять из текста в отдельную строку. Уравнения и формулы нумеруются в круглых скобках справа от формулы. Нумерация уравнений и формул осуществляется по тому же принципу, что и иллюстрации.

Цитирование различных источников в курсовой работе оформляется ссылкой на данный источник указанием его порядкового номера в списке литературы в квадратных скобках после цитаты. Если используется ссылка на фрагмент первоисточника (статистические данные, высказывание автора и т.п.), необходимо указывать страницу, где именно этот фрагмент расположен, например, [3, с. 22-23].

При использовании аббревиатуры, условных географических сокращений, следует указывать их сразу же после полного наименования данного сложного термина. Например, «персональный компьютер» (ПК). После этого можно свободно оперировать своей аббревиатурой без расшифровки.

Приложение оформляется как продолжение отчета с новой страницы с указанием посередине строки слова «Приложение » и прописной буквы, обозначающая приложение, например, «Приложение А», «Приложение Б». Приложение должно иметь заголовок, напечатанный строчными буквами с первой прописной симметрично относительно текста страницы.

Все листы курсовой работы и приложений аккуратно подшиваются (брошюруются) в папку и переплетаются. Страницы отчета, включая приложения, нумеруются арабскими цифрами с соблюдением сквозной нумерации. Номер страницы проставляют в правом верхнем углу страницы без точки в конце.

Заканчивается пояснительная записка личной подписью исполнителя курсового проекта (работы):

Пояснительную записку выполнил  
студент Петренко А.О.

## **6. Порядок проведения защиты курсового проекта (работы)**

Аттестация студента за курсовой проект (работу) проводится в процессе ее защиты.

Курсовой проект (работа) должен быть сдан научному руководителю на проверку в срок, определенному в индивидуальном задании. Научный руководитель в 10-дневный срок должен проверить курсовой проект (работу), после чего назначается дата защиты курсовой работы.

Курсовая работа допускается к защите при условии законченного оформления (наличие пояснительной записки и её электронной версии на CD или DVD-диске), наличия разработанного проекта программного обеспечения

на CD или DVD-диске и допуска научного руководителя или наличия рецензии (письменного отзыва).

Защита курсовых работ должна быть проведена до начала экзаменационной сессии. Порядок защиты, определяется руководителем и доводится до студента не позже, чем за неделю до защиты.

Защита курсовых работ проводится в установленное время в виде публичного выступления студента: защиты перед комиссией кафедры с участием руководителя работы, выступления на научно-практической конференции и т.д. В отсутствие руководителя курсовой работы защита может быть проведена при условии представления им письменного отзыва на курсовую работу.

Для защиты курсовой работы студенту предоставляется 7-10 мин. для изложения основных положений работы.

В ходе защиты курсовой работы необходимо продемонстрировать работу приложения.

Студенты, которые в срок, отведенный на выполнение курсовой работы, не оформили пояснительную записку и не сдали ее руководителю на проверку, к защите курсовой работы не допускаются и получают оценку „неудовлетворительно”. До повторной защиты курсовой работы они допускаются после оформления пояснительной записки и ее проверки.

У студентов, что не защитили свои работы или не явились на защиту, повторная защита может проводиться в период экзаменационной (зачетной) недели.

Критериями оценки курсовой работы являются:

- актуальность и степень разработанности темы;
- творческий подход и самостоятельность в анализе, обобщениях и выводах;
- полнота охвата первоисточников и исследовательской литературы;
- уровень овладения методикой исследования;
- качество программного обеспечения и прилагаемой к нему документации;
- научная обоснованность и аргументированность обобщений, выводов и рекомендаций;
- научный стиль изложения;
- соблюдение всех требований к оформлению курсовой работы и сроков ее исполнения;
- аргументированность и точность ответов на вопросы.

### **Список использованных источников и литературы**

1. Державний стандарт України «Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний

опис. Загальні вимоги та правила складання». ДСТУ ГОСТ 7.1:2006. – К.: Держспоживстандарт України, 2007 – 20 с.

2. Державний стандарт України «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення». ДСТУ 3008-95. – К.: Держстандарт України, 1995 – 37 с.

3. Наказ Міністерства освіти і науки, молоді та спорту України 29 березня 2012 року № 384 «Про затвердження форм документів з підготовки кадрів у вищих навчальних закладах I-IV рівнів акредитації». – К.: Міністерство освіти і науки, молоді та спорту України, 2012.

4. <http://www.crimea.ua/liba/docs/metod.pdf>

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ  
УНИВЕРСИТЕТ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МЕХАТРОНИКИ

«Утверждаю»  
Заведующий кафедрой  
профессор, доктор технических наук  
Никонов О.Я.  
« » сентября 2015 г.

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**  
по дисциплине «Объектно-ориентированное программирование»  
на тему «Разработка приложения для решения трансцендентных уравнений  
графическим способом»

студенту 1-го курса учебной группы РИ11  
направления подготовки «Компьютерные науки»  
специальности «Информационные управляющие  
системы и технологии на автомобильном транспорте»  
Петренко А.О.

Руководитель:  
доцент кафедры  
старший научный сотрудник  
кандидат технических наук  
Тимонин В.А.

Харьков 2015

### Задание

Разработать приложения, с помощью которого реализуется графический способ решения трансцендентных уравнений, и провести оценку точности решения трансцендентных уравнений.

В качестве входных данных использовать уравнение

$$\frac{(x+2)^2(x-1)^3}{2} - 4\sin(x) = 0,$$

а поиск корней уравнения осуществлять для значений  $x \in [-2.5, 2.1]$ .

Разработанное приложение «Решение уравнения» должно:

- реализовывать все способы решения;
- отображать на экране монитора персонального компьютера:
  - графики функций;
  - значения корней (при щелчке левой кнопкой мыши на соответствующей точке графика).

Результаты решения должны сохраняться в текстовом файле.

Работу представить в виде пояснительной записки, её электронной копии и приложения на CD или DVD-диске.

Срок сдачи курсового проекта (работы) – «    » декабря 2015 г.

### Список рекомендованных источников и литературы

1. Выгодский М.Я. Справочник по высшей математике. – М.: АСТ: Астрель, 2010 – 704 с.: ил.
2. Голощапов А.Л. Microsoft Visual Studio 2010. – СПб.: БХВ-Петербург, 2011. – 544 с.: ил.
3. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 1. Пер. с англ. - М.: «Русская Редакция», 2002.- 576 с.: ил.
4. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 2. Пер. с англ. - М.: «Русская Редакция», 2002.- 624 с.: ил.
5. Троелсен Э. Язык программирования С# 2010 и платформа .NET 4.0. Пер. с англ. - М.: Издательский дом "Вильямс", 2011. — 1392 с.: ил.
6. Фленов М.Е. Библия С#. - СПб.: БХВ-Петербург, 2011. – 560с.: ил.

Исполнитель

Студент учебной группы РИ11  
Петренко А.О.

Руководитель

Доцент кафедры информационных технологий и мехатроники  
Тимонин В.А.

«    » сентября 2015 г.

«    » сентября 2015 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ  
УНИВЕРСИТЕТ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МЕХАТРОНИКИ

**КУРСОВОЙ ПРОЕКТ (РАБОТА)**

по дисциплине «Объектно-ориентированное программирование»  
на тему «Разработка приложения для решения трансцендентных уравнений  
графическим способом»

студенту 1-го курса учебной группы РИ11  
направления подготовки «Компьютерные науки»  
специальности «Информационные управляющие  
системы и технологии на автомобильном транспорте»  
Петренко А.О.

Руководитель:

доцент кафедры  
старший научный сотрудник  
кандидат технических наук  
Тимонин В.А.

Национальная шкала \_\_\_\_\_  
Количество баллов: \_\_\_\_ Оценка ECTS: \_\_\_\_

Члены комиссии:

_____	_____
(подпись)	(фамилия и инициалы)
_____	_____
(подпись)	(фамилия и инициалы)
_____	_____
(подпись)	(фамилия и инициалы)

Харьков 2015



## СОДЕРЖАНИЕ

Введение	17
1. Описание работы	18
1.1. Постановка задачи	18
1.2. Математическое описание задачи	18
1.3. Алгоритм решения задачи	20
1.4. Описание приложения	29
1.4.1. Спецификация приложения	29
1.4.2. Исходный код приложения	32
1.4.3. Описание интерфейса приложения	38
1.5. Результаты решения задачи	39
Заключение	41
Список использованных источников и литературы	42
Приложение А. Руководство пользователя	43

## ВВЕДЕНИЕ

Трансцендентное уравнение – это уравнение вида  $f(x) = g(x)$ , где функции  $f(x)$  и  $g(x)$  являются аналитическими функциями, и, по крайней мере, одна из них не является алгебраической.

При решении трансцендентных уравнений пользуются приближенными методами, одним из которых является графическим способ. При графическом способе решения уравнений получение точных значений корней уравнений затруднено.

Решение уравнений на компьютере позволяет использовать в качестве графической поверхности экран монитора.

Для разработки программного обеспечения использован язык C#, как наиболее популярный и имеющий достаточную функциональность для решения поставленных задач. Язык C# постоянно находится в состоянии изменения, поскольку происходит его дополнение новыми функциональными возможностями.

## 1. ОПИСАНИЕ РАБОТЫ

### 1.1. Постановка задачи

Разработать приложения, с помощью которого реализуется графический способ решения трансцендентных уравнений, и провести оценку точности решения трансцендентных уравнений.

В качестве входных данных использовать уравнение

$$\frac{(x+2)^2(x-1)^3}{2} - 4\sin(x) = 0,$$

а поиск корней уравнения осуществлять для значений  $x \in [-2.5, 2.1]$ .

Разработанное приложение «Решение уравнения» должно:

- реализовывать все способы решения;
- отображать на экране монитора персонального компьютера:
  - графики функций;
  - значения корней (при щелчке левой кнопкой мыши на соответствующей точке графика).

Результаты решения должны сохраняться в текстовом файле.

### 1.2. Математическое описание задачи

Уравнения делятся на два класса – это алгебраические и трансцендентные. Первыми называют уравнения, которые содержат лишь алгебраические функции, то есть целые, рациональные или иррациональные. Уравнения, которые содержат прочие функции (тригонометрические, показательные либо логарифмические и иные) именуется трансцендентными.

Для ряда уравнений иногда вообще нельзя указать никакого способа, который позволял бы найти корни абсолютно точно. В таком случае приходится ограничиваться нахождением лишь приближенных значений корней. Современная математика располагает эффективными методами приближенного решения уравнений. Одним из таких методов является графический метод.

