

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Филиал в г. Славянске-на-Кубани

Кафедра математики, информатики и МП

ДОПУСТИТЬ К ЗАЩИТЕ В ГАК

Заведующий кафедрой

канд. физ.-мат. наук, доцент

_____ /А. Н. Чернышев/
(подпись)

_____ 2013 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Алгоритмизация и программирование задач по теме: «Метод координат» в
курсе «Геометрия» средней школы.

Работу выполнил _____ /В. В. Гладких/
(подпись, дата)

Специальность «Математика и информатика» _____

Научный руководитель

канд. физ.-мат. наук, доцент _____ /А. Н. Чернышев/
(подпись, дата)

Нормоконтролер

канд. физ.-мат. наук, доцент _____ /А. Н. Чернышев/
(подпись, дата)

Славянск-на-Кубани 2014

СОДЕРЖАНИЕ

Введение	3
1 «Метод координат» в курсе геометрии за 9 класс.	5
1.1 Систематизация задач в теме «Координаты вектора».	5
1.2 Систематизация задач в теме «Простейшие задачи в координатах»	7
2 Разработка и применение	
электронно - методического комплекса «Метод координат»	9
2.1 Объектно-ориентированный язык программирования C#.	9
2.2 Visual C#.	11
2.3 Открытый дистрибутив TeX для платформы Windows – MikTeX.	11
2.4 Применение электронно - методического комплекса «Метод координат».	12
2.5 Совместное применение объектно-ориентированного языка программирования C# и компилятора открытого дистрибутива TeX для платформы Windows – MikTex	16
2.6 Чтение PDF файлов в приложении	16
2.7 Генерирование исходного кода, для компиляции PDF документов	17
Заключение	26
Список использованных источников	27

ВВЕДЕНИЕ

В современном образовании, роль информационных технологий становится все значимей. Учителям и преподавателям все чаще приходится и становится необходимым использовать ИКТ на уроках. В курсе геометрии и алгебры, существует необходимость применять методику алгоритмизации. Ее применение от части уже не возможно, без ИКТ и специальных средств.

Курс IX класса является завершающим звеном в изучении планиметрии. В течении двух предыдущих лет учащиеся накапливали геометрические знания и умения, изучали свойства отрезков, углов, треугольников, четырехугольников, окружностей, доказательства.[2]

На основе всех накопленных навыков учащимися, учитель имеет возможность вводить в образовательный процесс методы алгоритмизации некоторых задач из курса геометрии, используя при этом элементы лекционно - семинарских занятий. На данном этапе ученики способны самостоятельно осваивать некоторые части учебного материала.[2]

Мы разработали электронно - методический комплекс «Метод координат», для применения на уроках геометрии в разделе курса «Метод координат». Применять пособие могут как ученики при самостоятельной работе с задачами, так и учителя при демонстрации алгоритмов решения.

Объектом исследования разработка является электронно - методического комплекса, для обучения школьников решению основных задач курса геометрии при помощи метода алгоритмизации.

Предмет исследования электронно - методический комплекс, для обучения школьников основным алгоритмам решению геометрических задач раздела «Метод координат».

Цель исследования: разработка электронно - методического комплекса «Метод координат», для обучения школьников решению типовых задач из раздела курса геометрии «Метод координат».

Для достижения цели мы использовали, объектно-ориентированный язык программирования C#, открытый дистрибутив TeX для платформы Windows – MikTeX и разработали классификацию задач, по алгоритмам, в разделе курса геометрии за 9 класс – «Метод координат».

Работа разделена на две основные части:

- а) введение;
- б) «Метод координат» в курсе геометрии за 9 класс – где рассмотрены основные задачи по теме и алгоритмы их решения;
- в) Разработка и применение электронно - методического комплекса «Метод координат» – рассмотрен комплекс «Метод координат». Приведен пример как его использовать, рассмотрен основной способ вывода алгоритмов на экран пользователя;
- г) заключение.

1 «Метод координат» в курсе геометрии за 9 класс.

Придавая геометрическим заданиям алгебраический характер, метод координат переносит в геометрию наиболее важную особенность алгебры — единообразие способов решения задач, что в свою очередь способствует введению алгоритмов их решения. Если в арифметике и элементарной геометрии приходится, как правило, искать для каждой задачи особый путь решения, то в алгебре и аналитической геометрии решения проводятся по общему для всех задач алгоритму, легко приспособляемому к любой задаче. Перенесение в геометрию свойств алгебры и поэтому обладающих большой общностью способов решения задач составляет главную ценность метода координат.

Для применения методики алгоритмизации, на уроках геометрии, глава «Метод координат» является одной из самых подходящих из всего курса геометрии за 7 – 9 класс. Разработанное приложение наглядно демонстрирует учащимся, как можно решить задачи из данной главы, путем применения не сложных алгоритмов. Применение электронно-методических пособий позволяет учителю эффективно использовать информационные технологии на уроках геометрии. Учитывая тенденции информатизации всего образовательного процесса, ожидается что такие пособия будут пользоваться популярностью не только на уроках геометрии и алгебры, но и в остальных предметах школьного курса.

1.1 Систематизация задач в теме «Координаты вектора».

Назначение темы - ввести понятие координат вектора и рассмотреть правила действий над векторами с заданными координатами. Некоторые понятия ученикам уже известны из предшествующих тем в курсе геометрии. В частности учащиеся уже владеют такими понятиями как вектор, сумма векторов, разность векторов и произведение вектора на число. Так же учащимся известно что при умножении данного вектора на число получается вектор, коллинеарный данному.[2]

В данной теме учащимся дается представление о координатных векторах, разложении вектора по координатным векторам, сумме двух или более

векторов, разности двух векторов и произведении вектора на число. Часть предложенных задач, можно использовать при применении алгоритмизации. [2]

В таблице 1 представлена систематизация задач по учебнику геометрии за 7 – 9 класс [1], где указаны задания и алгоритмы их решения:

Таблица 1 — Систематизация задач в теме «Координаты вектора».

