

ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. НИЗАМИ
ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

“Разрешить к защите”

Декан факультета

_____ Г. Ф. Джаббаров

“ ____ ” _____ 2020 год

Студент направления

“5110100 – Методика преподавания математики”

ЧЕПУХАЛИН Сергей Анатольевич

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

**“МЕТОДИКА ПРИМЕНЕНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
В ИЗУЧЕНИИ МНОГОЧЛЕНОВ В ОБЩЕОБРАЗОВАТЕЛЬНЫХ ШКОЛАХ”**

Выполнил: _____ С. А. Чепухалин

Научный руководитель:

Доцент кафедры “Математика и
методика ее преподавания”,

к. ф.-м. н. _____ Ш. Н. Исмаилов

Рецензенты:

Старший преподаватель кафедры
“Математика в гуманитарных
направлениях”,

_____ А. Р. Латыпова

Учитель математики

общеобразовательной школы №145

_____ Н. А. Гиенко

“Допустить к защите”

Заведующий кафедрой “Математика
и методика ее преподавания”,

к. п. н. _____ А. А. Акмалов

“ ____ ” _____ 2020 г.

Ташкент 2020 г.

ВВЕДЕНИЕ.....	3
ГЛАВА 1. MICROSOFT VISUAL STUDIO КАК ПЕДАГОГИЧЕСКОЕ ПРОГРАММНОЕ СРЕДСТВО.....	6
1.1. Понятие педагогического программного средства (ППС). Типы ППС. Требования к разработке ППС.	6
1.2. Информационные технологии в образовании	8
1.3. Описание среды разработки MICROSOFT VISUAL STUDIO	12
1.4. Язык программирования C#.....	16
1.5. Интерфейс MICROSOFT VISUAL STUDIO	16
Рисунок 15 - Окно запущенной программы	31
Рисунок 16 - Результат нажатия кнопки.....	31
Рисунок 20 - До изменения свойств элементов	33
Рисунок 21 - После изменения свойств элементов	33
1.6. Что лучше DELPHI или VISUAL STUDIO?	34
ГЛАВА 2. ТЕОРИЯ ДЕЛИМОСТИ МНОГОЧЛЕНОВ	36
2.1. Многочлены в школьной программе.	36
2.2. Общее понятие.	37
2.3. Действия с многочленами.....	39
2.4. делимость многочленов	42
2.5. Алгоритм Евклида.	43
2.6. Применение теории делимости.....	44
ПРИЛОЖЕНИЯ	46
ВЫВОДЫ.....	57
СПИСОК ЛИТЕРАТУРЫ.....	59

ВВЕДЕНИЕ

«В целях последовательного продолжения и выведения на новый, современный уровень начатой нами работы по развитию сферы науки и просвещения, воспитания нашей молодежи личностями, обладающими глубокими знаниями, высокой культурой и духовностью, формирования конкурентоспособной экономики. Предлагаю объявить 2020 год в нашей стране Годом развития науки, просвещения и цифровой экономики» [1], что еще раз говорит о том, что цифровые программные средства все активнее входят в нашу жизнь не исключением является и сфера образование. В настоящее время существует великое множество всевозможных программных средств облегчающих работу, как преподавателя, так и учащихся при выполнении стандартных рутинных работ. Однако многие из этих программ перегружены излишним функционалом, который не только выходит за рамки школьной программы, но порой и не выполняет возлагаемых на него надежд.

Как говорит президент республики Узбекистан Ш. М. Мирзиёев: «Математика - основа всех точных наук. Ребенок, хорошо знающий математику, будет расти благоразумным, сможет успешно работать в любой сфере».

В государственном образовательном стандарте присутствуют компетенции, отображающие значение математических и информационных знаний, умений и навыков в образовании и повседневной жизни:

Информационная компетенция — умение находить, отбирать, преобразовывать, сохранять, передавать необходимые сведения из медиасредств, соблюдая правила информационной безопасности, включая формирование медиакультуры.

Компетенция математической грамотности, осведомлённости о достижениях науки и техники, умения пользоваться ими — умение планировать личную, семейную жизнь и профессиональную деятельность на основе точного расчета; умение читать различные формулы, модели, графики, чертежи и диаграммы, пользоваться ими в повседневной жизни; осведомленность о достижениях науки и техники, облегчающих труд человека, повышающих его эффективность и действенность, умение ими пользоваться.

Непосредственно в базовый уровень (A2) входит: умение применять готовые компьютерные программы для несложных вычислений при решении уравнений и неравенств, используя изучаемые формулы и представлять плоские геометрические фигуры и тела в пространстве.

В углублённый базовый уровень (A2+): умение применять при решении математических задач информационно - коммуникативные средства, интернет сайты, социальные сети, блоги, форумы, символика вычислений. [2]

Актуальность: изучение многочленов является обширной частью теоретической и практической алгебры школьного курса, применение информационных технологий является неотъемлемой частью современных методов преподавания.

Объект работы: процесс обучение математике в средней общеобразовательной школе.

Предмет работы: дидактические возможности изложения темы «Многочлены» в школьном курсе математики.

Цель работы: педагогический анализ средств ИКТ (КСМ) с применением MS Visual Studio, разработка методического указания для учителя математики.

Ожидаемые результаты: Будет разработаны и проанализированы средства компьютерного сопровождения изучения многочленов на основе MS Visual Studio.

Актуальность темы. Широкое применение в образовании современных программ, которые характеризуются разносторонностью и простотой обращения, а также позволяют использовать их возможности для решения самого широкого круга образовательных задач.

Вторая половина XX века стала периодом перехода к информационным обществам. Развитие любой страны зависит не только от природных и физических, но, прежде всего от информационных ресурсов.

Из-за огромного количества информации появилось множество проблем, важнейшей из которых является проблема обучения. Особый интерес для нас представляют вопросы, связанные с применением информационных технологий в образовании, так как в наше время обычными методами обучения удовлетворить индивидуальные запросы учащихся стало труднее.

Одним из самых доступных средств в данном случае является использование компьютера. Средства информационных и компьютерных технологий позволяют автоматизировать, а тем самым упростить ту сложную процедуру, которую используют и учителя при создании методических пособий. Тем самым, представление различного рода электронных учебников, методических пособий на компьютере имеет ряд важных преимуществ. Во-первых, это автоматизация, как самого процесса создания таковых, так и хранения данных в любой необходимой форме. Во-вторых, это работа с практически неограниченным объёмом данных. В-третьих, в создании таковых участвуют ученики школы, приобретая для себя немало новых навыков и принося этим самым пользу не только себе, но и школе.

Создание учебников средствами компьютерных технологий схоже с изданием учебных пособий нового поколения, отвечающих потребностям личности обучаемого. Учебные издания нового поколения призваны обеспечить единство учебного процесса и современных, инновационных научных исследований, т. е. целесообразность использования новых информационных технологий в учебном процессе. Эффект от применения средств компьютерной техники в обучении может быть достигнут лишь тогда, когда специалист предметной области не ограничивается в средствах представления информации, коммуникаций и работы с базами данных и знаний.

При изучении информационных технологий и услуг сети Internet у учеников повышается кругозор знаний, также они получают практические навыки работы с персональным компьютером и сетью Internet, что обязательно пригодится им в дальнейшей жизни.

Сегодняшние ученики школ достаточно активно интересуются информационными и Internet технологиями, сеть Internet в данном случае выступает отличным средством для развития их творческих способностей, а также способствует:

- повышению интереса к обучению;
- развитию самостоятельности;

– развитию ответственности и целеустремлённости.

Цель работы: обоснование применения компьютерных программ в образовании как одного из перспективных средств, рассмотрение программных средств, используемых в учебных заведениях, а также разработка практических заданий в приложениях MS Visual Studio.

Задачи работы:

1. Исследовать существующие подходы в обучении школьников использованию различных компьютерных программ;
2. Разработать практические задания для изучения некоторых программ среды MS Visual Studio;
3. Рассмотреть программу обучения учащихся общеобразовательных школ теории многочленов;
4. Оценить актуальность применения программных педагогических средств при изучении темы «Многочлены»;
5. Внести предложение по разработке программы, производящей автоматизацию действий по решению задач на многочлены.

Объект работы. Образовательный процесс с использованием программных средств учебного назначения. Автоматизация работы в среде MS Visual Studio.

Предмет работы. Программные средства, широко применяемые в образовательном процессе.