В основе графического метода лежит нахождение точек пересечения графиков функций левой и правой частей уравнения. Обычно данный метод применяется, когда графики обеих частей уравнения достаточно просто строятся и легко находятся точки пересечения этих графиков.

Приближенное решение как алгебраических, так и трансцендентных уравнений можно найти графически по одному из следующих способов:

1-й способ. Для решения уравнения  $f(x) = 0$  строится график  $y = f(x)$  и снимаются абсциссы тех точек, где график пересекает ось ОХ. Например, решить уравнение  $x^3 - 9x^2 + 24x - 18 = 0$ . Строим график  $y = x^3 - 9x^2 + 24x - 18$  и снимаем абсциссы  $x_1 = 1.3$ ,  $x_2 = 3$ ,  $x_3 = 4.7$  (рис.1).

2-й способ. Уравнение  $f(x) = 0$  представляется в виде  $f_1(x) = f_2(x)$ , причем одна из функций  $f_1(x)$ ,  $f_2(x)$  произвольна. Это необходимо для того, чтобы легко можно было построить графики  $y = f_1(x)$  и  $y = f_2(x)$ . Строим графики  $y = f_1(x)$  и  $y = f_2(x)$  и снимаем абсциссы точек пересечения графиков, которые являются приближенными корнями исходного уравнения  $f(x) = 0$ . Например, решить уравнение  $3x - \cos(x) - 1 = 0$ . Представим данное уравнение в виде  $3x - 1 = \cos(x)$ . Строим графики функций  $y = 3x - 1$  и  $y = \cos(x)$  снимаем абсциссу точки пересечения  $x_1 = 0.6$  (рис.2).

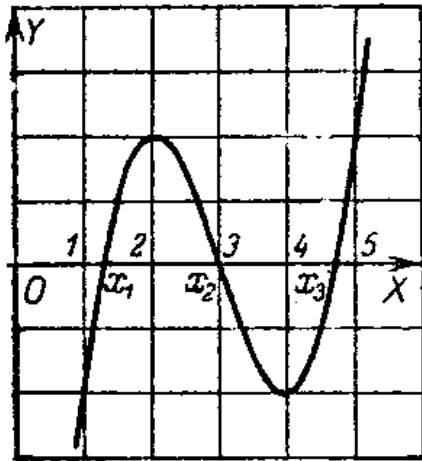


Рисунок 1

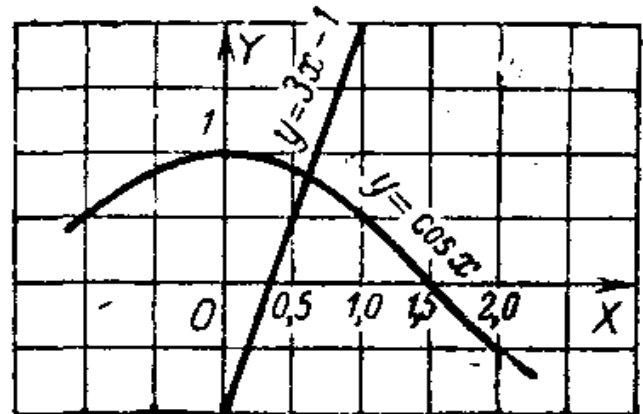


Рисунок 2

Для нахождения корней по 1-му способу строим график уравнения

$$\frac{(x+2)^2(x-1)^3}{2} - 4\sin(x) = 0$$

Абсциссы точек пересечения графика оси OX являются корнями заданного уравнения.

Для нахождения корней по 2-му способу уравнение

$$\frac{(x+2)^2(x-1)^3}{2} - 4\sin(x) = 0$$

представляем в виде  $\frac{(x+2)^2(x-1)^3}{2} = 4\sin(x)$  и построим графики функций

$$f_1(x) = \frac{(x+2)^2(x-1)^3}{2} \text{ и } f_2(x) = 4\sin(x).$$

Абсциссы точек пересечения графиков являются корнями заданного уравнения.

### **1.3. Алгоритм решения задачи**

Схема работы приложения представлена на рис.3, блок-схемы обработчика события `button1_Click`, `button2_Click` и обработчика события `pictureBox1_MouseClick` на рис.4, рис.5 и рис.6 соответственно.

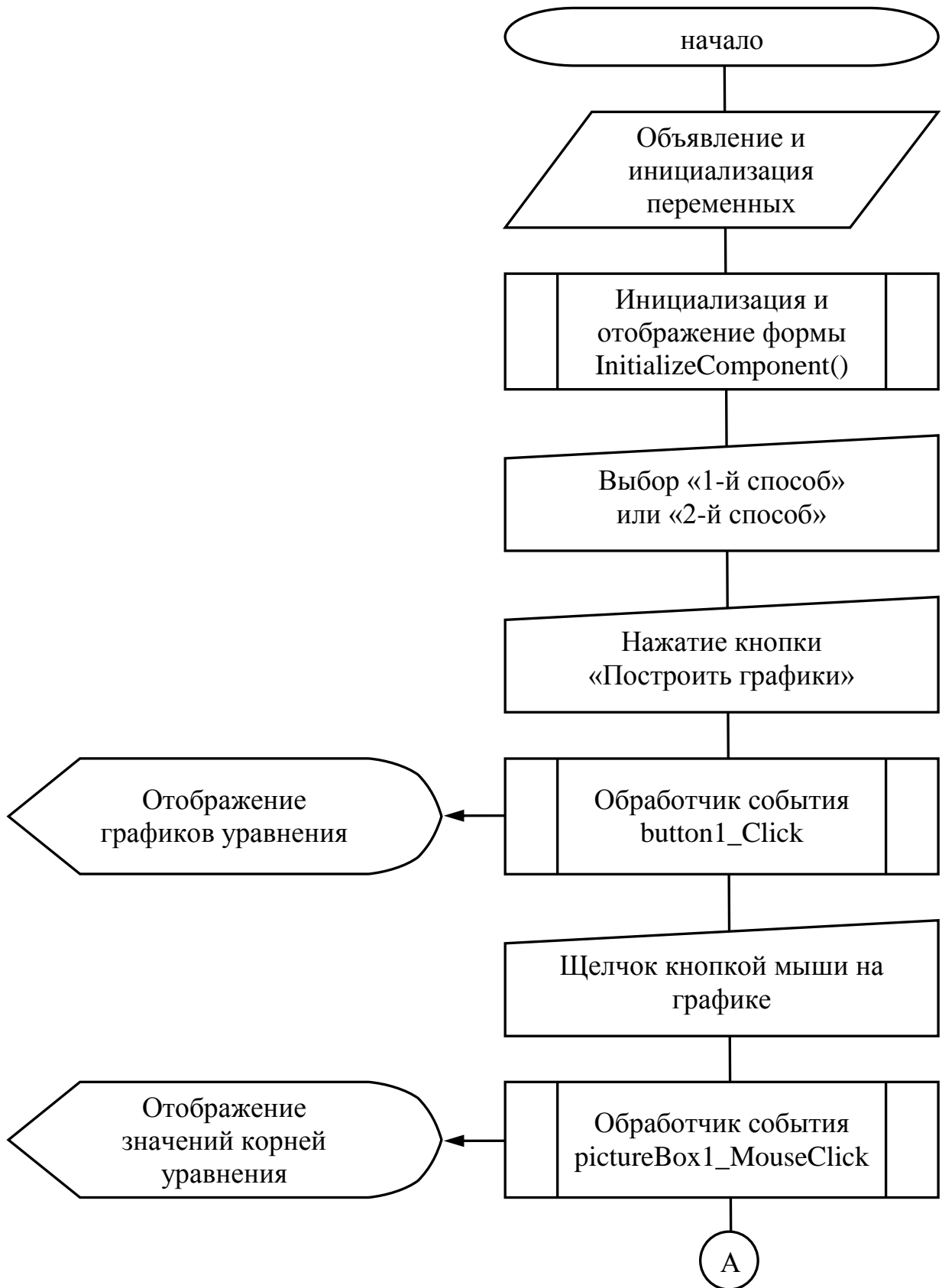


Рисунок 3 – Схема работы приложения

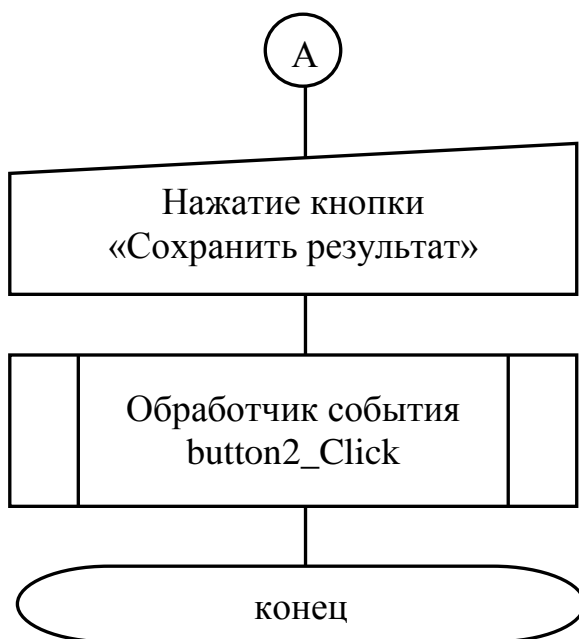


Рисунок 3 – Схема работы приложения (продолжение)