№ п/п	Задача	Алгоритм
1	Найдите такое число k , чтобы выполнялось равенство $\vec{n} = k\vec{m}$, если: а) Векторы \vec{m} и \vec{n} противоположно направлены. б) Векторы \vec{m} и \vec{n} сонаправлены.	Так как $\vec{n} = \vec{m}k$, то $ \vec{n} = \vec{m} k \Rightarrow k = \frac{ \vec{n} }{ \vec{m} }$ а) Так как $\vec{m} \uparrow\downarrow \vec{n}$, то $k < 0$; б) Так как $\vec{m} \uparrow\uparrow \vec{n}$, то $k > 0$.
2	Найдите координаты вектора $\vec{a} = x\vec{i} + y\vec{j}$.	Так как $\vec{a} = x\vec{i} + y\vec{j}$, то $\vec{a}\{x; y\}$
3	Найдите разложение по координатным векторам \vec{i} и \vec{j} вектора $\vec{a}\{-x; y\}$.	Так как $\vec{a} = x_1 \cdot \vec{i} + x_2 \cdot \vec{j}$, где $x_1 = -x; x_2 = y$, то $\vec{a} = -x\vec{i} + y\vec{j}$
4	Найдите координаты вектора: а) $\alpha\vec{a} + \beta\vec{b}$, если $\vec{a}\{x_1; y_1\}; \vec{b}\{x_2; y_2\}$; б) $\alpha\vec{a} - \beta\vec{b}$, если $\vec{a}\{x_1; y_1\}; \vec{b}\{x_2; y_2\}$;	а) Так как $\vec{a}\{x_1; y_1\}, \vec{b}\{x_2; y_2\} : \alpha\vec{a} = \{x_1 \cdot \alpha; y_1 \cdot \alpha\}; \beta\vec{b} = \{x_2 \cdot \beta; y_2 \cdot \beta\}; (\alpha\vec{a} + \beta\vec{b})\{x_1 \cdot \alpha + x_2 \cdot \beta; y_1 \cdot \alpha + y_2 \cdot \beta\}$; б) Так как $\vec{a}\{x_1; y_1\}, \vec{b}\{x_2; y_2\} : \alpha\vec{a} = \{x_1 \cdot \alpha; y_1 \cdot \alpha\}; \beta\vec{b} = \{x_2 \cdot \beta; y_2 \cdot \beta\}; (\alpha\vec{a} - \beta\vec{b})\{x_1 \cdot \alpha - x_2 \cdot \beta; y_1 \cdot \alpha - y_2 \cdot \beta\}$;
5	Найти попарно коллинеарные векторы если: $\vec{a}\{x_1; y_1\}; \vec{b}\{x_2; y_2\}; \vec{c}\{x_3; y_3\}$	а) $\vec{a} \parallel \vec{b} \Rightarrow \frac{x_1}{x_2} = \frac{y_1}{y_2} = k_1$ б) $\vec{a} \parallel \vec{c} \Rightarrow \frac{x_1}{x_3} = \frac{y_1}{y_3} = k_2$ в) $\vec{c} \parallel \vec{b} \Rightarrow \frac{x_3}{x_2} = \frac{y_3}{y_2} = k_3$ Если полученные равенства – верны, то векторы коллинеарны.

Продолжение таблицы 1

6	Векторы \vec{a} и \vec{b} не коллинеарны. Найдите числа x и y , удовлетворяющие равенству: $n\vec{a} + tx\vec{a} + m\vec{b} + py\vec{b} = \vec{0}$	Так как \vec{a} и \vec{b} не коллинеарны, то $(n+t \cdot x)\vec{a} + (m+p \cdot y)\vec{b} = \vec{0} \Rightarrow n+t \cdot x = 0$ и $m+p \cdot y = 0$, где x и y - неизвестные.
---	---	---

Из таблицы 1 видно, что задачи для закрепления материала достаточно просты и очевидны. Но алгоритмы решения приведенных заданий важны при изучении последующего курса геометрии и будут применяться ни один раз.

1.2 Систематизация задач в теме «Простейшие задачи в координатах»

Назначение темы – рассмотреть простейшие алгоритмы решения задач в координатах и как они используются при решении более сложных задач методом координат.

В таблице 2 представлена систематизация задач, где указаны задания и алгоритмы их решения:

Таблица 2 — Систематизация задач в теме «Координаты вектора».

№ п/п	Задача	Алгоритм
1	Точка A лежит на положительной полуоси Ox , а точка B – на положительной полуоси Oy . Найдите координаты вершин треугольника ABO , если $OA = 5$, $OB = 3$.	Так как O – точка начала координат, то $O(0; 0)$. Так как $OA = 5$; $OB = 3 \Rightarrow A(5; 0), B(0; 3)$.
2	Найдите координаты вершины D параллелограмма $ABCD$, если $A(0; 0)$, $B(5; 0)$; $C(12, -3)$	$\vec{AC} = \vec{AD} + \vec{AB} \Rightarrow \vec{AD} = \vec{AC} - \vec{AB}$ – по свойству параллелограмма. $D(7; -3)$, т.к. $x = x_C - x_B = 7$; $y = y_C = -3$.
3	Найдите координаты вектора \vec{AB} , если известны координаты его начала и конца: $A(2; 7)$; $B(-2, 7)$.	Так как $\vec{AB} = \{x_B - x_A; y_B - y_A\} = \{-2 - 2; 7 - 7\} = \{-4; 0\}$
3.1	Найдите координаты начала вектора \vec{AB} , если известно, что: $\vec{AB}\{-4; 0\}$; $B(-2, 7)$.	Так как $\vec{AB} = \{x_B - x_A; y_B - y_A\}$, то $x_A = x_B - x_{AB}$; $y_A = y_B - y_{AB} \Rightarrow x_A = -2 - -4 = 2$; $y_A = 7 - 0 = 7$