Структура работы. Работа включает в себя введение, две главы, заключение, список литературы и приложение.

В первой главе «Microsoft Visual Studio как педагогическое программное средство» рассмотрены понятие «Педагогическое программное средство», типы ППС и требования к ее разработки, приводятся мнения отечественных и зарубежных специалистов-педагогов о достоинствах и недостатках использования ППС в различных сферах образования, описание среды разработки Microsoft Visual Studio, язык программирования C#, интерфейс Microsoft Visual Studio, произведено сравнение среды разработки Microsoft Visual Studio со средой разработки Delphi.

Во второй главе «Теория делимости многочленов» приведены основные понятия «одночлен», «многочлен», действие над многочленами – сложение, вычитание, умножение, деление; применение теории делимости многочленов, и их реализация в программе, разработанной в среде MS Visual Studio.

В приложении приведены разработки поурочных занятий по темам «Сложение и вычитание многочленов», «Умножение одночлена на многочлен», «Умножение многочлена на многочлен» для 7 класса общеобразовательной школы, и в заключении даны основные выводы по проведенной работе.

Список литературы содержит 14 источников, среди которых «Государственный образовательный стандарт общего среднего образования», «Указ Президента РУз № УП-5712 "Об утверждении Концепции развития системы народного образования Республики Узбекистан до 2030 года"», «Алгебра Учебник для 7 классов школ общего среднего образования „O‘qituvchi“ Ташкент 2017» и многие другие.

ГЛАВА 1. MICROSOFT VISUAL STUDIO КАК ПЕДАГОГИЧЕСКОЕ ПРОГРАММНОЕ СРЕДСТВО

1.1. Понятие педагогического программного средства (ППС). Типы ППС. Требования к разработке ППС.

Программные средства учебного назначения, с одной стороны, способствуют сглаживанию внутренних противоречий, присущих процессу обучения, с другой стороны, способствуют возникновению новых. Так, противоречие при классно-урочной системе обучения между коллективной формой обучения и индивидуальным темпом усвоения знаний можно решить за счет использования обучающих программ (тренировочные программы). Эти программы вводит в учебный процесс учитель, и именно он является организатором процесса обучения.

Под **программными средствами** учебного назначения (ПС) будем понимать программные продукты, предназначенные для решения отдельных учебно-воспитательных задач.

Под **обучающей программой со встроенной технологией обучения** будем понимать программу учебного назначения, способную осуществлять некоторые функции преподавателя, а именно:

- предъявление учебного материала;
- осуществление контроля за результатом усвоения знаний;
- формирование навыков учебной деятельности.

О преимуществах и недостатках применения ППС в различных дисциплинах высказывалось множество педагогов многих стран мира, в том числе и Узбекистана:

Например, Гаюбова, К. А. в статье «Использование новых педагогических технологий в преподавании узбекского языка» писала: «При анализе целесообразности использования компьютера в учебном процессе можно выявить следующие его дидактические возможности: расширение самостоятельной творческой деятельности учащихся, особенно при исследовании и систематизации явлений узбекского языка; привитие навыков самоконтроля, самостоятельного исправления ошибок; развитие познавательных способностей; быстрое освоение материала; интегрированное обучение предмету; развитие мотивации учащихся; экономия времени...»

К сожалению, при использовании мультимедийных технологий на занятии возникает ряд проблем: недостаточное обеспечение учебного процесса дифференцированными программами; отсутствие программного обеспечения в условиях поликультурного региона». [5]

Байбулатова Г. К. в своей работе «Новые государственные стандарты в Узбекистане» указала: «Сегодня дети уже в первый класс приходят с навыками работы не только на компьютере, но и на прочих мобильных гаджетах, так что учить их элементарным азам не нужно. Поэтому стандарты и учебные планы по предмету «Информатика и информационные технологии» сегодня ориентированы на то, чтобы обучить детей более широкому и активному применению этих технологий в повседневной жизни...»

Это сделано с учетом требований времени и для формирования у детей 1–4-х классов начальных знаний по информационным технологиям.

А что касается преподавания математики, то эти изменения окажут только

благоприятное влияние на его качество, поскольку урок станет более интересным и увлекательным. Ведь дети уже в этом возрасте начнут получать информацию о том, как работают всевозможные гаджеты, которыми они пользуются ежедневно.

К тому же программирование в этот период будет представлено в основном задачками на логику. Главная цель этого нововведения – развитие логического мышления у учащихся.»[6]

Существует большое количество различных классификаций и типологий программ учебного назначения (Н. В. Апатова, Л. Х. Зайнутдинова, А. С. Лесневский, И. В. Роберт, Т. А. Сергеева, Н. В. Софронова, и др.). Например, в институте средств обучения РАО выделили несколько классификационных критериев типологии педагогических программных средств:

По предметному содержанию;

По функции: диагностические, контролирующие, обучающие (демонстрационные, справочно-информационные, формирующие, тренажерные);

По степени активности учащихся, которая определяется структурой и характером деятельности — программы, рассчитанные на минимальную степень активности (демонстрационные), на максимальную степень (конструирующие программы);

По целевой группе пользователя — инструментальные программные средства для учителя: для создания новых обучающих программ, применение имеющихся программных средств в процессе подготовки к уроку («электронный конспект», «электронный журнал»); программные средства для учащихся (общего назначения и специализированные). Примером таких программ являются графические и текстовые редакторы, базы данных или электронные таблицы;

По уровню коммуникативности можно выделить следующие типы программ: предметно-ориентированные обучающие программы с разными степенями интерактивности и коммуникативно-ориентированные (локальные сети).

В работах И. В. Роберт дана классификация программных средств учебного назначения, основанная на их методическом назначении: обучающие, контролирующие, тренажеры, информационно-справочные, информационно-поисковые, имитационные, моделирующие, демонстрационные, игровые, досуговые.

Зайнутдинова предлагает различать 3 типа компьютерных обучающих программ:

-педагогические программные средства (ППС) (компьютерные учебные программы одноцелевого назначения: сервисные, контролирующие, тренажеры, моделирующие, демонстрационные и т. п);

-информационно-поисковые справочные программные системы (ИПСПС) (базы данных и базы знаний.);

-обучающие программные системы (ОПС) (представляют пользователю комплекс возможностей, в их число входят автоматизированные обучающие системы (АОС), электронные учебники (ЭУ); -экспертные обучающие системы (ЭОС); -интеллектуальные обучающие системы (ИОС)).

В. М. Монахов выделил следующие четыре направления использования программных средств в процессе обучения:

- экспертные системы,
- разветвленно-диалоговые обучающие системы,
- имитаторы эксперимента;
- обучающие среды.

Программные средства используются на уроках информатики, прежде всего, для формирования информационной культуры учащихся. Но понятие это весьма многогранно и включает ряд умений, необходимых для жизни в информационном обществе: умение работать с различного вида информацией, обрабатывать большие объемы информации, преобразовывать задачи для решения с помощью компьютера и т. д.

Функции использования программно-методических средств можно охарактеризовать следующим образом: с помощью графических редакторов можно развивать мышление и воображение школьников, обучать рисованию и черчению с помощью современных технологий; текстовые редакторы служат для подготовки отчетов и рефератов по предметам; с помощью электронных таблиц осуществляется проведение вычислительных экспериментов, формируются навыки моделирования; системы управления базами данных и разработки мультимедиа приложений развивают навыки исследовательской работы с большими объемами информации.

1.2. Информационные технологии в образовании

В XXI веке уже недостаточно иметь академические знания и критичность мышления, для достижения успеха нужно ещё иметь технические навыки. Поэтому уже в раннем возрасте дети стремятся получить знания в области информационных технологий, как наиболее перспективной области деятельности человека. Как следствие, в последнее время стали очень популярными различные компьютерные курсы и образовательные информационные системы. На сегодняшний день уже многие школы имеют современную компьютерную технику, использование которой возможно в обучении не только информатике, но и другим дисциплинам.

Любой ученик индивидуален, и осваивает новый учебный материал по-разному. Преподавателям приходится находить индивидуальный подход к каждому ученику. С использованием передовых информационных технологий, сети Internet, средств мультимедиа удовлетворить индивидуальные запросы учащихся стало намного проще.

Под средствами новых информационных технологий (СНИТ) будем понимать программно-аппаратные средства и устройства, функционирующие на базе микропроцессорной, вычислительной техники, а также современных средств и систем информационного обмена, обеспечивающие операции по сбору, продуцированию, накоплению, хранению, обработке, передаче информации.