Рисунок 4 – Блок-схема обработчика события button1\_Click

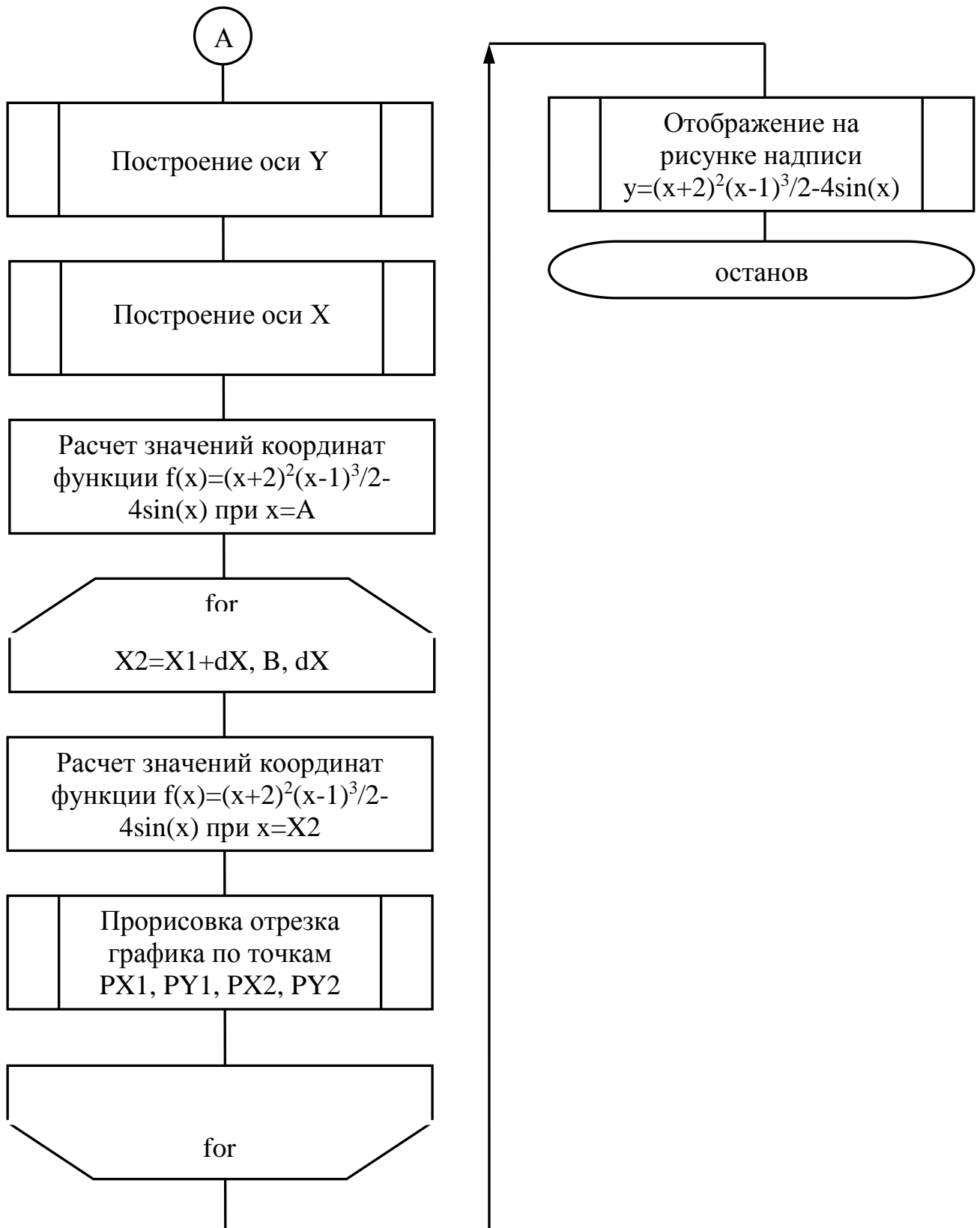


Рисунок 4 – Блок-схема обработчика события button1\_Click (продолжение)



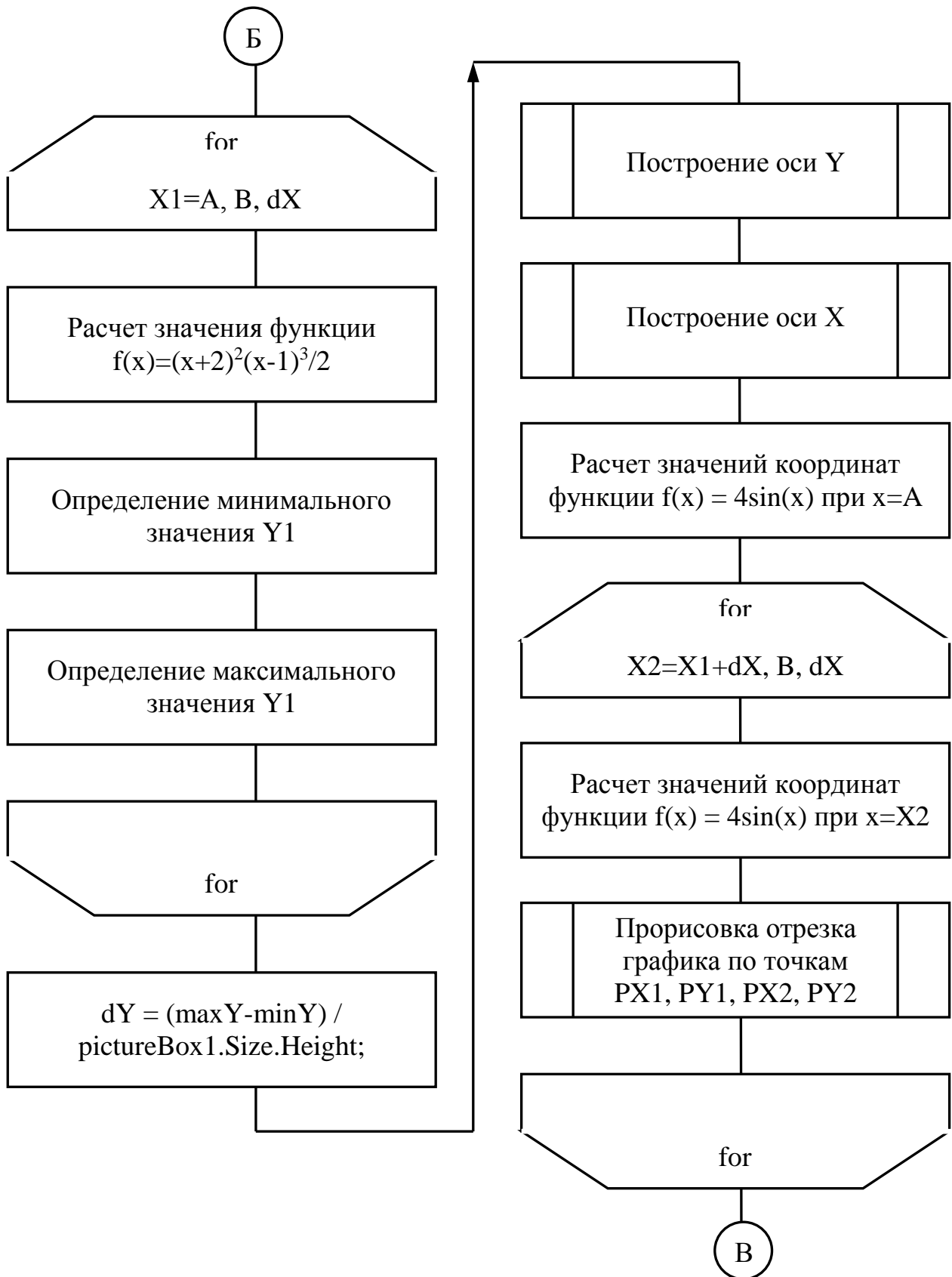


Рисунок 4 – Блок-схема обработчика события `button1_Click` (продолжение)

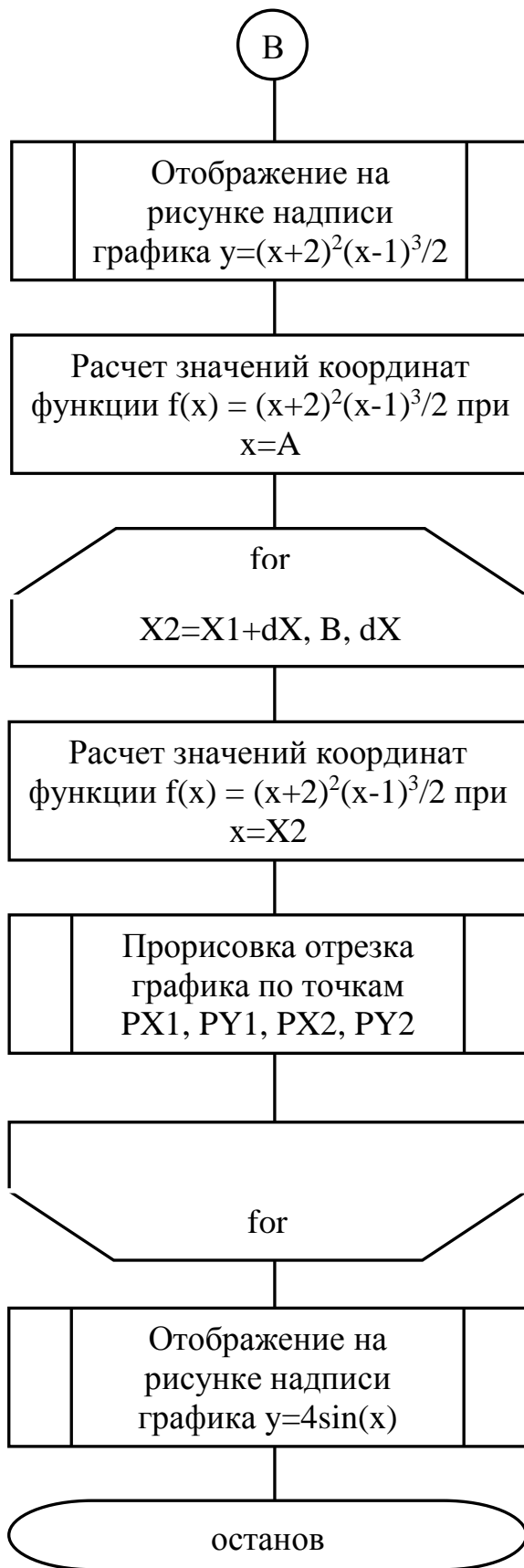


Рисунок 4 – Блок-схема обработчика события button1\_Click (продолжение)