Продолжение таблицы 2

3.2	Найдите координаты конца вектора \vec{AB} , если известно, что: $\vec{AB}\{-4; 0\}$; $A(2, 7)$.	Так как $\vec{AB} = \{x_B - x_A; y_B - y_A\}$, то $x_B = x_A + x_{AB}$; $y_B = y_A + y_{AB} \Rightarrow x_B = -4 + 2 = -2$; $y_B = 0 + 7 = 7$
4	Найдите длину вектора $\vec{a}\{5; 9\}$	$ \vec{a} = \sqrt{x^2 + y^2} = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$
4.1	Найдите x , если $\vec{a}\{x; 3\}$, $ \vec{a} = 5$	$ \vec{a} = \sqrt{x^2 + y^2} \Rightarrow x = \sqrt{ \vec{a} ^2 - y^2} \Rightarrow x_1 = \sqrt{25 - 9} = 4$; $x_2 = -4$
4.2	Найдите y , если $\vec{a}\{-4; y\}$, $ \vec{a} = 5$	$ \vec{a} = \sqrt{x^2 + y^2} \Rightarrow y = \sqrt{ \vec{a} ^2 - x^2} \Rightarrow y_1 = \sqrt{25 - 16} = 3$; $y_2 = -3$
5	Даны точки $A(0; 1)$ и $C(5; -3)$. Найдите координаты точки B , если известно, что точка B – середина отрезка AC	Так как B – середина AC , то $x_B = \frac{x_C + x_A}{2}$; $y_B = \frac{y_C + y_A}{2} \Rightarrow x_B = \frac{5+0}{2} = 2,5$; $y_B = \frac{-3+1}{2} = -1$
5.1	Даны точки $A(0; 1)$ и $B(2, 5; -1)$. Найдите координаты точки C , если известно, что точка B – середина отрезка AC	Так как B – середина AC , то $x_C = 2 \cdot x_B - x_A$; $y_C = 2 \cdot y_B - y_A \Rightarrow x_C = 2 \cdot 2,5 - 0 = 5$; $y_C = 2 \cdot -1 - 1 = -3$
6	Найдите расстояние между точками A и B , если: $A(2; 7)$; $B(-5; 7)$	$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} = \sqrt{(-2 - 2)^2 + (7 - 7)^2} = \sqrt{16 + 0} = 4$;
6.1	Найдите x , если известно что: $ \vec{AB} = 7$; $A(x; 7)$; $B(-5; 7)$	Так как $ \vec{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$, то $7 = \sqrt{(-5 - x)^2 + (7 - 7)^2} \Rightarrow 7 = \sqrt{(-5 - x)^2} \Rightarrow 49 = (-5 - x)^2 \Rightarrow x_1 = -12$; $x_2 = 2$
6.2	Найдите y , если известно что: $ \vec{AB} = 7$; $A(2; 7)$; $B(-5; y)$	Так как $ \vec{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$, то $7 = \sqrt{(-5 - 2)^2 + (y - 7)^2} \Rightarrow 7 = \sqrt{49 + (y - 7)^2} \Rightarrow 49 = 49 + (y - 7)^2 \Rightarrow y_1 = 7$; $y_2 = 7$
7	Найдите периметр треугольника MNP , если $M(4; 0)$; $N(12; -2)$; $P(5; -9)$	$MN = \sqrt{(x_M - x_N)^2 + (y_M - y_N)^2} = \sqrt{(12 - 4)^2 + (-2)^2} = 8,2462$; $NP = \sqrt{(x_N - x_P)^2 + (y_N - y_P)^2} = \sqrt{(12 - 5)^2 + (-2 + 9)^2} = 9,8995$; $MP = \sqrt{(x_P - x_M)^2 + (y_P - y_M)^2} = \sqrt{(5 - 4)^2 + (-9)^2} = 9,0554$; $P_{\triangle MNP} = MN + NP + MP = 8,2462 + 9,8995 + 9,0554 = 27,2011$

2 Разработка и применение

электронно - методического комплекса «Метод координат»

Цель разработки электроно - методического комплекса «Метод координат» – разработка настольного приложения для обучения школьников решению типовых задач из раздела курса геометрии «Метод координат», методом алгоритмизации. Для достижения поставленной цели использовались:

- Объектно-ориентированный язык программирования C#.
- Microsoft Visual C# 2013.
- Открытый дистрибутив TeX для платформы Windows – MikTeX.
- Систематизация задач [1,2], по алгоритмам, в разделе курса геометрии за 9 класс – «Метод координат».

2.1 Объектно-ориентированный язык программирования C#.

Синтаксис C# очень выразителен, но прост в изучении. Все, кто знаком с языками C, C++ или Java с легкостью узнают синтаксис с фигурными скобками, характерный для языка C#. Разработчики, знающие любой из этих языков, как правило, смогут добиться эффективной работы с языком C# за очень короткое время. Синтаксис C# делает проще то, что было сложно в C++, и обеспечивает мощные возможности, такие как типы значений Nullable, перечисления, делегаты, лямбда-выражения и прямой доступ к памяти, чего нет в Java. C# поддерживает универсальные методы и типы, обеспечивая более высокий уровень безопасности и производительности, а также итераторы, позволяющие при реализации коллекций классов определять собственное поведение итерации, которое может легко использоваться в клиентском коде. Выражения LINQ делают строго типизированный запрос очень удобной языковой конструкцией.[3]

Как объектно-ориентированный язык, C# поддерживает понятия инкапсуляции, наследования и полиморфизма. Все переменные и методы, включая метод Main – точку входа приложения – инкапсулируются в определения классов. Класс может наследовать непосредственно из одного родительского класса, но может реализовывать любое число интерфейсов. Для

методов, которые переопределяют виртуальные методы в родительском классе, необходимо ключевое слово `override`, чтобы исключить случайное повторное определение. В языке `C#` структура похожа на облегченный класс: это тип, распределяемый по стопкам, реализующий интерфейсы, но не поддерживающий наследование.[3]

В дополнение к основным описанным объектно-ориентированным принципам, язык `C#` упрощает разработку компонентов программного обеспечения благодаря нескольким инновационным конструкциям языка, в число которых входят следующие:

- Инкапсулированные сигнатуры методов, называемые делегатами, которые поддерживают типобезопасные уведомления о событиях.
- Свойства, выступающие в роли методов доступа для закрытых переменных-членов.
- Атрибуты с декларативными метаданными о типах во время выполнения.
- Встроенные комментарии XML-документации.
- LINQ, предлагающий встроенные возможности запросов в различных источниках данных.

[3] Если потребуется обеспечить взаимодействие с другим программным обеспечением Windows, таким как объекты COM или собственные библиотеки DLL Win32, в языке `C#` можно использовать процесс, который называется «Interop.» Процесс Interop позволяет программам на `C #` выполнять практически любые действия, которые может выполнять исходное приложение на `C++`. Язык `C#` поддерживает даже указатели и понятие «небезопасного» кода для тех случаев, когда прямой доступ к памяти имеет крайне важное значение.[3]

Процесс построения `C#` по сравнению с `C` и `C++` прост и является более гибким, чем в Java. Нет отдельных файлов заголовка, а методы и типы не требуется объявлять в определенном порядке. В исходном файле `C#` может быть определено любое число классов, структур, интерфейсов и событий. [3]

2.2 Visual C#.

C# (произносится «Си-шарп») является языком программирования, который разработан для создания множества приложений, работающих в среде .NET Framework. Язык C# прост, типобезопасен и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую C-подобным языкам.[4]

Visual C# — это реализация языка C# корпорацией Майкрософт. Поддержка Visual C# в Visual Studio обеспечивается с помощью полнофункционального редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, мощного и удобного отладчика и многих других средств. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и к другим полезным, хорошо спроектированным классам, что существенно ускоряет цикл разработки. [4]

2.3 Открытый дистрибутив TeX для платформы Windows – MikTeX.