Ускорение научно-технического прогресса, основанное на внедрении в производство гибких автоматизированных систем, микропроцессорных средств и устройств программного управления, роботов и обрабатывающих центров, поставило перед современной педагогической наукой важную задачу – воспитать и подготовить подрастающее поколение, способное активно включиться в качественно новый этап развития современного общества, связанный с информатизацией. Решение вышеназванной задачи – выполнение социального

заказа общества – коренным образом зависит как от технической оснащенности учебных заведений электронно-вычислительной техникой с соответствующим периферийным оборудованием, учебным, демонстрационным оборудованием, функционирующим на базе СНИТ, так и от готовности обучаемых к восприятию постоянно возрастающего потока информации, в том числе и учебной.

Повсеместное использование информационных ресурсов, являющихся продуктом интеллектуальной деятельности наиболее квалифицированной части трудоспособного населения общества, определяет необходимость подготовки в подрастающем поколении творчески активного резерва. По этой причине становится актуальной разработка определенных методических подходов к использованию СНИТ для реализации идей развивающего обучения, развития личности обучаемого. В частности, для развития творческого потенциала индивида, формирования у обучаемого умения осуществлять прогнозирование результатов своей деятельности, разрабатывать стратегию поиска путей и методов решения задач – как учебных, так и практических.

Не менее важна задача обеспечения психолого-педагогическими и методическими разработками, направленными на выявление оптимальных условий использования СНИТ в целях интенсификации учебного процесса, повышения его эффективности и качества.

Актуальность вышеперечисленного определяется не только социальным заказом, но и потребностями индивида к самоопределению и самовыражению в условиях современного общества этапа информатизации.

Особого внимания заслуживает описание уникальных возможностей СНИТ, реализация которых создает предпосылки для небывалой в истории педагогики интенсификации образовательного процесса, а также создания методик, ориентированных на развитие личности обучаемого. Перечислим эти возможности:

- незамедлительная обратная связь между пользователем и СНИТ;
- компьютерная визуализация учебной информации об объектах или закономерностях процессов, явлений, как реально протекающих, так и «виртуальных»;
- архивное хранение достаточно больших объемов информации с возможностью ее передачи, а также легкого доступа и обращения пользователя к центральному банку данных;
- автоматизация процессов вычислительной информационно-поисковой деятельности, а также обработки результатов учебного эксперимента с возможностью многократного повторения фрагмента или самого эксперимента;
- автоматизация процессов информационно-методического обеспечения, организационного управления учебной деятельностью и контроля за результатами усвоения.

Реализация вышеперечисленных возможностей СНИТ позволяет организовать такие виды деятельности как.

- регистрация, сбор, накопление, хранение, обработка информации об изучаемых объектах, явлениях, процессах, в том числе реально протекающих, и передача достаточно больших объемов информации, представленной в различных формах;

– интерактивный диалог – взаимодействие пользователя с программной (программно-аппаратной) системой, характеризующееся в отличие от диалогового, предполагающего обмен текстовыми командами (запросами) и ответами (приглашениями), реализацией более развитых средств ведения диалога (например, возможность задавать вопросы в произвольной форме, с использованием «ключевого» слова, в форме с ограниченным набором символов).

При этом обеспечивается возможность выбора вариантов содержания учебного материала, режима работы:

– управление отображением на экране моделей различных объектов, явлений, процессов, в том числе и реально протекающих;

– автоматизированный контроль (самоконтроль) результатов учебной деятельности, коррекция по результатам контроля, тренировка, тестирование.

Педагогические цели использования СНИТ.

Развитие личности обучаемого, подготовка индивида к комфортной жизни в условиях информационного общества:

– развитие мышления, (например, наглядно-действенного, наглядно-образного, интуитивного, творческого, теоретического видов мышления);

– эстетическое воспитание (например, за счет использования возможностей компьютерной графики, технологии Мультимедиа);

– развитие коммуникативных способностей;

– формирование умений принимать оптимальное решение или предлагать варианты решения в сложной ситуации (например, за счет использования компьютерных игр, ориентированных на оптимизацию деятельности по принятию решения);

– развитие умений осуществлять экспериментально-исследовательскую деятельность (например, за счет реализации возможностей компьютерного моделирования или использования оборудования, сопрягаемого с ЭВМ);

– формирование информационной культуры, умений осуществлять обработку информации (например, за счет использования интегрированных пользовательских пакетов, различных графических и музыкальных редакторов).

Реализация социального заказа, обусловленного информатизацией современного общества:

– подготовка специалистов в области информатики и вычислительной техники;

– подготовка пользователя средствами новых информационных технологий.

Интенсификация всех уровней учебно-воспитательного процесса:

– повышение эффективности и качества процесса обучения за счет реализации возможностей СНИТ;

– обеспечение побудительных мотивов (стимулов), обуславливающих активизацию познавательной деятельности (например, за счет компьютерной визуализации учебной информации, вкрапления игровых ситуаций, возможности управления, выбора режима учебной деятельности);

– углубление межпредметных связей за счет использования современных средств обработки информации, в том числе и аудиовизуальной, при решении задач различных предметных областей.

СНИТ могут быть использованы в качестве:

1. Средства обучения, совершенствующего процесс преподавания, повышающего его эффективность и качество. При этом обеспечивается:

- реализация возможностей программно-методического обеспечения современных ПЭВМ и др. в целях сообщения знаний, моделирования учебных ситуаций осуществления тренировки, контроля за результатами обучения;

- использование объектно-ориентированных программных средств или систем (например, системы подготовки текстов, электронных таблиц, баз данных) в целях формирования культуры учебной деятельности;

- реализация возможностей систем искусственного интеллекта в процессе применения обучающих интеллектуальных систем;

2. Инструмента познания окружающей действительности и самопознания;

3. Средства развития личности обучаемого;

4. Объекта изучения (например, в рамках освоения курса информатики);

5. Средства информационно-методического обеспечения и управления учебно-воспитательным процессом, учебными заведениями;

6. Средства коммуникаций (например, на базе асинхронной телекоммуникационной связи) в целях распространения передовых педагогических технологий;

7. Средства автоматизации процессов контроля, коррекции результатов учебной деятельности, компьютерного педагогического тестирования и психодиагностики;

8. Средства автоматизации процессов обработки результатов эксперимента (лабораторного, демонстрационного) а управления учебным оборудованием;

9. Средства организации интеллектуального досуга, развивающих игр.

Информационные технологии уже давно применяются не только в информатике, но и в изучение других предметов. Использование обучающих программ на уроках информатики повышает качество образовательного процесса. По информации компании Microsoft, за пять месяцев учащиеся школы Hacienda La Puente Unified School District в южной Калифорнии, используя компьютерную программу обучения чтению, по всем восьми показателям в среднем на 15 % превосходили других учеников, которые учились читать традиционным образом. При этом учащиеся, имеющие ограниченные познания в английском языке, получили вдвое более высокие оценки, чем их англоязычные одноклассники.