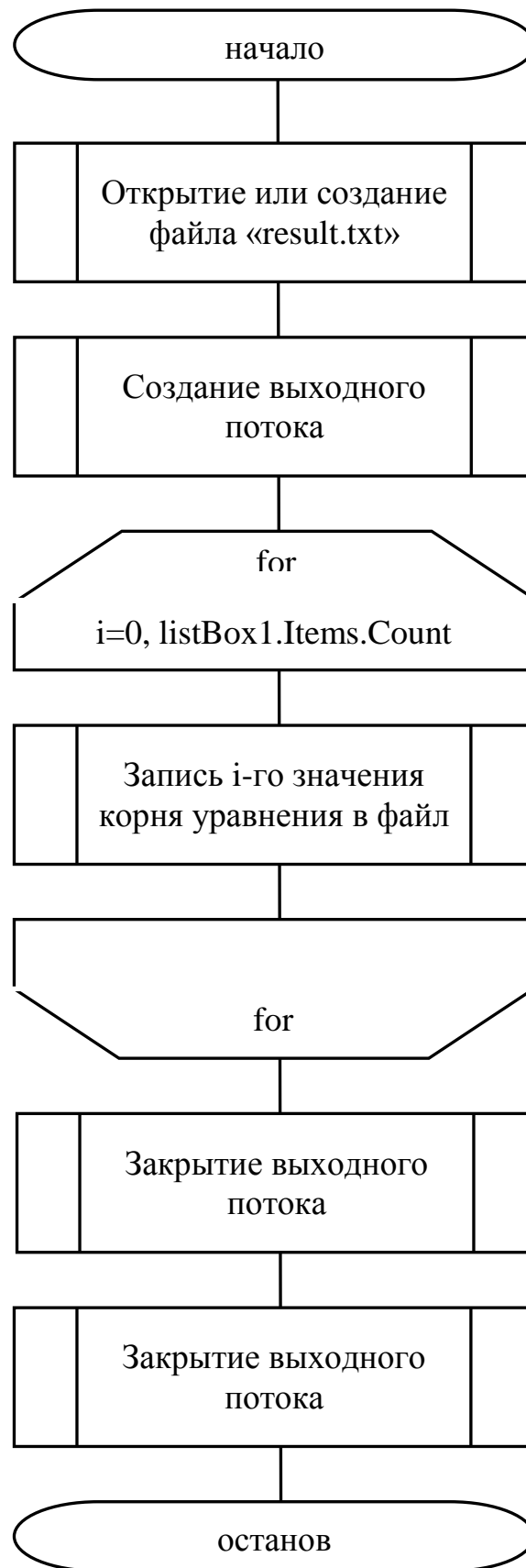


Рисунок 5 – Блок-схема обработчика события button2\_Click

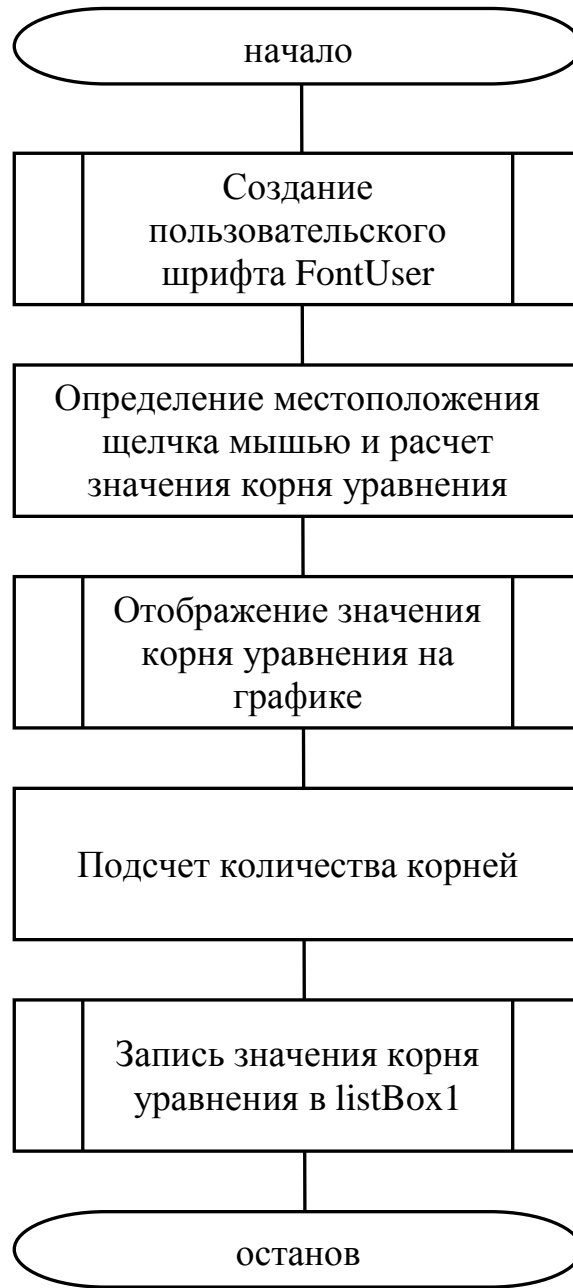


Рисунок 6 – Блок-схема обработчика события pictureBox1\_MouseClick

## 1.4. Описание приложения

### 1.4.1. Спецификация приложения

В состав приложения «Решение уравнения» входят следующие файлы и папки:

- Zadanie1.sln – содержит информацию о составе приложение;
- Zadanie1.csproj – файл, содержащий все необходимые характеристики проекта;
- Program.cs – главный модуль приложения, содержит исходный код, в котором находится точка входа для приложения;
- Form1.cs – модуль формы, содержит исходный код приложения;
- Form1.Designer.cs – записывает код, реализующий все действия, связанные с элементами управления;
- Form1.resx – файл, содержащий ресурсы формы Form1;
- папка bin – содержит папку debug, в которой находятся откомпилированные файлы:
  - Zadanie1.exe – содержит исполняемый файл приложения;
  - Zadanie1.pdb – содержит формат для хранения отладочной информации о скомпилированном исполняемом файле \*.exe;
  - Zadanie1.vshost.exe – файл главного процесса предназначен для ускорения запуска приложения;
  - Zadanie1.vshost.exe.manifest – конфигурационный файл главного процесса, предназначенный для ускорения запуска приложения;
- папка Properties – содержит совокупность файлов, в которых хранятся сведения о конфигурации проекта.

В состав приложения входят стандартные объекты классов языка C# и элементы класса Form1, разработанного на базе класса Form. Спецификация класса Form1 представлена в таблице 1.

Таблица 1 – Спецификация класса Form1

Тип элемента	Имя в программе	Структура данных	Тип данных	Назначение
поле	im	объект	Graphics	Поверхность рисования
поле	dX	простая	double	Значение x в одном пикселе поверхности

				рисования
поле	A	простая	double	Минимальное значение интервала изменения переменной x (константа равная - 2,5)
поле	B	простая	double	Максимальное значение интервала изменения переменной x (константа равная 2,1)
поле	CountX	простая	int	Количество корней уравнения
метод	Form1()			Конструктор по умолчанию
метод	button1_Click()			Обработчик события «Щелчок на кнопке». Спецификация обработчика события представлена в таблице 2.
метод	button2_Click()			Обработчик события «Щелчок на кнопке «Сохранить результат»». Спецификация обработчика события представлена в таблице 3.
метод	pictureBox1_MouseClick()			Обработчик события «Щелчок кнопкой мыши на рисунке». Спецификация обработчика события представлена в таблице 4.

Таблица 2 – Спецификация обработчика события button1\_Click

Переменная	Математическое обозначение	Имя в программе	Структура данных	Тип данных	Назначение переменной
x1		X1	простая	double	Значение переменной x в точке x1, используемое при рисовании отрезка прямой
y1		Y1	простая	double	Значение переменной y в точке y1, используемое при рисовании отрезка прямой
x2		X2	простая	double	Значение переменной x в точке x2, используемое при рисовании отрезка прямой
y2		Y2	простая	double	Значение переменной y в точке y2, используемое при рисовании отрезка прямой
px1		PX1	простая	int	Значение переменной x в пикселях в точке px1, используемое при рисовании отрезка прямой
py1		PY1	простая	int	Значение переменной y в точке py1, используемое при рисовании отрезка прямой
px2		PX2	простая	int	Значение переменной x в точке px2, используемое при рисовании отрезка прямой
py2		PY2	простая	int	Значение переменной y в точке py2, используемое при рисовании отрезка прямой
minY		minY	простая	double	Минимальное значение функции $f(x) = \frac{(x+2)^2(x-1)^3}{2}$
maxY		maxY	простая	double	Максимальное значение функции $f(x) = \frac{(x+2)^2(x-1)^3}{2}$
dy		dY	простая	double	Значение y в одном пикселе поверхности рисования

PenUser		PenUser	объект	Pen	Пользовательское перо
---------	--	---------	--------	-----	-----------------------

Таблица 3 – Спецификация обработчика события button2\_Click

Переменная	Математическое обозначение	Имя в программе	Структура данных	Тип данных	Назначение переменной
fout		fout	объект	FileStream	Идентификатор файлового потока
fstr_out		fstr_out	объект	StreamWriter	Идентификатор выходного потока

Таблица 4 – Спецификация обработчика события pictureBox1\_MouseClick

Переменная	Математическое обозначение	Имя в программе	Структура данных	Тип данных	Назначение переменной
str		Str	простая	string	Значение корня уравнения в текстовом формате
FontUser		FontUser	объект	Font	Пользовательский шрифт

#### 1.4.2. Исходный код приложения

Главный модуль приложения, содержащий исходный код, в котором находится точка входа для приложения, находится в файле Program.cs и имеет вид:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
```

```
namespace Zadanie1
{
    static class Program
```



```

{
    /// <summary>
    /// Главная точка входа для приложения.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }
}

```

Метод `Application.Run()` запускает скрытый цикл обработки сообщений ОС и делает видимой форму приложения «Решение уравнения», инициализируя объект класса `Form1`.

Модуль формы, содержащий исходный код приложения, находится в файле `Form1.cs` и имеет вид:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Zадание1
{
    public partial class Form1 : Form
    {
        Graphics im;
        double dX;
        double A = -2.5, B = 2.1; // границы интервала
        int CountX = 0; // количество корней
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    CountX = 0;
    listBox1.Items.Clear();
    double X1, Y1, X2, Y2, minY, maxY, dY;
    int PX1, PY1, PX2, PY2;
    im = pictureBox1.CreateGraphics();
    im.Clear(Color.White);
    Pen PenUser = new System.Drawing.Pen(Color.Black, 2);
    dX = (B-A) / pictureBox1.Size.Width;
    if (radioButton1.Checked==true)
    {
        for (X1 = A, minY = Double.MaxValue, maxY = Double.MinValue;
            X1 < B; X1 += dX)
        {
            Y1 = Math.Pow((X1 + 2), 2) * Math.Pow((X1 - 1), 3) / 2 -
                4*Math.Sin(X1); // функция y=f1(x)

            if (Y1 < minY) minY = Y1;
            if (Y1 > maxY) maxY = Y1;
        }
        dY = (maxY - minY) / pictureBox1.Size.Height;
        // Строим ось Y
        im.DrawLine(System.Drawing.Pens.Black, (int)(-A / dX), 0,
                    (int)(-A / dX), pictureBox1.Size.Height);
        im.DrawLine(System.Drawing.Pens.Black, (int)(-A / dX), 0,
                    (int)(-A / dX - 3), 10);
        im.DrawLine(System.Drawing.Pens.Black, (int)(-A / dX), 0,
                    (int)(-A / dX + 3), 10);
        im.DrawString("Y", Font, Brushes.Black, (int)(-A / dX - 15), 3);
        // Строим ось X
        im.DrawLine(System.Drawing.Pens.Black, 0, pictureBox1.Size.Height / 2,
                    pictureBox1.Size.Width, pictureBox1.Size.Height / 2);
        im.DrawLine(System.Drawing.Pens.Black, (pictureBox1.Size.Width - 10),
                    (pictureBox1.Size.Height / 2 - 3),
                    pictureBox1.Size.Width, pictureBox1.Size.Height / 2);
        im.DrawLine(System.Drawing.Pens.Black, (pictureBox1.Size.Width - 10),
                    (pictureBox1.Size.Height / 2 + 3),
                    pictureBox1.Size.Width, pictureBox1.Size.Height / 2);
        im.DrawString("X", Font, Brushes.Black, (pictureBox1.Size.Width - 15),
                    (pictureBox1.Size.Height / 2 + 3));
        X1 = A; PX1 = 0;
        Y1 = Math.Pow((X1 + 2), 2) * Math.Pow((X1 - 1), 3) / 2 -