LATEX — это настольная издательская система. Её применения простираются от подготовки одностраничных писем до создания многотомных фолиантов. Реализации LATEX'а существуют для всех типов компьютеров. LATEX упрощает работу с текстом, позволяя сосредоточить внимание на его содержании. Заботы по оформлению текста LATEX принимает на себя. В исходном виде документ LATEX является текстовым файлом и поэтому одинаково пригоден для компьютера в издательском офисе в Нью-Йорке, Мадриде или Новосибирске. Редакции научных журналов рекомендуют, а иногда и вынуждают готовить статьи в системе LATEX и принимают их по электронной почте. Заменяв всего лишь одно слово — название класса печатного документа в преамбуле входного файла, издатель придаст тексту тот облик, который отличает выбранный журнал и который при ином методе общения с издательством требует немалых затрат времени. [5]

MiKTeX — открытый (open source) дистрибутив TeX для платформы Windows. Одним из существенных достоинств MiKTeX является возмож-

ность автоматического обновления установленных компонентов и пакетов. Особенности последних версии MiKTeX (2.7-2.9) является интегрированная поддержка XeTeX, LuaTeX, MetaPost, pdfTeX и совместимость с Windows Vista, Windows Server 2008, Windows 7 и Windows 8.

В настоящее время в состав MiKTeX включены:

- Текстовый редактор TeXworks.
- Компиляторы TeX, pdfTeX, XeTeX и LuaTeX
- Различные варианты TeX: e-TeX, Omega, NTS;
- Конверторы TeX в PDF: Dvipdfm/Dvipdfmx;
- MetaPost
- Полный набор общеиспользуемых макропакетов: LaTeX, ConTeXt и др.
- Средство просмотра Yip

[6]

2.4 Применение электронно - методического комплекса «Метод координат».

Приложение имеет оконный интерфейс, который позволяет легко и интуитивно использовать все функции приложения. Работа с программой начинается с того, что перед пользователем открывается главное окно «Метод координат» [Рисунок – 1].

В главном окне пользователь может посредством выбора типов задач из древовидного списка, выбрать и просмотреть нужный пример задачи. [Рисунок – 2]

В этом же окне выбрав пункт меню «Задачи» пользователь может обратиться к нужному типу задач и вызвать необходимое окно для работы с задачами [Рисунок – 3].

Для получения конкретных результатов и алгоритмов, пользователю необходимо вбить данные в вновь открытом окне, в соответствии с условием задачи. Например, выбираем раздел «2. Простейшие задачи в координатах». далее подраздел «2.2 Найти координаты вершины D», в открытом окне [Рисунок – 4] можно посмотреть пример задачи [Рисунок – 5], нажав на кнопку «Пример» или ввести данные и по нажатию на кнопку «Найти координаты D» и получить полный алгоритм решения задачи, с учетом указанных дан-

ных [Рисунок – 6] и [Рисунок – 7].

Таким образом ученик или учитель могут использовать программу для работы с алгоритмами решения задач из геометрии. Интерфейс программы построен однотипно, а значит увидев на примере одной из задач, как использовать данное приложение, пользователь сможет работать и с остальными задачами.