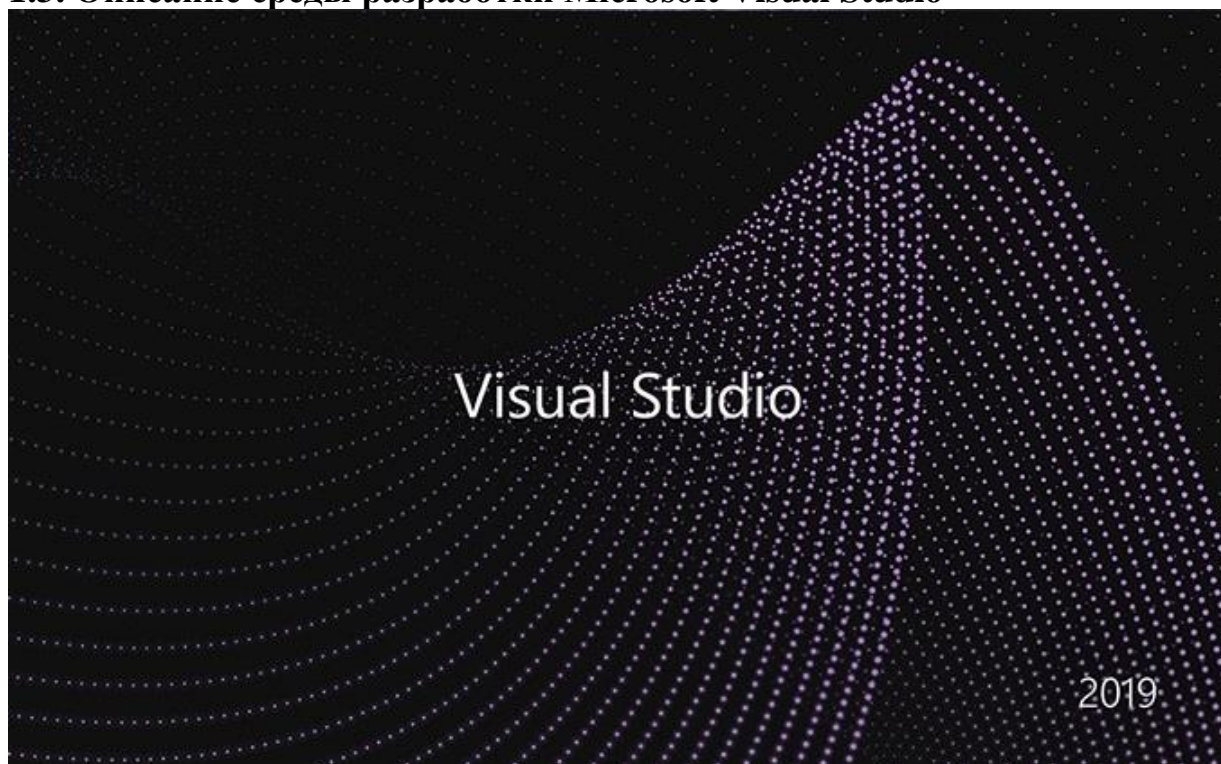
Используя информационные технологии, школы поощряют преподавателей на проведение экспериментов, внедряя групповое преподавание, увеличивая продолжительность занятий, реализуя эмпирическое обучение и другие педагогические модели. Подобные организационные изменения помогают раскрыть творческий потенциал учащихся.

Использование информационных технологий в школе предусматривает применение информационных технологий во всех сторонах деятельности школы. А именно:

- обучение;
- управление;
- методическая работа преподавателей;
- внеклассная работа.

Это означает, что компьютер должен использоваться не только как средство обучения, но также для решения учебных и производственных задач, как учащимися, так и преподавателями и администрацией школы.

1.3. Описание среды разработки Microsoft Visual Studio



Microsoft Visual Studio - это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом.

В комплект входят следующие основные компоненты:

1. Visual Basic. NET - для разработки приложений на Visual Basic;
2. Visual C++ - на традиционном языке C++;
3. Visual C# - на языке C# (Microsoft);
4. Visual F# - на F# (Microsoft Developer Division).

Функциональная структура среды включает в себя:

- редактор исходного кода, который включает множество дополнительных функций, как автодополнение IntelliSense, рефакторинг кода и т. д. ;
- отладчик кода;
- редактор форм, предназначенный для упрощённого конструирования графических интерфейсов;
- веб-редактор;
- дизайнер классов;
- дизайнер схем баз данных.

Visual Studio также позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (Subversion и VisualSourceSafe), добавление новых наборов инструментов (для редактирования и визуального проектирования кода на предметно-ориентированных языках

программирования или инструментов для прочих аспектов процесса разработки программного обеспечения).

Коммерческие версии в порядке возрастания цены: Visual Studio Professional, Visual Studio Premium и Visual Studio Ultimate.

Достоинства и недостатки

Интегрированная среда разработки (Integrated Development Environment - IDE) Visual Studio предлагает ряд высокоуровневых функциональных возможностей, которые выходят за рамки базового управления кодом.

Ниже перечислены основные преимущества IDE-среды Visual Studio.

Встроенный Web-сервер. Для обслуживания Web-приложения ASP. NET необходим Web-сервер, который будет ожидать Web-запросы и обрабатывать соответствующие страницы. Наличие в Visual Studio интегрированного Web-сервера позволяет запускать Web-сайт прямо из среды проектирования, а также повышает безопасность, исключая вероятность получения доступа к тестовому Web-сайту с какого-нибудь внешнего компьютера, поскольку тестовый сервер может принимать соединения только с локального компьютера.

Поддержка множества языков при разработке. Visual Studio позволяет писать код на своем языке или любых других предпочитаемых языках, используя все время один и тот же интерфейс (IDE). Более того, Visual Studio также еще позволяет создавать Web-страницы на разных языках, но помещать их все в одно и то же Web-приложение. Единственным ограничением является то, что в каждой Web-странице можно использовать только какой-то один язык (очевидно, что в противном случае проблем при компиляции было бы просто не избежать).

Меньше кода для написания. Для создания большинства приложений требуется приличное количество стандартного стереотипного кода, и Web-страницы ASP. NET тому не исключение. Например, добавление Web-элемента управления, присоединение обработчиков событий и корректировка форматирования требует установки в разметке страницы ряда деталей. В Visual Studio такие детали устанавливаются автоматически.

Интуитивный стиль кодирования. По умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам. Применяемые Visual Studio автоматически параметры форматирования можно даже настраивать, что очень удобно в случаях, когда разработчик предпочитает другой стиль размещения скобок (например, стиль K&R, при котором открывающая скобка размещается на той же строке, что и объявление, которому она предшествует).

Более высокая скорость разработки. Многие из функциональных возможностей Visual Studio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее. Удобные функции, вроде функции IntelliSense (которая умеет перехватывать ошибки и предлагать правильные варианты), функции поиска и замены (которая позволяет отыскивать ключевые слова как в одном файле, так и во всем проекте) и функции автоматического добавления и удаления комментариев

(которая может временно скрывать блоки кода), позволяют разработчику работать быстро и эффективно.

Возможности отладки. Предлагаемые в Visual Studio инструменты отладки являются наилучшим средством для отслеживания загадочных ошибок и диагностирования странного поведения. Разработчик может выполнять свой код по строке за раз, устанавливать интеллектуальные точки прерывания, при желании сохраняя их для использования в будущем, и в любое время просматривать текущую информацию из памяти.

Visual Studio также имеет и множество других функций: возможность управления проектом; встроенная функция управления исходным кодом; возможность рефакторизации кода; мощная модель расширяемости. Более того, в случае использования Visual Studio 2008 Team System разработчик получает расширенные возможности для модульного тестирования, совместной работы и управления версиями кода (что значительно больше того, что предлагается в более простых инструментах вроде Visual SourceSafe).

В качестве недостатка можно отметить невозможность отладчика (Microsoft Visual Studio Debugger) отслеживать в коде режима ядра. Отладка в Windows в режиме ядра в общем случае выполняется при использовании WinDbg, KD или SoftICE.

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в собственном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, . NET Framework, . NET Compact Framework и Microsoft Silverlight.

Visual Studio включает в себя редактор исходного кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов цикла разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

Интегрированная среда разработки программ Microsoft Visual Studio предназначена для создания программных приложений и позволяет:

- создавать приложения с использованием языков Visual Basic, Visual C#, Visual C++, Visual J#;
- создавать Windows- и Web-приложения, включая приложения для портативных устройств;

- создавать клиент-серверные приложения;
- создавать корпоративные Web-приложения масштаба предприятия.

Microsoft Visual Studio увеличивает производительность труда разработчиков при создании приложений. Включает, например, Visual Studio Team System — интегрированный и расширяемый набор инструментов для управления программными проектами на всех этапах разработки и тестирования, что обеспечивает взаимодействие и совместную работу коллектива разработчиков.

Microsoft Visual Studio содержит среду выполнения . *NET Framework*, которая состоит из двух частей: единой среды исполнения (Common Language Runtime, CLR) и библиотеки классов. Библиотека классов является всесторонней, объектно-ориентированной коллекцией типов, которую можно использовать для разработки приложений, начиная с традиционных приложений с командной строкой и с использованием графического пользовательского интерфейса и заканчивая приложениями, использующими Web-формы и XML Web-сервисы. Класс — это тип, описывающий устройство объектов, в то время как объект является конкретным представителем определенного класса. Таким образом, каждый объект является экземпляром определенного класса. Единая среда исполнения управляет кодом во время его выполнения.