```

```

                                4 * Math.Sin(X1); // функция y=f1(x)
PY1 = (int)(pictureBox1.Size.Height / 2 - Y1 / dY);
for (X2 = X1 + dX, PX2 = 1; X2 < B;
     X2 = X2 + dX, PX1 = PX2, PX2++, PY1 = PY2)
{
    Y2 = Math.Pow((X2 + 2), 2) * Math.Pow((X2 - 1), 3) / 2 -
        4 * Math.Sin(X2);
    PY2 = (int)(pictureBox1.Size.Height / 2 - Y2 / dY);
    im.DrawLine(PenUser, PX1, PY1, PX2, PY2);
}
im.DrawString("(x+2)", Font, Brushes.Black,
               (pictureBox1.Size.Width - 130), 8);
im.DrawString("2", Font, Brushes.Black,
               (pictureBox1.Size.Width - 103), 3);
im.DrawString("(x-1)", Font, Brushes.Black,
               (pictureBox1.Size.Width - 97), 8);
im.DrawString("3", Font, Brushes.Black, (pictureBox1.Size.Width - 73), 3);
im.DrawString("y = ", Font, Brushes.Black,
               (pictureBox1.Size.Width - 147), 15);
im.DrawString("_____ ", Font, Brushes.Black,
               (pictureBox1.Size.Width - 130), 10);
im.DrawString("2", Font, Brushes.Black,
               (pictureBox1.Size.Width - 103), 23);
im.DrawString(" - 4sin(x)", Font, Brushes.Black,
               (pictureBox1.Size.Width - 67), 14);
}
else
{
    for (X1 = A, minY = Double.MaxValue, maxY = Double.MinValue;
         X1 < B; X1 += dX)
    {
        Y1 = Math.Pow((X1 + 2), 2) * Math.Pow((X1 - 1), 3) / 2;
        if (Y1 < minY) minY = Y1;
        if (Y1 > maxY) maxY = Y1;
    }
    dY = (maxY - minY) / pictureBox1.Size.Height;
    // Строим ось Y
    im.DrawLine(System.Drawing.Pens.Black, (int)(-A / dX), 0,
                (int)(-A / dX), pictureBox1.Size.Height);
    im.DrawLine(System.Drawing.Pens.Black, (int)(-A / dX), 0,
                (int)(-A / dX - 3), 10);
    im.DrawLine(System.Drawing.Pens.Black, (int)(-A / dX), 0,

```

```

                (int)(-A / dX + 3), 10);
im.DrawString("Y", Font, Brushes.Black, (int)(-A / dX - 15), 3);
// Строим ось X
im.DrawLine(System.Drawing.Pens.Black, 0, pictureBox1.Size.Height / 2,
            pictureBox1.Size.Width, pictureBox1.Size.Height / 2);
im.DrawLine(System.Drawing.Pens.Black, (pictureBox1.Size.Width - 10),
            (pictureBox1.Size.Height / 2 - 3),
            pictureBox1.Size.Width, pictureBox1.Size.Height / 2);
im.DrawLine(System.Drawing.Pens.Black, (pictureBox1.Size.Width - 10),
            (pictureBox1.Size.Height / 2 + 3),
            pictureBox1.Size.Width, pictureBox1.Size.Height / 2);
im.DrawString("X", Font, Brushes.Black, (pictureBox1.Size.Width - 15),
            (pictureBox1.Size.Height / 2 + 3));
X1 = A; PX1 = 0;
Y1 = 4 * Math.Sin(X1);
PY1 = (int)(pictureBox1.Size.Height / 2 - Y1 / dY);
for (X2 = X1 + dX, PX2 = 1; X2 < B;
        X2 = X2 + dX, PX1 = PX2, PX2++, PY1 = PY2)
{
    Y2 = 4 * Math.Sin(X2);
    PY2 = (int)(pictureBox1.Size.Height / 2 - Y2 / dY);
    im.DrawLine(PenUser, PX1, PY1, PX2, PY2);
}
im.DrawString("(x+2)", Font, Brushes.Black,
                (pictureBox1.Size.Width - 80), 8);
im.DrawString("2", Font, Brushes.Black, (pictureBox1.Size.Width - 53), 3);
im.DrawString("(x-1)", Font, Brushes.Black,
                (pictureBox1.Size.Width - 47), 8);
im.DrawString("3", Font, Brushes.Black, (pictureBox1.Size.Width - 23), 3);
im.DrawString("y = ", Font, Brushes.Black,
                (pictureBox1.Size.Width - 97), 15);
im.DrawString("_____ ", Font, Brushes.Black,
                (pictureBox1.Size.Width - 80), 10);
im.DrawString("2", Font, Brushes.Black,
                (pictureBox1.Size.Width - 53), 23);
X1 = A; PX1 = 0;
Y1 = Math.Pow((X1 + 2), 2) * Math.Pow((X1 - 1), 3) / 2;
PY1 = (int)(pictureBox1.Size.Height / 2 - Y1 / dY);
for (X2 = X1 + dX, PX2 = 1; X2 < B;
        X2 = X2 + dX, PX1 = PX2, PX2++, PY1 = PY2)
{
    Y2 = Math.Pow((X2 + 2), 2) * Math.Pow((X2 - 1), 3) / 2;

```

```

        PY2 = (int)(pictureBox1.Size.Height / 2 - Y2 / dY);
        im.DrawLine(PenUser, PX1, PY1, PX2, PY2);
    }
    im.DrawString("y = 4sin(x)", Font, Brushes.Black,
                  (pictureBox1.Size.Width - 97), 46);
}
}

private void button2_Click(object sender, EventArgs e)
{
    FileStream fout;
    try
    {
        fout = new FileStream("result.txt", FileMode.OpenOrCreate);
    }
    catch (IOException exc)
    {
        MessageBox.Show(exc.Message, "Ошибка при открытии файла");
        return;
    }
    StreamWriter fstr_out = new StreamWriter(fout);
    try
    {
        for (int i=0; i<listBox1.Items.Count; i++)
        {
            fstr_out.Write(listBox1.Items[i].ToString());
        }
    }
    catch (IOException exc)
    {
        MessageBox.Show(exc.Message, "Ошибка при записи в файл");
        return;
    }
    fstr_out.Close();
    fout.Close();
}

private void pictureBox1_MouseClick(object sender, MouseEventArgs e)
{
    Font FontUser = new System.Drawing.Font("Times New Roman",
                                             12, FontStyle.Bold);
    string Str = (e.X*dX+A).ToString("f2");
}

```

```

im.DrawString(Str, FontUser, Brushes.Red, e.X-10, e.Y-20);
CountX++;
listBox1.Items.Add("X" + CountX.ToString() + " = " + Str);
    }
}
}

```

### 1.4.3. Описание интерфейса приложения

При разработке приложения использовались следующие компоненты:

- pictureBox1 – предназначен для отображения результата выполнения задачи в виде графика, на котором при нажатии кнопки мыши должны отображаться значения корней уравнения;
- groupBox1 – предназначен для создания группы элементов, связанных по выбору вариантов решения;
- radioButton1 (кнопка «1-й способ») – предназначен для выбора способа решения;
- radioButton2 (кнопка «2-й способ») – предназначен для выбора способа решения;
- button1 (кнопка «Построить графики») – предназначен для построения графиков функций в зависимости от выбранного способа решения;
- groupBox2 – предназначен для создания группы элементов, связанных по отображению и сохранения результатов решения;
- listBox1 – предназначен для сохранения и отображения списка значений корней уравнения;
- button2 (кнопка «Сохранить результат») – предназначен для сохранения результатов решения в файле на внешнем носителе.

Форма приложения и используемые компоненты представлены на рис.7.

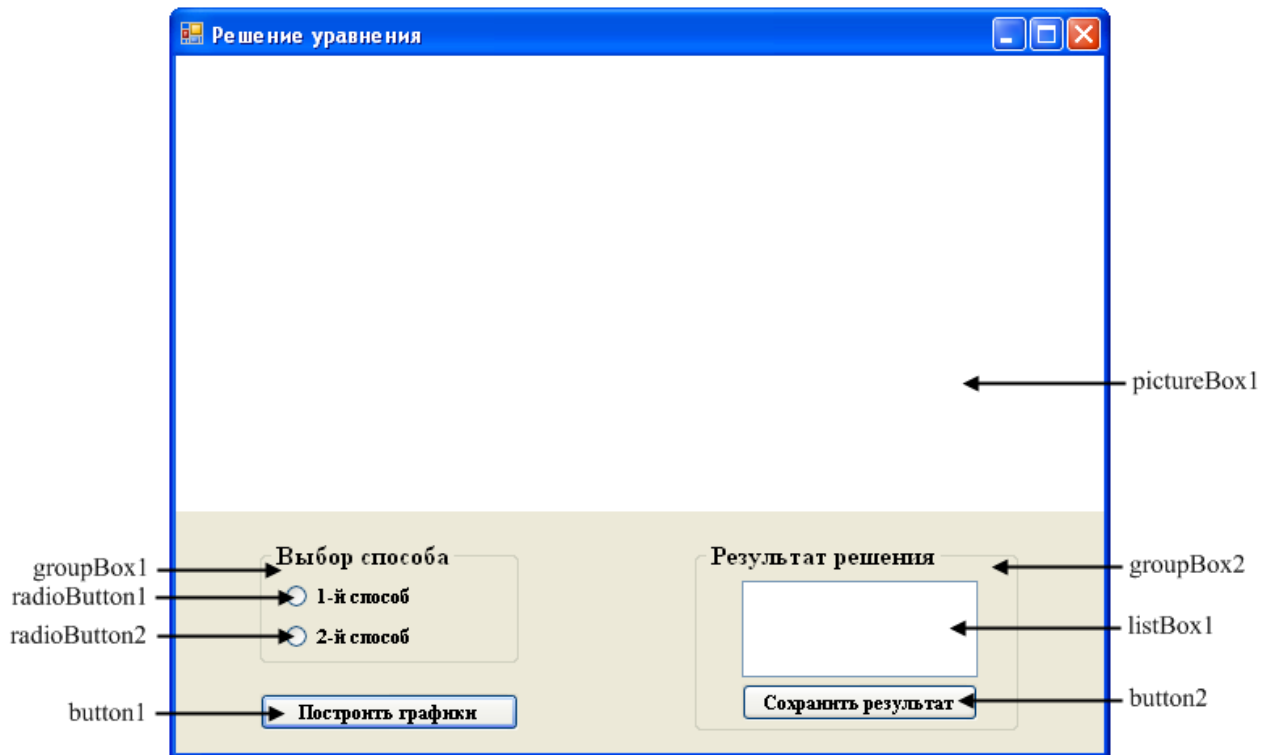


Рисунок 7 – Внешний вид формы приложения «Решение уравнения»