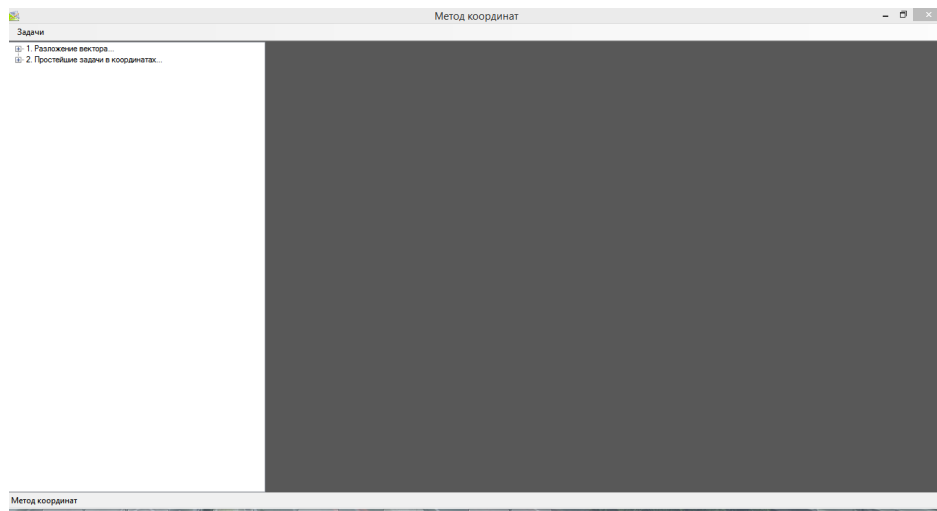


Рисунок 1 – Главное окно приложения Метод координат

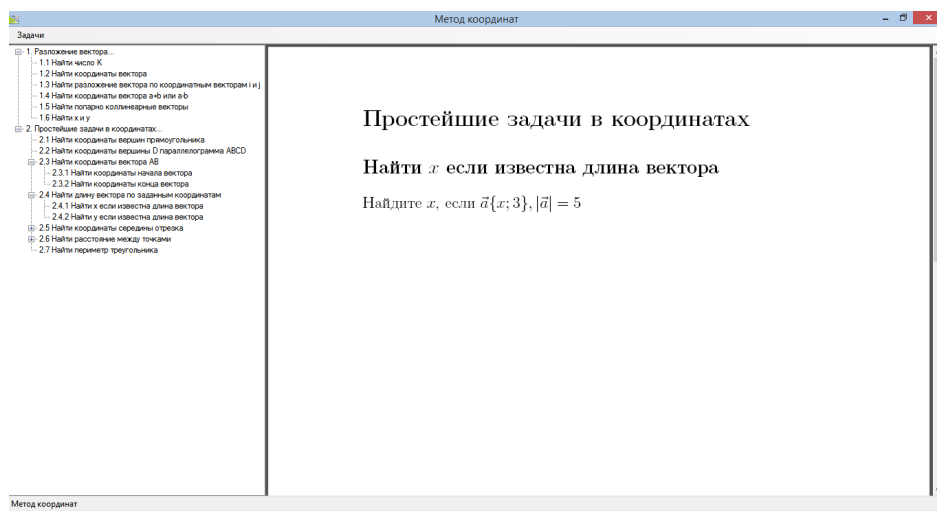


Рисунок 2 – Древовидный список примеров задач

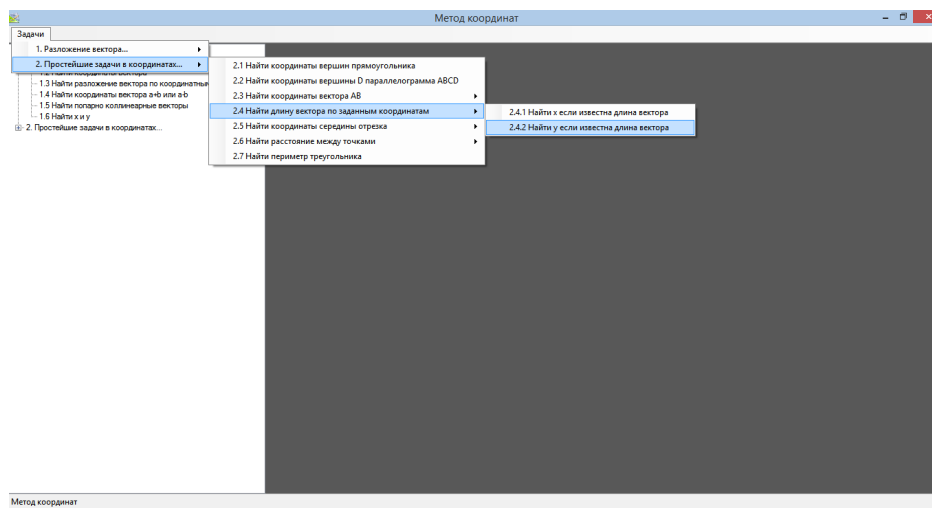


Рисунок 3 – Выбор задачи

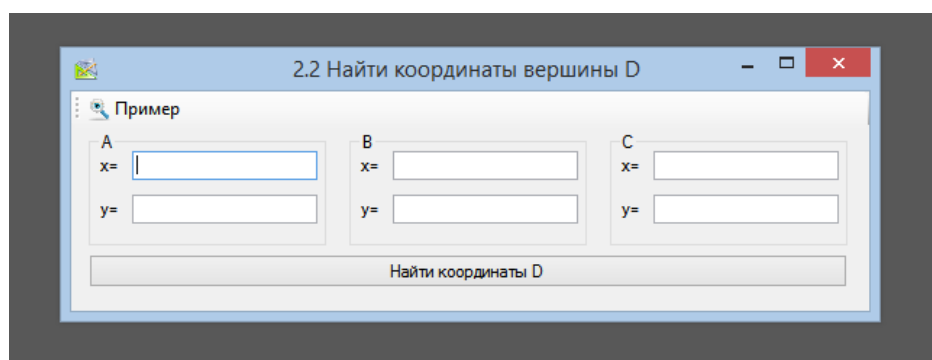


Рисунок 4 – Окно ввода данных для задачи

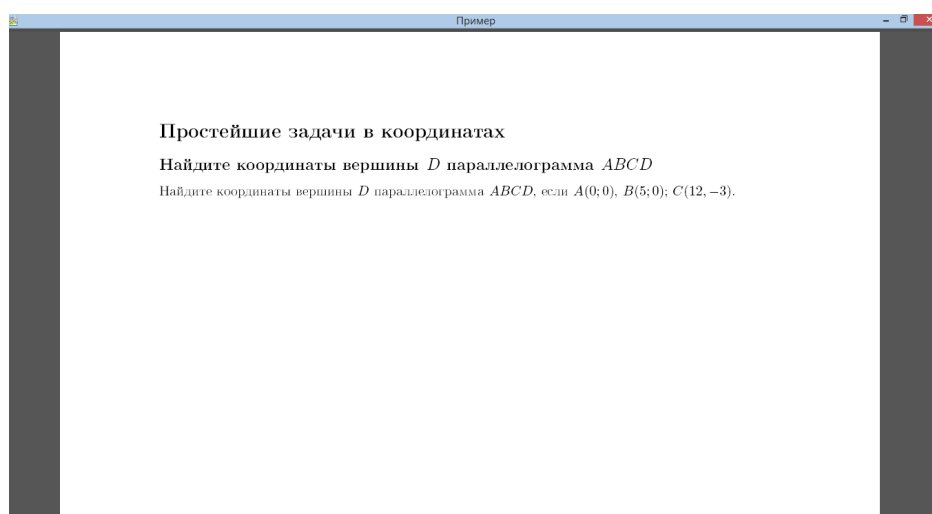


Рисунок 5 – Пример задачи

2.2 Найти координаты вершины D

Пример

Точка	x	y
A	0	0
B	5	1
C	12	-4

Найти координаты D

Рисунок 6 – Ввод данных

Пример

Задача. Найдите координаты вершины D параллелограмма $ABCD$, если $A(0; 0)$, $B(5; 1)$; $C(12, -4)$.

Дано:
 $ABCD$ – параллелограмм, $A(0; 0)$, $B(5; 1)$, $C(12, -4)$.

Найти: $D(x; y)$.

Решение. $\vec{AC} = \vec{AD} + \vec{AB} \Rightarrow \vec{AD} = \vec{AC} - \vec{AB}$ – по свойству параллелограмма. $D(7; -4)$, т.к. $x = x_C - x_B = 7$; $y = y_C = -4$. ■

Рисунок 7 – Алгоритм решения задачи

2.5 Совместное применение объектно-ориентированного языка программирования C# и компилятора открытого дистрибутива TeX для платформы Windows – MikTeX

Основная идея работы приложения состоит в том чтобы, осуществить возможность динамической генерации алгоритмов решения задач и их вывод. Что бы реализовать такой подход, мы предлагаем использовать компилятор \LaTeX – MikTeX. В данной ситуации задача состоит в следующем: вызвать компилятор LATEX при помощи Windows API, передать ему путь к файлу с алгоритмом решения задачи, оформленным в LATEX и получить откомпилированный файл в PDF формате. Далее вывести содержимое PDF файла на экран.