Элементы графического интерфейса среды разработки Visual Studio, называемого сокращенно *IDE (Integrated Development Environment)* (меню, панели инструментов, диалоговые окна), характерны для среды Windows. Среда разработки Visual Studio является интегрированной, так как в ней можно выполнять различные действия при разработке программного продукта, такие как проектирование графического интерфейса приложения, редактирование программного кода, компиляция всех элементов приложения и сборка в выполняемый файл, а также пошаговая отладка приложения. Пользователь может изменять расположение и форму окон IDE, а также сворачивать их, чтобы сделать доступными и видимыми на экране необходимые элементы среды программирования. Главное окно Visual Studio после запуска программы содержит несколько основных объектов: главное меню и стандартная панель инструментов, окна Start Page (Начальная страница), Solution Explorer (Обозреватель решений), Toolbox (Инструментарий), Properties (Свойства), Object Browser (Просмотр объектов) и Dynamic Help (Динамическая справка). Диалоговое окно Start Page (Начальная страница) позволяет открывать недавно использовавшиеся проекты, осуществляет поиск примеров, как из справочной системы, так и Интернета. В окне Solution Explorer (Обозреватель решений) размещаются проекты и файлы текущего решения. К средству, призванному облегчить разработку, относится также окно Toolbox (Инструментарий), отображающее элементы, используемые в проектах. Окно Properties (Свойства) предназначено для отображения и настройки свойств объектов решения, включая форму и размещенные в ней объекты. Для получения подробной информации об объектах используется диалоговое окно Object Browser (Просмотр объектов). Оно позволяет искать и исследовать элементы, их свойства, методы, события, находящиеся в проектах и ссылках на них, как бы представляя собой внутреннюю библиотеку. Для удобства разработки используется окно Dynamic Help

(Динамическая справка). Во время работы оно постоянно обновляется и в нем предлагается справочная информация, относящаяся к текущим действиям.

Решения и проекты — это контейнеры, которые Visual Studio использует для размещения и группировки кода, который пишется в интегрированной среде. Контейнеры — это объекты, внутри которых размещены другие объекты. *Решения* — это виртуальные контейнеры; они группируют свойства, относящиеся к одному (или нескольким) содержащимся в решении проектам. Решения не подвергаются обработке компилятором. Внутри интегрированной среды можно настроить несколько свойств на уровне решения. *Проекты* имеют и виртуальный, и физический характер. Помимо функционирования в качестве организационных единиц для создающегося кода они также однозначно соответствуют результатам, получаемым на выходе компилятора. Иначе говоря, Visual Studio превращает проекты в откомпилированный код (эквивалентную программу на машинном языке).

Каждый проект приводит к созданию *.NET-компонента* (такого как файл с расширением *dll* — динамическая библиотека, или *exe* — исполняемый файл приложения).

1.4. Язык программирования C#

C# (произносится «си шарп») — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Java, Delphi, Модула и Smalltalk — C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# не поддерживает множественное наследование классов (в отличие от C++).

1.5. Интерфейс Microsoft Visual Studio

Для начала работы в интегрированной среде разработки программного обеспечения Microsoft Visual Studio необходимо ее запустить путем выбора в меню «Пуск» - подменю «Все программы», далее находим пункт «Microsoft Visual Studio 20xx».

Итак, загрузив MVS, пользователю откроется окно программы, содержащее различные элементы интерфейса (рисунок 1).

Рассмотрим интерфейс открывшегося окна. Он включает совокупность панелей меню и инструментов сверху; слева обозреватель серверов; справа панель элементов

и обозреватель решений; ниже «Список ошибок»; в центре в виде вкладки «Начальная страница».

Остановимся подробнее на элементах интерфейса программы. Можно отметить, что «Панель меню» и «Панель инструментов» имеет вид и содержит широко распространенные элементы, присущие интерфейсу программ пакета Microsoft Office.

Стоит лишь обратить внимание на то, что содержимое элементов «Панель меню» будет изменяться в зависимости от того, на какой вкладке работает пользователь.

Интерфейс MVS интуитивен, т. е. понятен на интуитивном уровне, а мощная справочная система позволит найти ответы на возникшие вопросы и не только по интерфейсу MVS.

Рассмотрим, что же располагается в рабочей области окна.

1.«Обозреватель серверов». Предназначен для подключения к проекту серверов данных, баз данных, а также отображения списка подключенных серверов.

2.«Начальная страница». На этой странице пользователь может выбрать различные действия. Например: «Создать проект», «Открыть проект», обратиться к помощи различных руководств и ресурсов по MVS.

3.«Панель элементов». Здесь размещается список доступных элементов (компонентов, объектов), которые пользователь может использовать для разработки интерфейса разрабатываемой программы.

4.«Обозреватель решений». В этой области отображаются файлы и элементы проекта.

5.«Список ошибок». В данной области выводятся ошибки при компиляции разрабатываемого проекта.

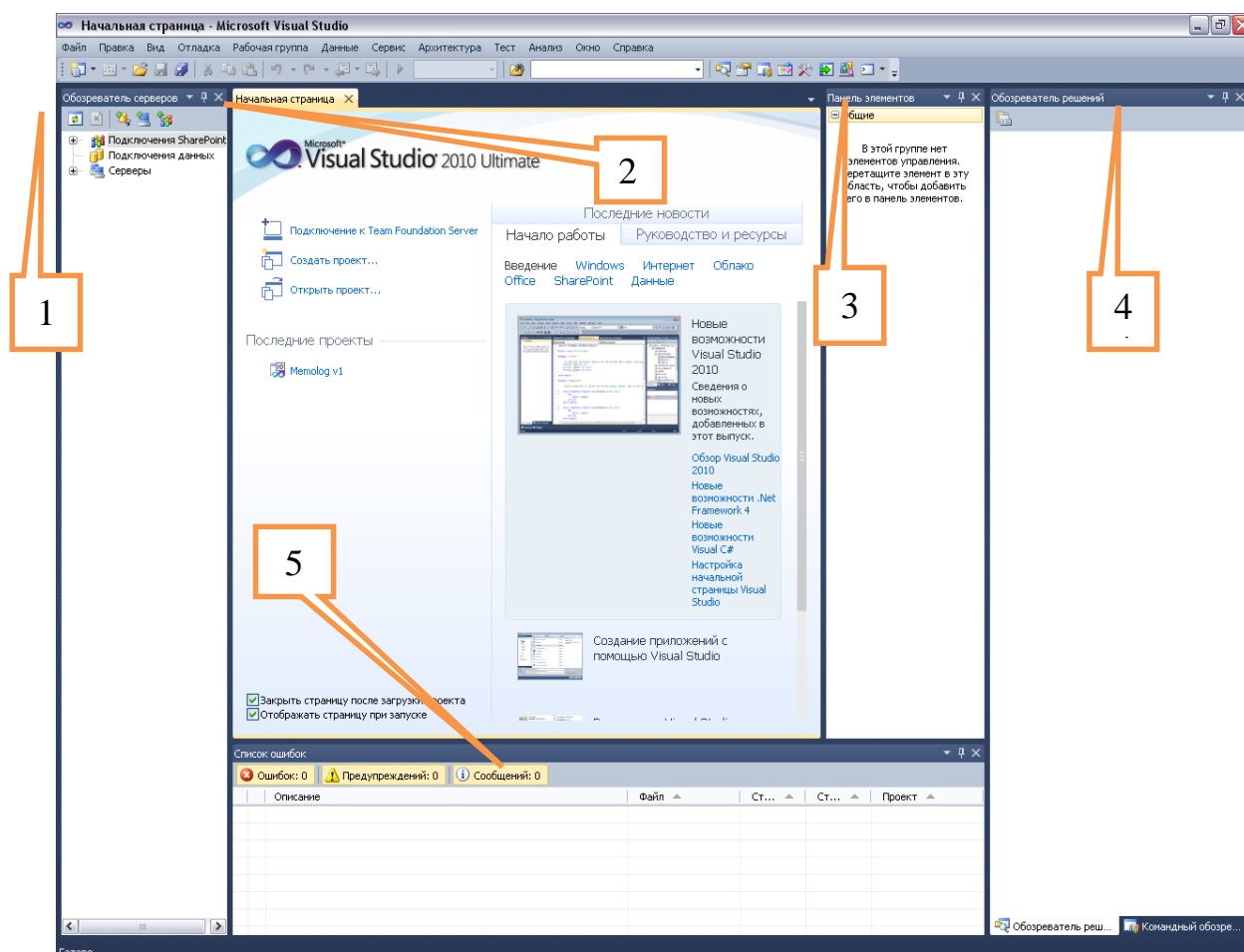


Рисунок 1 - Окно программы

Для создания нового проекта, кликните по ссылке «Создать проект», расположенной на «Начальной странице». В результате этого действия откроется окно «Создать проект», где пользователю предлагается выбрать язык разработки, тип проекта, задать имя проекта, указать место размещения проекта и задать название решения (результата разработки) (рисунок 2).