### 1.5. Результаты решения задачи

Результатом решения уравнения  $\frac{(x+2)^2(x-1)^3}{2} - 4\sin(x) = 0$  являются:

- по 1-му способу значения  $x_1=-2.37$ ,  $x_2=-1.14$ ,  $x_3=1.81$  (рис.8). Подставив данные значения в заданное уравнение, получаем:

- при  $x_1=-2.37$  уравнение равно 0.17;
- при  $x_2=-1.14$  уравнение равно 0.01;
- при  $x_3=1.81$  уравнение равно -0.03;

- по 2-му способу значения  $x_1=-2.38$ ,  $x_2=-1.15$ ,  $x_3=1.82$  (рис.9). Подставив данные значения в заданное уравнение, получаем:

- при  $x_1=-2.38$  уравнение равно -0.02;
- при  $x_2=-1.15$  уравнение равно 0.06;
- при  $x_3=1.82$  уравнение равно 0.15;

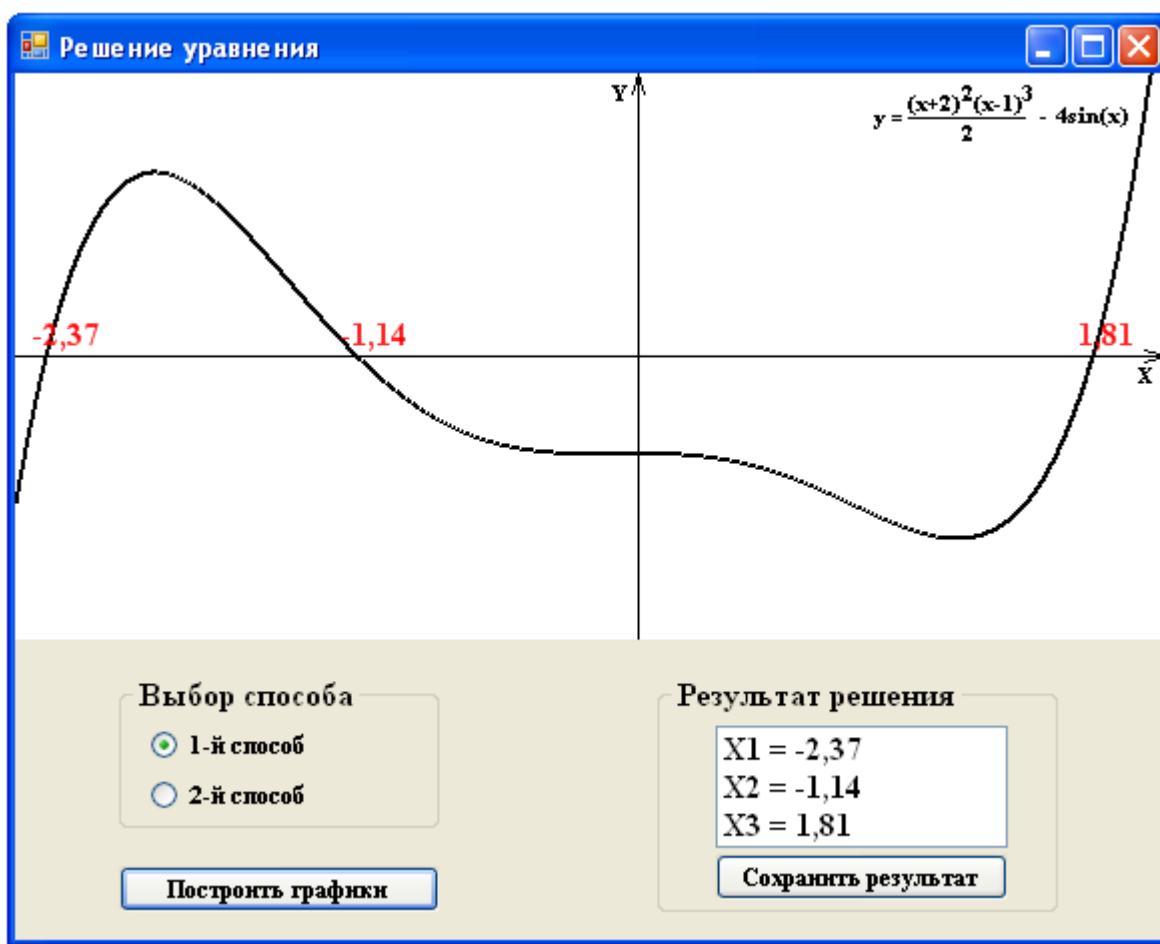


Рисунок 8 – Результат выполнения приложения «Решение уравнения» (1-й способ)



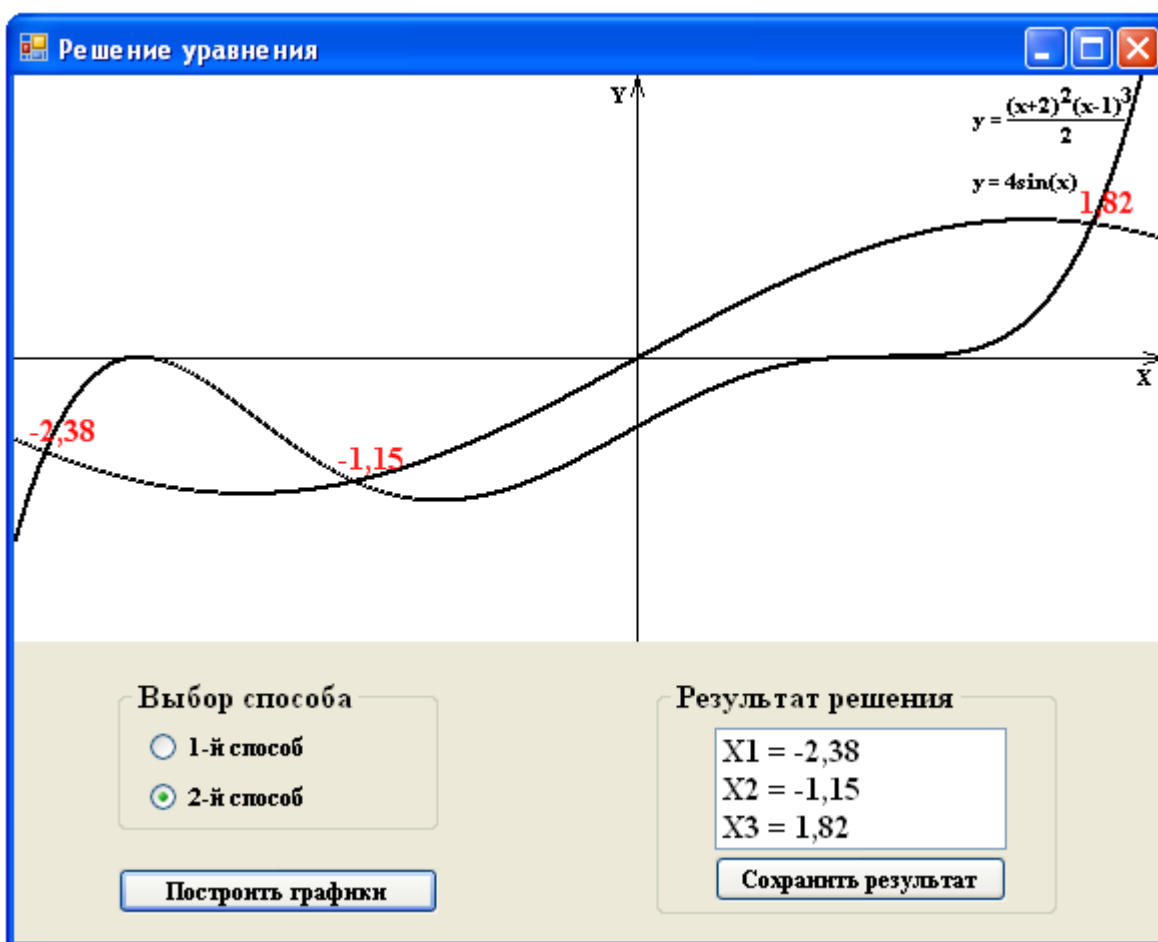


Рисунок 9 – Результат выполнения приложения «Решение уравнения» (2-й способ)

Анализ результатов работы приложения показывает, что:

- приложение функционирует правильно;
- максимальная погрешность ошибки составляет 0,17. Погрешность ошибки связана с форматом представления значения корней и точностью наведения курсора мыши на соответствующую точку. Кроме того, на точность решения уравнения влияет разрешающая способность экрана монитора.

## ЗАКЛЮЧЕНИЕ

Курсовой проект (работа) выполнен в полном объёме в соответствии с индивидуальным заданием и требованиям нормативных документов.

Курсовой проект (работа) включает в себя описание приложения «Решение уравнения» в виде пояснительной записки, проект приложения и исполняемый файл (прилагаются на CD или DVD-диске) и руководство пользователя.

Данное приложение можно усовершенствовать с целью расширения возможностей по вводу минимального и максимального значения интервала

аргумента функции  $f(x)$ , а также может быть модифицировано с целью решения любых уравнений путем ввода с клавиатуры соответствующих формул.

### **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ**

1. Выгодский М.Я. Справочник по высшей математике. – М.: АСТ: Астрель, 2010 – 704 с.: ил.
2. Голощапов А.Л. Microsoft Visual Studio 2010. – СПб.: БХВ-Петербург, 2011. – 544 с.: ил.
3. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 1. Пер. с англ. - М.: «Русская Редакция», 2002.- 576 с.: ил.
4. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 2. Пер. с англ. - М.: «Русская Редакция», 2002.- 624 с.: ил.
5. Троелсен Э. Язык программирования С# 2010 и платформа .NET 4.0. Пер. с англ. - М.: Издательский дом "Вильямс", 2011. — 1392 с.: ил.
6. Фленов М.Е. Библия С#. - СПб.: БХВ-Петербург, 2011. – 560с.: ил.