ООП C# позволяет взаимодействовать с другими приложениями в Windows[13]. Один из способов – это создать объект класса Process, для запуска локальных процессов в операционной системе. Для компиляции PDF файлов мы использовали именно этот способ.

Что открыть PDF файл в режиме чтения мы использовали возможности Adobe Reader, в частности был подключен COM - объект Adobe Acrobat 7.0 Browser ControlLibrary 1.0, позволяющий открывать PDF файлы прямо в форме нашего приложения.

2.6 Чтение PDF файлов в приложении

Использование COM – объекта Adobe Acrobat 7.0 Browser ControlLibrary 1.0 заключается в следующем: На форму помещается элемент AxAcroPDFLib[9], далее в коде указываем файл который надо открыть. Пример, кода открытия файла отражен в листинге [1]. В рамках приложения был разработан универсальный способ открытия файлов по средствам одной формы. При решении различных задач, пользователь имеет возможность просмотреть пример задания и алгоритм его решения, обращаясь к одной и той же форме, в которую передан путь к файлу PDF. В листинге [1] путь передан в строковую переменную path_example.

В листинге [2] можно увидеть, что форме «Пример» для открытия файла, добавлено новое свойство Path_example. Свойство Path_example за-

ключается в том что, это строка, в которую из каждой формы с задачей передается путь к нужному файлу PDF. Пример передачи пути файла в листинге [3]. Здесь для того что бы указать путь к файлу, создается ссылка на форму «Пример», а затем изменяется свойство Path_example.

2.7 Генерирование исходного кода, для компиляции PDF документов

Алгоритмы решения задач, оформляются следующим образом:

- а) условие задачи;
- б) дано;
- в) найти;
- г) решение.

Например:

Задача. Найти попарно коллинеарные векторы если: $\vec{a}\{3; 2\}; \vec{b}\{1; 2\}; \vec{c}\{5; 6\}$.

Дано:

$\vec{a}\{3; 2\}; \vec{b}\{1; 2\}; \vec{c}\{5; 6\}$.

Найти:

$\vec{a}||\vec{b} - ?$

$\vec{a}||\vec{c} - ?$

$\vec{c}||\vec{b} - ?$

Решение.

$\vec{a}||\vec{b} \Rightarrow \frac{x_a}{x_b} = \frac{y_a}{y_b} = k$ проверим: $\frac{3}{2} \neq \frac{1}{2} \Rightarrow \vec{a}, \vec{b}$ -не коллинеарны.

$\vec{a}||\vec{c} \Rightarrow \frac{x_a}{x_c} = \frac{y_a}{y_c} = k$ проверим: $\frac{3}{2} \neq \frac{5}{6} \Rightarrow \vec{a}, \vec{b}$ -не коллинеарны.

$\vec{c}||\vec{b} \Rightarrow \frac{x_c}{x_b} = \frac{y_c}{y_b} = k$ проверим: $\frac{5}{6} \neq \frac{1}{2} \Rightarrow \vec{a}, \vec{b}$ -не коллинеарны. ■

Один из приемов в работе приложения, состоит в том, что программа динамически формирует исходный код для LaTeX компилятора. Что бы ускорить процесс разработки, а так же добиться наибольшей комфортности, мы создали класс[11] «GeneratorTexText». Цель создания класса заключается в том, что бы избавиться от постоянного повторения больших участков кода, но при этом их использовать. Класс представлен в листинге [4].

Как известно любой документ LaTeX начинается с преамбулы[10]. В классе «GeneratorTexText» представлен метод[12] preambleWrite(). Этот метод в заданный файл записывает преамбулу документа. Из листинга [4]

видно, что для записи файлов в приложении используются потоки. При чем, в методе `preambleWrite()` работа с файлом производится в режиме `FileCreate`, что означает, что если файл существует, то он будет удален и на его место будет записан новый. А если не существует то файл будет создан. Этот режим используется один раз при записи преамбулы файла. После того как запись преамбулы завершилась, в файл добавляется LaTeX команда `\begin{document}`.

Запись алгоритма решения любой задачи разделена на несколько методов:

- `taskWrite(string strTask)` – запись условия задачи;
- `givenWrite(string strGiven)` – запись «Дано» задачи;
- `searchWrite(string strSearch)` – запись «Найти» задачи;
- `solutionWrite(String strSolution)` – запись решения задачи.

Все эти методы построены по одинаковой логике. Серьезное различие этих методов с методом `preambleWrite()`, заключается в то что, здесь при работе с файлом используется режим `FileMode.Append`. То есть файл уже не создается, а в уже существующий файл дописываются необходимые данные. Что бы обеспечить гибкость, методы принимают строковые параметры, в которых содержатся сформированные строчки LaTeX текста.

После вызовов выше описанных методов, вызывается метод `EndDocumentWrite()`. Суть этого метода в том, что бы завершить описание документа LaTeX, дописав туда команду `\end{document}`.

Теперь когда задача оформлена в LaTeX, ее можно компилировать в PDF файл и выводить на экран пользователя. Для компиляции PDF разработан метод `compilePDF()`. В указанном методе используется класс `ProcessStartInfo` и компонент `Process`.

Класс `ProcessStartInfo` позволяет точнее управлять запущенным процессом. В любом случае, нужно, по меньшей мере, задать свойство `FileName`, сделав это вручную или с помощью конструктора. Имя файла — это имя любого приложения или документа. Здесь документ определяется как файл любого типа, с которым связано активное действие или действие по умолчанию. [7]

Компонент `Process` является полезным инструментом для запуска, остановки, контролирования и наблюдения за приложением. Используя компонент `Process`, можно получить список выполняющихся процессов или запустить новый процесс. Компонент `Process` используется для доступа к системным процессам. После инициализации компонента `Process` его можно использовать для получения информации о выполняющемся процессе. Такая информация включает набор потоков, загруженные модули (файлы с расширением `DLL` и `EXE`) и информацию о производительности, например количество памяти, используемой процессом. [8]

Создаем объект класса `ProcessStartInfo` – `startInfo`, вызываем компилятор `pdflatex`, в качестве аргумента передаем имя записанного `LaTeX` документа. Что бы пользователь не видел процесс компиляции `PDF` файла свойству `WindowStyle` присваиваем значение `ProcessWindowStyle.Hidden`. То есть окно командной строки скрыто.