Очень важно всегда отслеживать место расположения проекта. Если вы работаете на компьютере с локальной ОС, то место размещения файлов проекта может не играть особой роли, можно согласиться с размещением проекта, предлагаемым по умолчанию. Если Вы работаете на компьютере с сетевой ОС, то стоит размещать файлы проекта только в отведенном Вам разделе жесткого диска. Будьте внимательны с тем, куда вы сохраняете Ваши проекты.

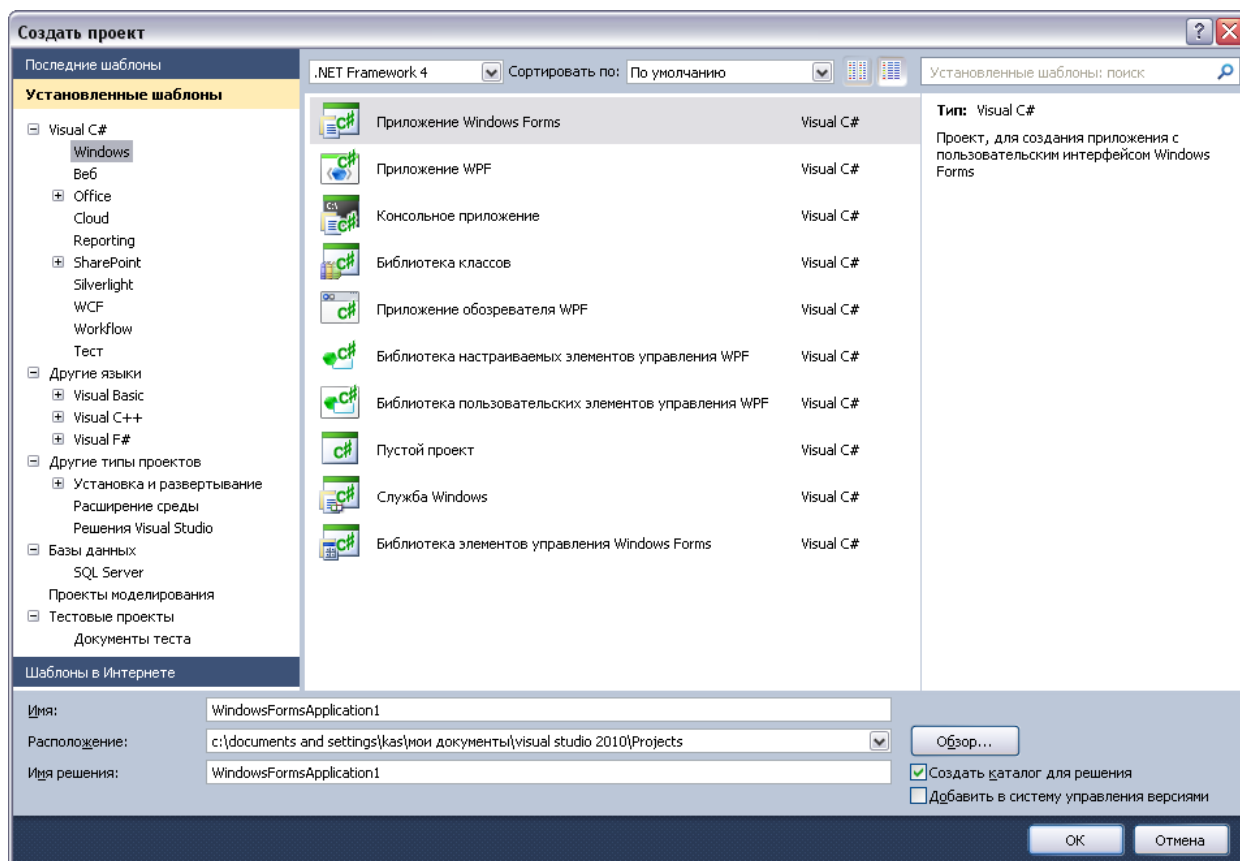


Рисунок 2 - Окно создания проекта

Итак, приступим к созданию нового проекта. В первую очередь необходимо выбрать язык разработки, в нашем случае это C#. Кликнув по названию в средней части окна, необходимо выбрать тип проекта. В списке, типов проектов достаточно много, они предназначены для решений различных направлений. Пока обратим внимание только на два типа проектов:

1. Консольное приложение. Предназначено для разработки программ без загрузки графических интерфейсов и драйверов, они работают только с клавиатурой.

2. Приложение Windows Form. Предназначено для разработки программ под операционную систему Windows, с окнами, графическим интерфейсом, кнопочными элементами управления и т. д.

Выбирая тот или иной тип проекта, обратите внимание на пояснения, которые отображаются в правой части окна.

Далее зададим имя проекта. Не стоит называть проект набором цифр, букв, символов, а также специальными символами. Следует дать название проекту в соответствии с его назначением. Например: ChSAdip1, первые три — ФИО, а дальше dip1 обозначает, что разрабатывается программа в соответствии с заданием на первую дипломную работу, можно также добавить в название и номер группы.

Назвав проект, следует указать место его хранения (можно и нужно заранее создать в своей части жесткого диска каталог с названием предмета и подкаталогами с названием или номерами дипломных работ). Достаточно удобно будет, если у Вас в Вашем разделе жесткого диска будет размещен каталог, MSRPPPO/C#/diplom/dip1 и для каждой новой дипломной работы в папке C#/diplom/ будут размещаться папки с названиями dip 2, dip 3 и так далее.

Указав место размещения файлов проекта, дайте имя решению (имя можно задать такое же, как и у проекта).

После выполнения перечисленных выше действий, нажмем на кнопку ОК. В результате откроется окно проекта и соответственно изменится интерфейс. Главные изменения произойдут в области «Обозреватель решений», добавиться структура проекта (рисунок 3) и произойдут изменения в «Панели меню».

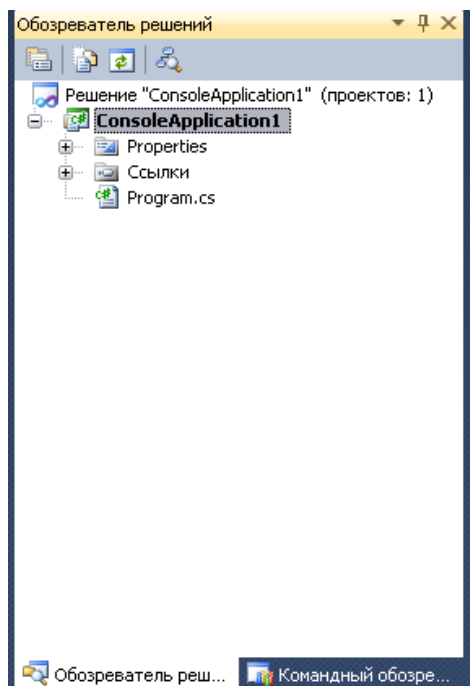


Рисунок 3 - "Область решений" для созданного проекта

Но не стоит на этом останавливаться, для дальнейшей корректной работы и во избежание потери данных необходимо произвести сохранение созданного проекта. Для это необходимо в меню «Файл» выбрать пункты «Сохранить все» и «Сохранить (название проекта)».

Теперь, обратим внимание на центральную часть окна MVS. В ней будут размещаться страницы файлов проекта. Самой первой страницей является «Program.cs». В ней будет происходить написание кода программы. MVS автоматически генерирует часть кода будущей программы (рисунок 4).