Пояснительную записку выполнил  
студент Петренко А.О.

## Приложение А

### Руководство пользователя

Приложение «Решение уравнения» предназначено для решения трансцендентных уравнений графическим способом. В настоящее время существует 2 способа решения:

- 1-й способ – построение общего графика уравнения и поиск точек пересечения графика с осью X;
- 2-й способ – разбиение общего графика на составные части и поиск точек пересечения составных графиков.

Приложение позволяет выдавать информацию в виде значений корней уравнения на графике по щелчку правой клавишей мыши и в текстовом виде в окне «Результат решения», а также записывать результаты в файл.

Порядок работы с приложением заключается в следующем:

1. Запуск приложения. Запуск приложения осуществляется с помощью исполняемого файла `Zadanie1.exe`. При запуске приложения на экране отображается первоначальная форма приложения (рис.А.1).

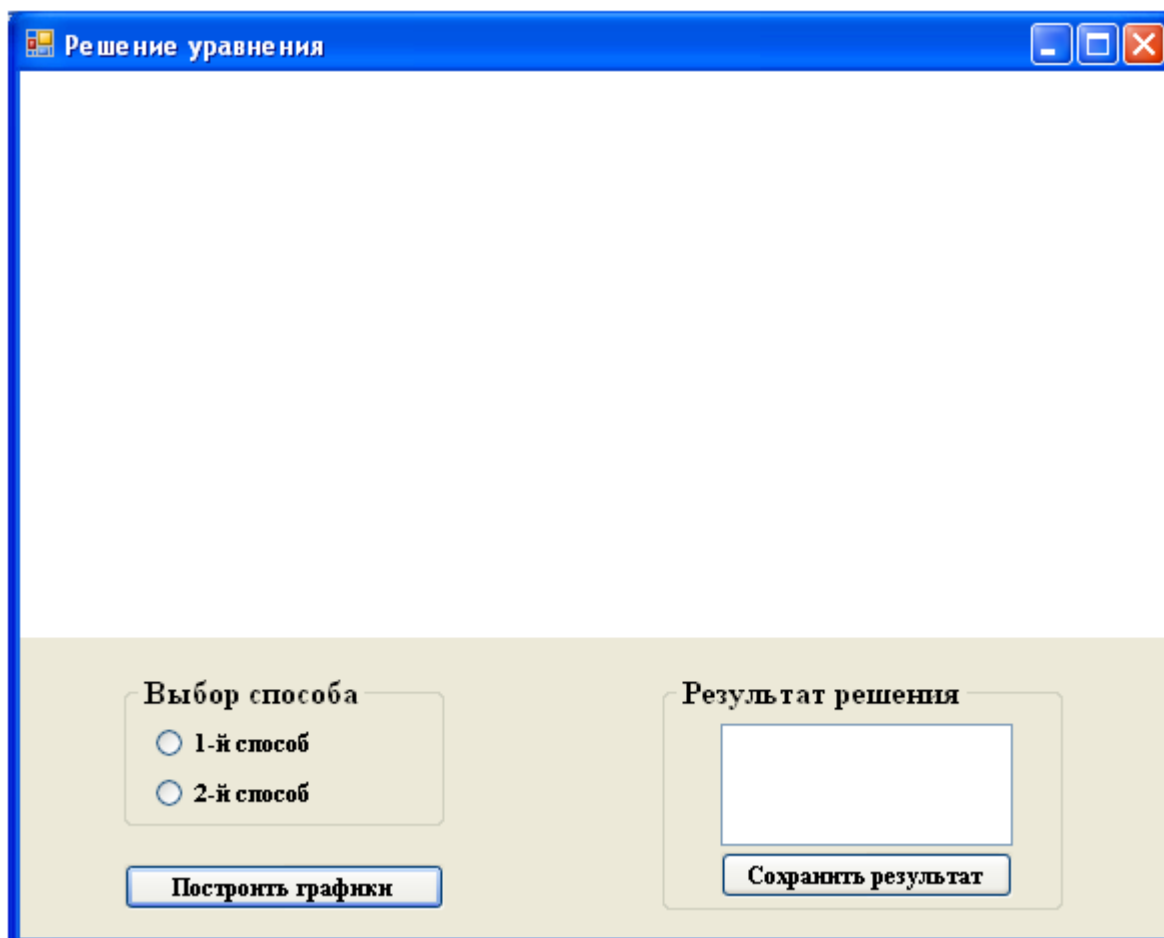


Рисунок А.1 – Вид приложения «Решение уравнения» при запуске

2. Построение графиков. Перед построением графиков необходимо выбрать способ решения уравнения (используются управляющие кнопки «1-й способ» и «2-й способ»). При нажатии кнопки «Построить графики» происходит расчет и отображение графика  $f(x) = \frac{(x+2)^2(x-1)^3}{2} - 4\sin(x)$  с осью X (рис.А.2) или графиков  $f_1(x) = \frac{(x+2)^2(x-1)^3}{2}$  и  $f_2(x) = 4\sin(x)$  (рис.А.3). На форме отображаются графики и поясняющие надписи.

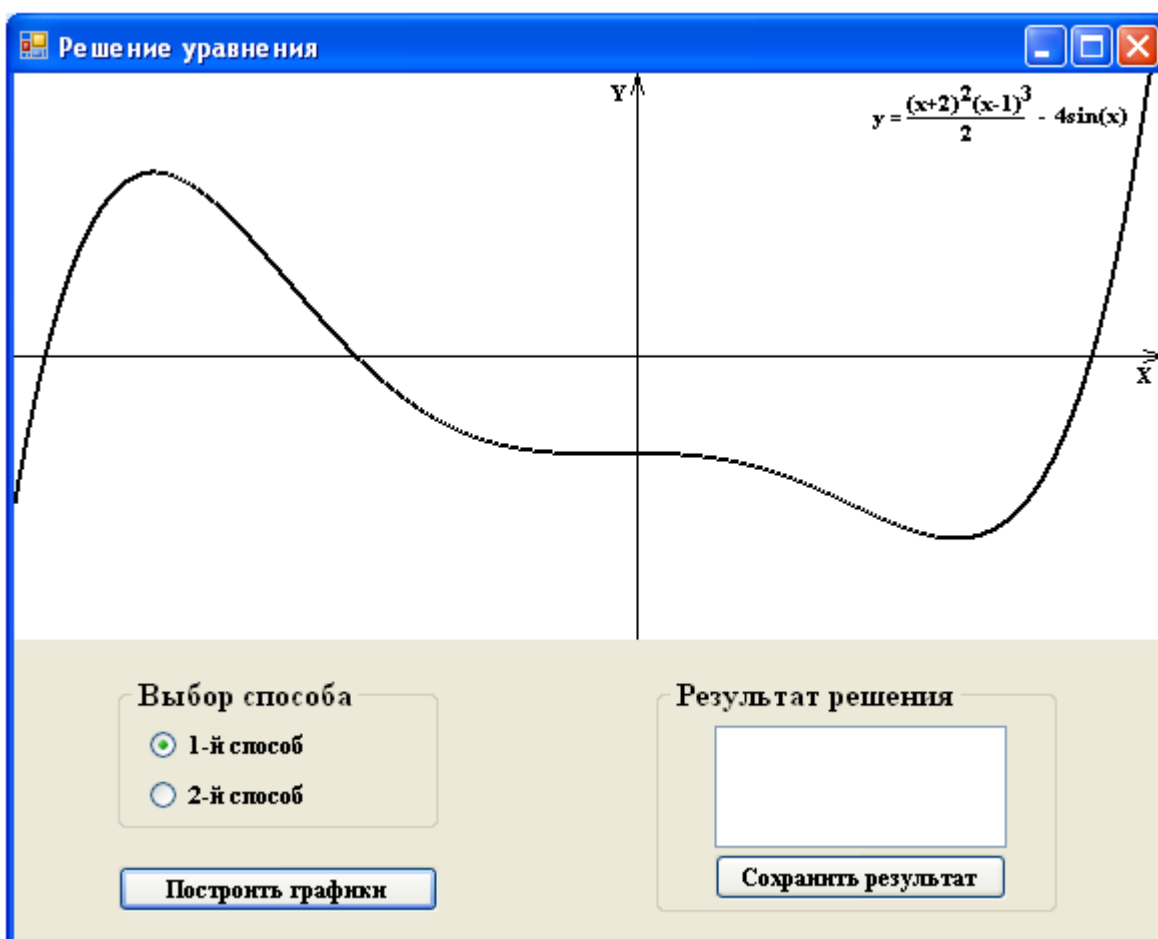


Рисунок А.2 – Вид приложения «Решение уравнения» после нажатия кнопки «Построить графики» (1-й способ)