Далее запускаем компонент `Process` и передаем в него в качестве параметра объект класса `ProcessStartInfo` – `startInfo`. Метод компонента `Process` – `WaitForExit()`, позволяет приостановить выполнение программы пока не завершится запущенный процесс. Полученный `PDF` файл отправляем в форму «Пример», отображаем форму.

Для того что бы использовать класс `GeneratorTexText`, был специально объявлен конструктор класса и метод `Finish()`. Конструктор класса приведен в листинге [4]. Цели его задания следующие:

- а) указать имя и путь создаваемого файла;
- б) запустить метод `preambleWrite()` для записи преамбулы в файл.

Метод `Finish` (`string task`, `string given`, `string search`, `string solution`) принимает четыре строковых параметров. В строке `task` содержится условие задачи. В строке `given` содержится «Дано», в строке `search` – «Найти», в `solution` – решение задачи. Смысл метода `Finish` () в том что, здесь в нужном порядке вызываются методы: `taskWrite(task)`; `givenWrite(given)`; `this.searchWrite(search)`; `this.solutionWrite(solution)`; `compilePDF()`. Такой вызов позволяет упростить процесс компиляции `PDF` файла. Способ применения класса «`GeneratorTexText`» отражен в листинге [5]

Listing 1 — Код открытия файла `PDF`

```
1 axAcroPDF1.LoadFile(path_example);
```

Listing 2 — Объявление свойства Path_example

```
1 string path_example = "";
2
3 public string Path_example
4 {
5     get { return path_example; }
6     set { path_example = value; }
7 }
```

Listing 3 — Пример передачи пути файла

```
1 fViewExample fView = new fViewExample();
2 fView.Path_example = @"Examples\1\7.pdf";
3 fView.Show();
```

Listing 4 — Класс – GeneratorTexText

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Diagnostics;
6 using System.IO;
7 using System.Windows.Forms;
8 namespace Coordinate_method
9 {
10     class GeneratorTexText
11     {
12         public GeneratorTexText() {
13             Application.UseWaitCursor=true;
14             fileName = @"makePDF\1.tex";
15             preambleWrite();
16         }
17
18         string fileName;
19
20         Encoding enc = Encoding.GetEncoding(1251);
21
22         private void preambleWrite()
```

```

23      {
24          String str = "";
25          using (FileStream fs = File.Create(fileName))
26              {// Зануль нпреамбулы
27                  using (StreamWriter sw = new StreamWriter(fs, enc))
28                      {
29                          str = "\\documentclass[14pt,a4paper]{extarticle}";
30                          sw.WriteLine(str);
31                          str = "\\usepackage{amsmath,amsfonts,amssymb}";
32                          sw.WriteLine(str);
33                          str = "\\usepackage[mathscr]{eucal}";
34                          sw.WriteLine(str);
35                          str = "\\usepackage{graphicx}";
36                          sw.WriteLine(str);
37                          str = "\\usepackage[russian]{babel}";
38                          sw.WriteLine(str);
39                          str = "\\usepackage[cp1251]{inputenc}";
40                          sw.WriteLine(str);
41                          str = "\\usepackage[T2A]{fontenc}";
42                          sw.WriteLine(str);
43                          str = "\\newenvironment{proof}[1][Доказательство]
{\\noindent\\textbf{#1. \\ }}{\\ \\rule{0.5em}{0.5em}}";
44
45                          str = "\\newenvironment{solution}[1][Решение]
{\\noindent\\textbf{#1. \\ }}{\\ \\rule{0.5em}{0.5em}}";
46                          sw.WriteLine(str);
47                          str = "\\usepackage{geometry}";
48                          sw.WriteLine(str);
49                          str = "\\geometry{left=3cm}";
50                          sw.WriteLine(str);
51                          str = "\\geometry{right=1cm}";
52                          sw.WriteLine(str);
53                          str = "\\geometry{top=2cm}";
54                          sw.WriteLine(str);
55                          str = "\\geometry{bottom=2cm}";
56                          sw.WriteLine(str);
57                          str = "\\renewcommand{\\baselinestretch}{1.3}";
58                          sw.WriteLine(str);
59                          str = "\\tolerance=500";
60                          sw.WriteLine(str);

```

```

61         str = "\\begin{document}";
62         sw.WriteLine(str);
63
64     }
65 }
66 }
67 public void searchWrite(string strSearch) {
68     String str;
69     using (System.IO.FileStream fs =
new System.IO.FileStream(fileName, FileMode.Append))
70     {
71         using (StreamWriter sw = new StreamWriter(fs, enc))
72         {
73             str = "\\noindent{\\ "+"bf_Найти:}";
74             sw.WriteLine(str);
75             sw.WriteLine(strSearch);
76             str = "\\\\";
77             sw.WriteLine(str);
78         }
79     }
80
81 }
82 public void givenWrite(string strGiven) {
83     String str;
84     using (System.IO.FileStream fs =
new System.IO.FileStream(fileName, FileMode.Append))
85     {
86         using (StreamWriter sw = new StreamWriter(fs, enc))
87         {
88             str = "\\ "+"noindent{\\ "+"bf_Дано:}\\\\";
89             sw.WriteLine(str);
90             sw.WriteLine(strGiven);
91             str = "\\\\";
92             sw.WriteLine(str);
93         }
94     }
95 }
96 public void taskWrite(string strTask)
97 {
98     String str;

```

```

99         using (System.IO.FileStream fs =
new System.IO.FileStream(fileName, FileMode.Append))
100     {
101         using (StreamWriter sw = new StreamWriter(fs, enc))
102     {
103         str = "\\noindent{\\bf_Задача.}";
104         sw.WriteLine(str);
105         str = "{\\it";
106         sw.WriteLine(str);
107         sw.WriteLine(strTask);
108         str = "}\\\\\\\\";
109         sw.WriteLine(str);
110     }
111 }
112 }
113 }
114 public void solutionWrite(String strSolution)
115 {
116     using (System.IO.FileStream fs =
new System.IO.FileStream(fileName, FileMode.Append))
117     {
118         String str;
119         using (StreamWriter sw = new StreamWriter(fs, enc))
120     {
121         str = "\\begin{solution}";
122         sw.WriteLine(str);
123         sw.WriteLine(strSolution);
124         str = "\\end{solution}";
125         sw.WriteLine(str);
126     }
127 }
128 }
129 private void EndDocumentWrite(){
130     String str = "";
131     using (System.IO.FileStream fs =
new System.IO.FileStream(fileName, FileMode.Append))
132     {
133         using (StreamWriter sw = new StreamWriter(fs, enc))
134     {
135         str = "\\end{document}";

```