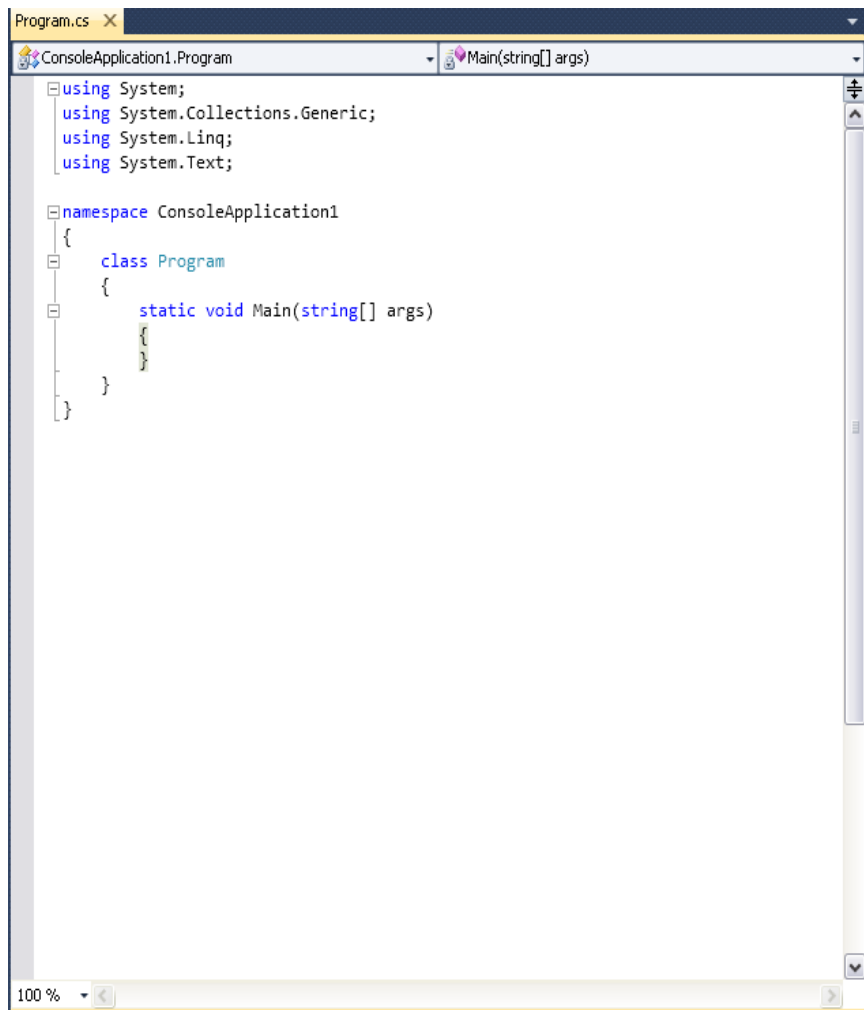


Рисунок 4 - Страница с кодом программы

Рассмотрим подробнее, что содержится на данной странице.

Блок:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

Здесь перечислены директивы. Если в начало программы поместить директиву *using System;*, то классы *System* и методы можно будет использовать напрямую без указания их полного имени. Например, можно вызвать *Console.WriteLine* вместо *System.Console.WriteLine*.

Программы на C#, как правило, используют службы ввода/вывода, предоставляемые библиотекой времени выполнения в .NET Framework. Оператор *System.Console.WriteLine* использует метод *WriteLine*. Это один из методов вывода класса *Console* в библиотеке времени выполнения. Он выводит свои строковые параметры в стандартном потоке вывода, за которым следует новая строка. Другие методы *Console* используются для разных операций ввода и вывода.

Блок:

```

namespace ConsoleApplication1
{

```

```
class Program
```

```
{  
    static void Main(string[] args)
```

Здесь автоматически генерируется код для различных методов и классов, использующихся при разработке программы.

Указываются пространство имен, класс и метод. Консольное приложение C# должно содержать метод Main, в котором начинается и заканчивается управление. В методе Main создаются объекты и выполняются другие методы.

Метод Main является статическим методом static, расположенным внутри класса или структуры. В рассматриваемом блоке, он расположен в классе с именем Program. Метод Main можно объявить одним из следующих способов:

– Он возвращает значение void:

```
static void Main()  
{  
    //...  
}
```

– Также может возвращаться целое значение:

```
static int Main()  
{  
    //...  
    return 0;  
}
```

– С этими возвращаемыми типами он может принимать следующие

аргументы:

```
static void Main(string[] args)  
{  
    //...  
}
```

-или-

```
static int Main(string[] args)  
{  
    //...  
    return 0;  
}
```

Параметр метода Main args является массивом значений типа string, который содержит аргументы командной строки, используемые для вызова программы.

Рассмотрим для начала, простейшую программу (с которой начинают все уважающие себя программисты при знакомстве с новым языком программирования).

Как это выглядит в MVS представлено на рисунке 6.

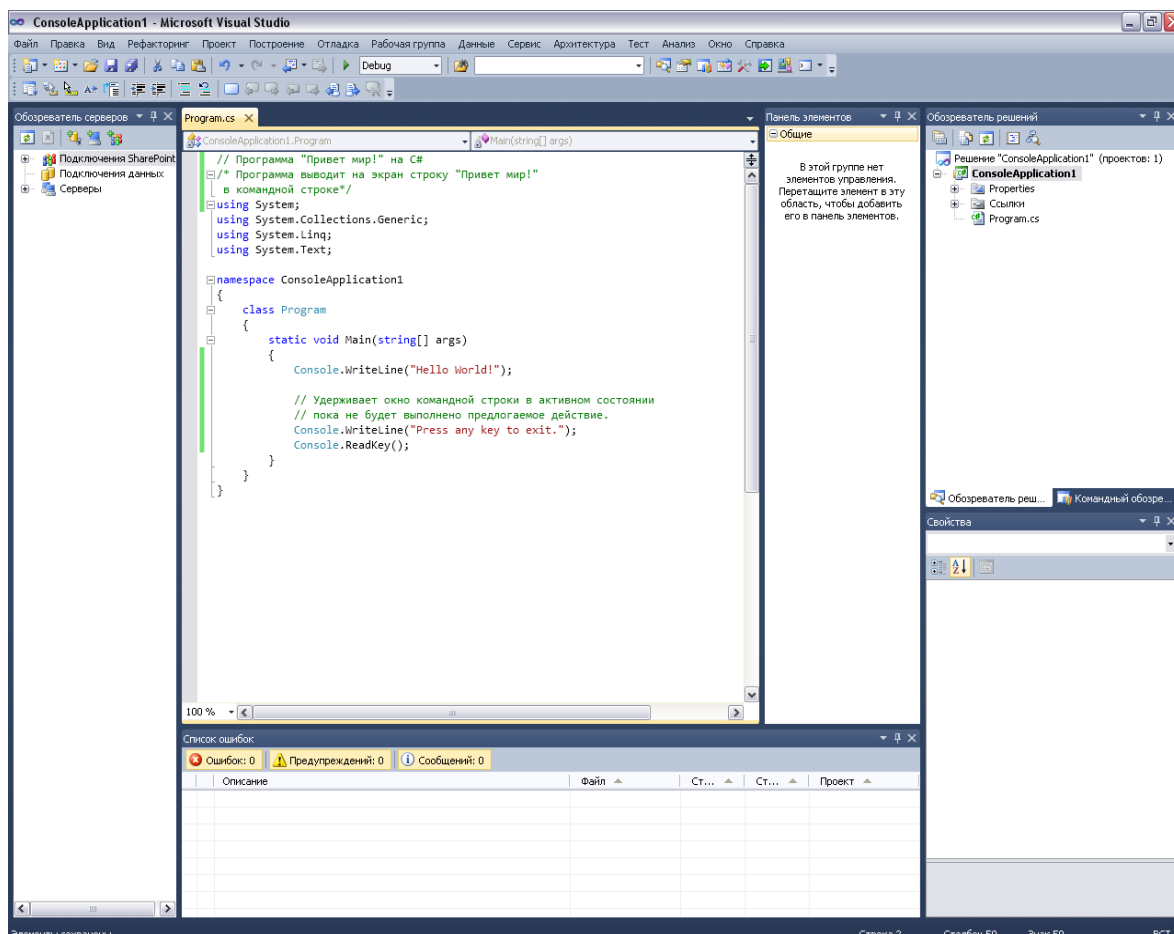


Рисунок 5 - Код программы в интерфейсе MVS

Программы пишутся не только для работы с клавиатурой, а еще под различные операционные системы (ОС). В нашем случае в дальнейшем нам предстоит научиться писать прикладные программы для работы под ОС Windows, поэтому программное обеспечение должно иметь форму (окно), удобный и понятный интерфейс (совокупность информативных управляющих элементов). Для этого нам понадобится изменить тип проекта с консольного на Windows Form.