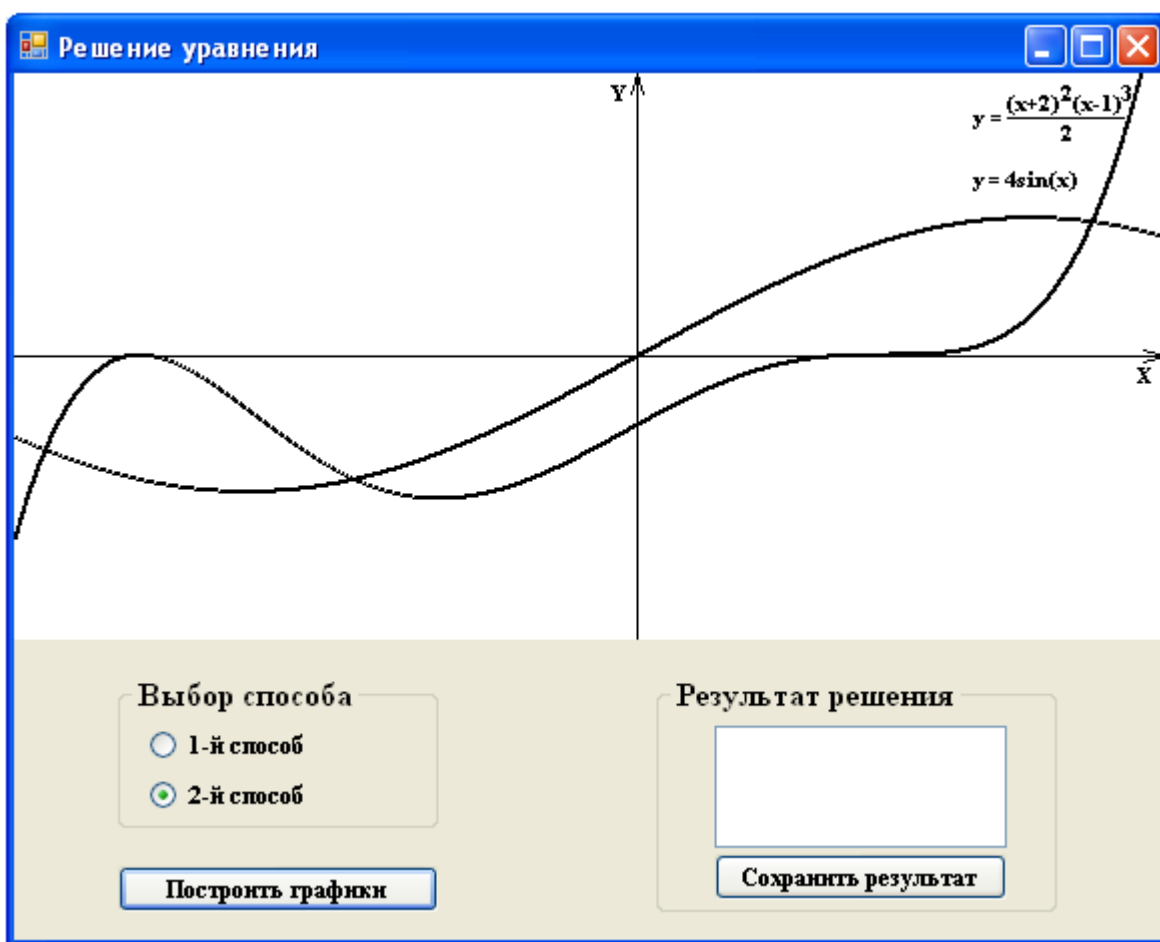


Рисунок А.3 – Вид приложения «Решение уравнения» после нажатия кнопки «Построить графики» (2-й способ)

3. Отображение результатов. Значения корней заданного уравнения отображаются при щелчке левой кнопкой мыши на соответствующей точке графика.

В соответствии с поставленной задачей результатом решения уравнения являются точки пересечения графика  $f(x) = \frac{(x+2)^2(x-1)^3}{2} - 4\sin(x)$  с осью  $X$  (рис.А.4) или графиков  $f_1(x) = \frac{(x+2)^2(x-1)^3}{2}$  и  $f_2(x) = 4\sin(x)$  (рис.А.5).

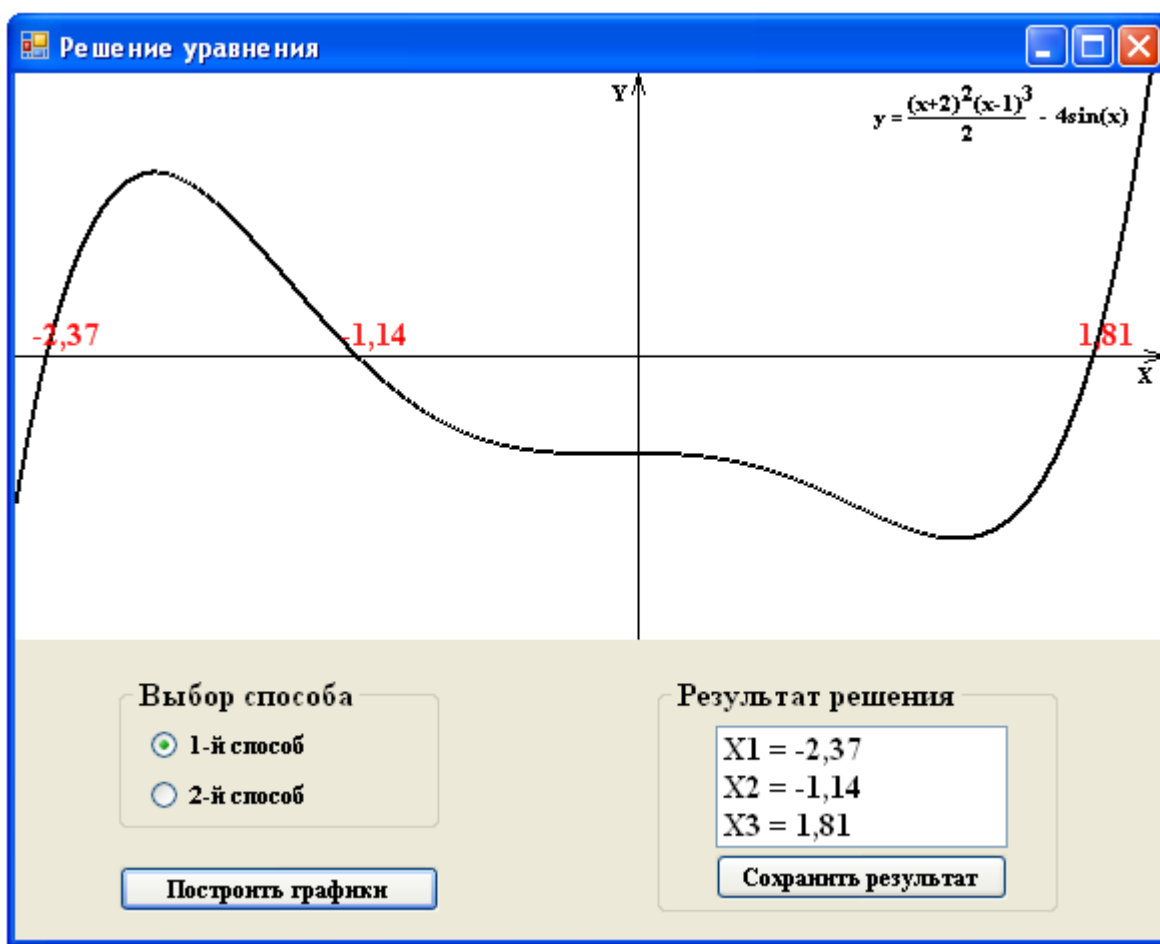


Рисунок А.4 – Результат выполнения приложения «Решение уравнения»

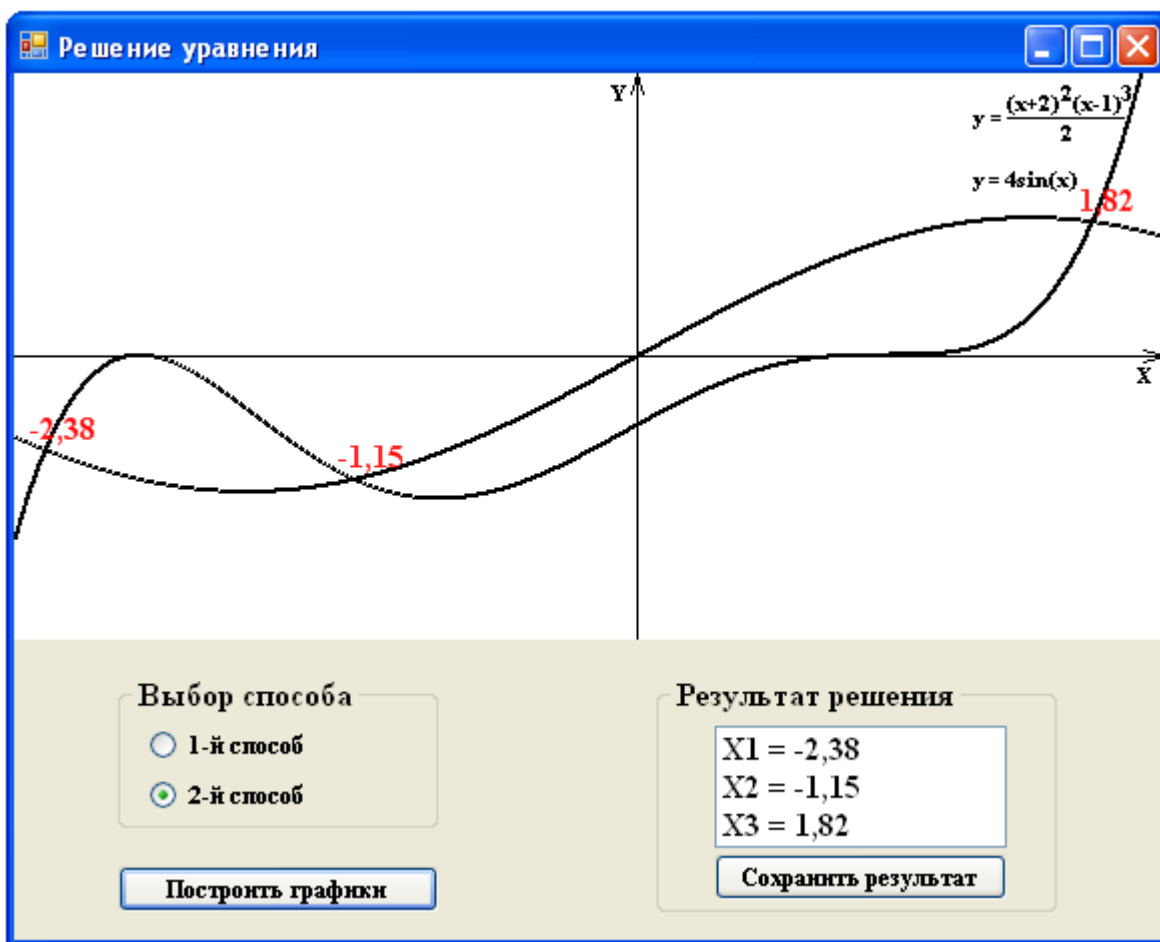


Рисунок А.5 – Результат выполнения приложения «Решение уравнения»

4. Сохранение результатов. Сохранение результатов решения на внешнем накопителе осуществляется по нажатию кнопки «Сохранить результат». Результаты решения будут записаны в файл «result.txt», в достоверности которых можно убедиться, воспользовавшись любым текстовым редактором.

При каждом использовании приложения результаты будут дописываться в файл «result.txt».

## СОДЕРЖАНИЕ

1. Цель и задачи курсового проекта (работы)	3
2. Требования к выполнению курсового проекта (работы)	3
3. Структура пояснительной записки	4
4. Методические указания по выполнению курсового проекта (работы)	6
5. Требования по оформлению курсового проекта (работы)	8
6. Порядок проведения защиты курсового проекта (работы)	10
Список использованных источников и литературы	11
Приложение 1. Пример задания на курсовой проект (работу)	12
Приложение 2. Пример пояснительной записки	15