```

136             sw.WriteLine(str);
137         }
138
139     }
140
141 }
142     public void compilePDF() {
143         EndDocumentWrite();
144         ProcessStartInfo startInfo =
145 new ProcessStartInfo("pdflatex");
146         startInfo.WindowStyle = ProcessWindowStyle.Hidden;
147         startInfo.Arguments = fileName;
148         Process process = Process.Start(startInfo);
149         process.WaitForExit();
150         process.Close();
151         fViewExample fView = new fViewExample();
152         fView.Path_example = @"1.pdf";
153         fView.Show();
154         Application.UseWaitCursor = false;
155     }
156     public void Finish
157 (string task, string given, string search, string solution) {
158         this.taskWrite(task);
159         this.givenWrite(given);
160         this.searchWrite(search);
161         this.solutionWrite(solution);
162         this.compilePDF();
163     }
164 }
165 }

```

Listing 5 — Способ применения класса «GeneratorTexText»

```

1     string solution;
2     string task;
3     string given;
4     string search;
5     string str;
6     if (y < 0)
7     {
8         str = "_";

```



```

9          }
10         else
11         {
12             str = "+";
13         }
14         task = "Найдите координаты вектора  $\vec{a}$  =
" + x.ToString() + " $\vec{i}$ " + str + y.ToString() + " $\vec{j}$ " + "$.";
15         given = "$ $\vec{a}$  = " + x.ToString() + " $\vec{i}$ " + str +
y.ToString() + " $\vec{j}$ " + "$";
16         search = "$ $\vec{a}$   $\{x; y\}$ " + "$";
17         solution = "Так как  $\vec{a}$  =
" + x.ToString() + " $\vec{i}$ " + str + y.ToString() + " $\vec{j}$ ", то  $x$  =
" + x.ToString() + ",  $y$  = " + y.ToString();
18         solution += " $\rightarrow$ 
 $\vec{a}$   $\{$ " + x.ToString() + "; " + y.ToString() + " $\}$ " + "$";
19
20         GeneratorTexText generator = new GeneratorTexText();
21         generator.Finish(task, given, search, solution);

```

ЗАКЛЮЧЕНИЕ

Поставленная цель – выполнена. Итогом работы – является разработанный электронно - методический комплекс «Метод координат». В приложении реализован подход, позволяющий грамотно отображать алгоритмы решения геометрических задач. На основе разработанного класса «GeneratorTexText», можно выводить алгоритмы решения задач из курса геометрии. Приложение работает с известным форматом документов PDF.

Что бы оценить эффективность разработанного комплекса, необходимо опробовать данное приложение в школе. Предложить учащимся и учителям работу с приложением, как один из дополнительных способов решения задач и самоконтроля.

Для развития приложения имеет смысл как можно больше реализовать различных алгоритмов решения задач не только из курса геометрии, но и из курса алгебры, физики, химии. Развивать приложение и его идею целесообразно уже после получения конкретных результатов и отзывов в ходе учебного процесса.

На основе заложенной идеи задания всего курса естественно - научных предметов – могут быть представлены в такой форме. А система разметки LaTeX позволит оформить задания не только для естественно - научного курса, но и задания для гуманитарного цикла предметов.

Язык программирования C# показал свою эффективность в области работы как с COM - объектами, так и с внешними программами Windows. Но при этом установлено, что C# зависит от библиотек .NET Framework. Рассматриваемый язык и его компилятор не позволяют создавать комплексы для других операционных систем таких как Linux и iOS.

Система LaTeX показала свою универсальность в работе с сложными математическими текстами. Положительная сторона этой системы разметки состоит в том, что дистрибутивы компиляторов и библиотек классов находятся в свободном доступе. Документация по LaTeX открыта. Учебные пособия доступны для чтения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Геометрия. 7 — 9 классы: учеб. для общеобразоват. уч-ГЗ6 реждений [Л. С, Атанасян, В. Ф. Бутузов, С. Б. Кадомцев и др.]. — 20 -е изд. — М.: Просвещение, 2010. — 384 с..
2. Изучение геометрии в 7 — 9 классах. Пособие для ИЗ9 учителей. / [Л.С.Атанасян, В. Ф.Бутузов,Ю. А. Глазков и др.] — 7-е изд — М.:Просвещение. 2009 — 225с.
3. Введение в язык C# и .NET Framework // Microsoft Developer Network (Рус.) — URL: <http://msdn.microsoft.com/ru-ru/library/z1zx9t92.aspx> [9 июня 2014]
4. Visual C# // Microsoft Developer Network (Рус.) — URL: <http://msdn.microsoft.com/ru-ru/library/kx37x362.aspx>
5. Котельников И. А., Чеботаев П. З. К26 Л^AT_EX по-русски.— 3-е издание, перераб. и доп.— Новосибирск: Сибирский хронограф, 2004. — 496 с.
6. MiKTeX // Википедия — Свободная энциклопедия (Рус.) — URL: <http://ru.wikipedia.org/wiki/MiKTeX>
7. ProcessStartInfo—класс // Microsoft Visual Studio (Рус.) —URL: [http://msdn.microsoft.com/ru-ru/library/vstudio/system.diagnostics.processstartinfo\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/vstudio/system.diagnostics.processstartinfo(v=vs.100).aspx)
8. Process - класс // Microsoft Visual Studio (Рус.) — URL: [http://msdn.microsoft.com/ru-ru/library/vstudio/system.diagnostics.process\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/vstudio/system.diagnostics.process(v=vs.100).aspx)
9. PDF component in WinForms using C# // CODE PROJECT For those who code (Англ.) — URL: <http://www.codeproject.com/Questions/290805/PDF-component-in-WinForms-using-Csharp>
10. Г.Грэтцер. Первые шаги в Л^AT_EX. — М.: Мир, 2000. — 172 с.
11. Фленов М. Е. Ф69 Библия C#. — 2-е изд., перераб и доп — СПб.: БХВ-Петербург, 2011. — 560с..
12. Шилдт, Герберт. Ш57 C# 4.0: полное руководство.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2011 — 1056с..
13. Уотсон, Карли, Нейгел, Кристиан, Педерсен, Якоб Хаммер, Рид, Джон

Д., Скиннер, Морган. У65 Visual C# 2010: полный курс.: Пер. с англ.
— М.: ООО «И.Д. Вильямс», 2011. — 960с..