Создадим новый проект, как это было описано выше, за исключением того, что выберем проект — приложение Windows Form.

В результате окно разработки приложения изменится (рисунок 7).

Нужно отметить появившиеся изменения в интерфейсе. Появились элементы в области «Панель элементов», область «Свойства», а так же появилась область «Источник данных».

На «Панели элементов» размещены наборы элементов, при помощи которых, пользователь может формировать интерфейс будущего ПО, в области «Свойства» перечислены свойства выбранного объекта (форма, или элемент), а в области «Источник данных» будут отображаться различные источники данных для программ, например, база данных для программ, работающих с наборами данных и т. п.

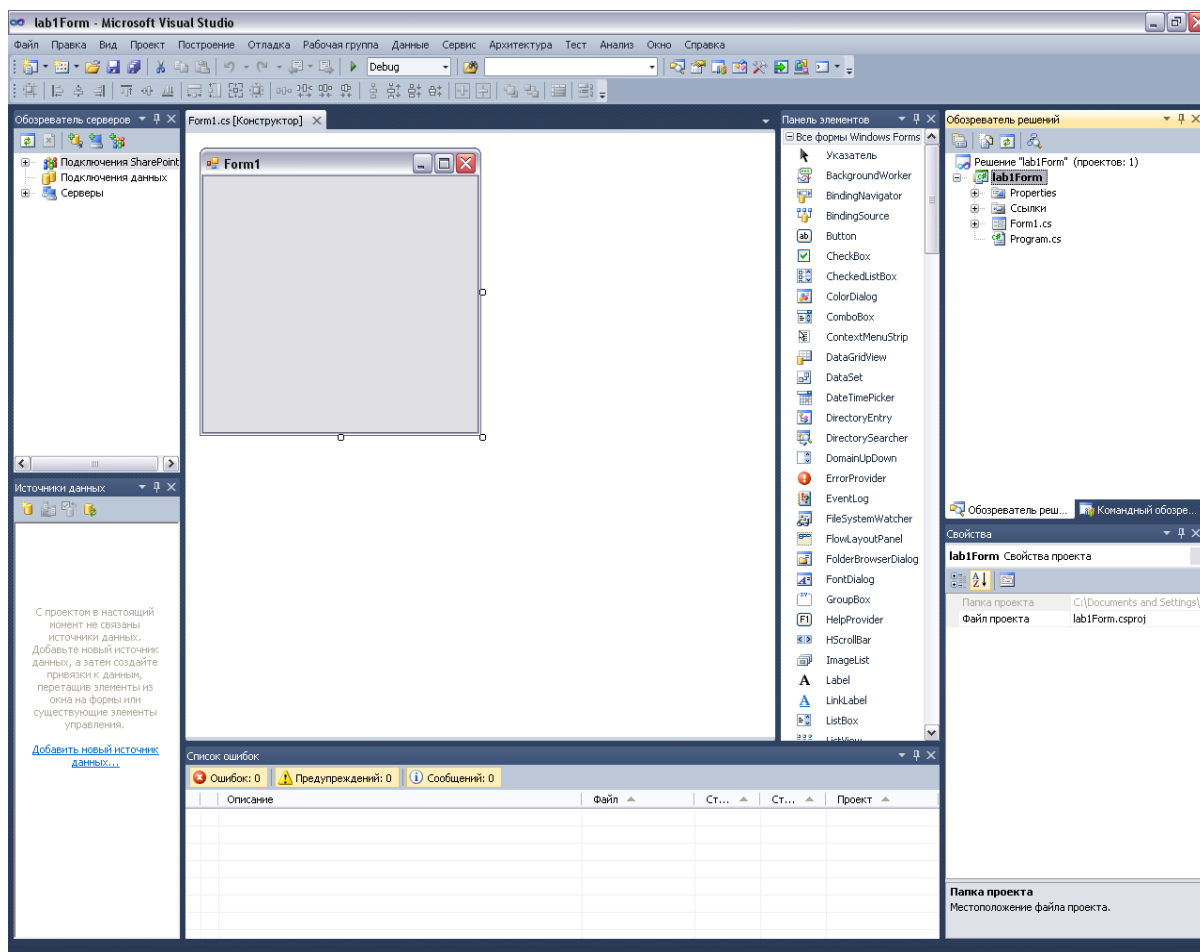


Рисунок 6 - Окно разработки приложения Windows Form.

Есть и еще одно отличие, теперь в центре рабочей области вместо страницы с кодом, как при разработке консольного приложения, размещена форма. Форма – область (пространство), где пользователь будет размещать те или иные элементы интерфейса и управляющие элементы (кнопки, поля ввода данных, метки, меню и т. д.)

Обратите внимание, разместив тот или иной элемент на форме или даже просто кликнув по форме или элементу, находящемуся на ней, в области «Свойства» появится список свойств и параметров выбранного объекта. Например, выбрав форму, можно просмотреть и изменить некоторые параметры формы (название, размер, цвет фона, разместить фоновый рисунок и т. д.) (рисунок 8).

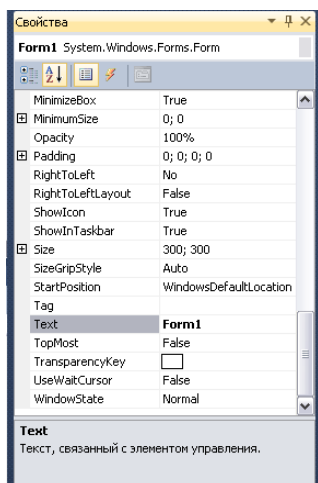


Рисунок 7 - Область "Свойства" для объекта "Форма"

Кроме свойств объектов, в той же области, если кликнуть по значку «События», появится список событий для выбранного элемента (рисунок 9). Если свойства объектов направлены на хранение и изменение их характеристик, то события описывают поведенческую сторону объектов, например, нажатие, двойное нажатие, перетаскивание, окончание редактирования, движение мыши над объектом и др. Для каждого выбранного свойства, параметра или события элемента ниже можно увидеть описание.

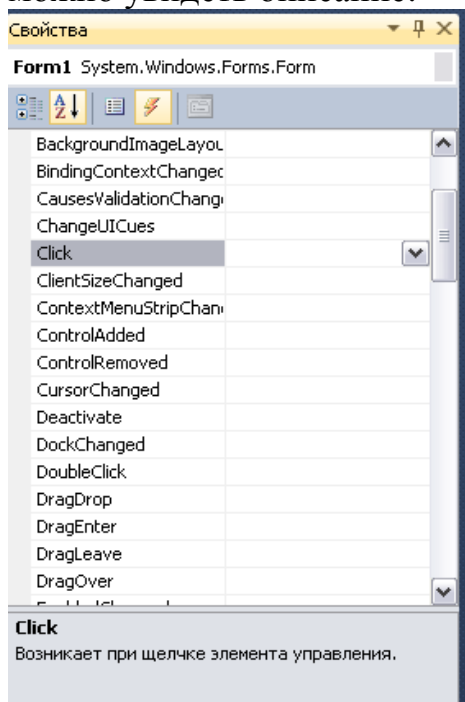


Рисунок 8 - Область "Свойства", раздел "События" для объекта "Форма"

После того как элементы вынесены на форму, посмотрим, как изменилась область «Обозреватель решений» (рисунок 10).

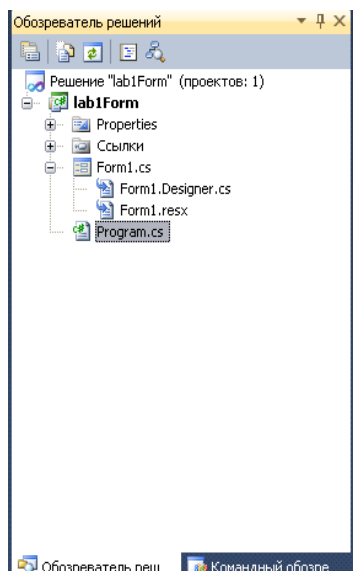


Рисунок 9 - "Обозреватель решений"

Обратим внимание на то, что появились новые файлы в структуре проекта Form1.Designer.cs и Form1.resx. В этих файлах содержится информация о размещенных на форме объектах и информация о ресурсах формы. В файле Form1.cs содержится автоматически сгенерированный код формы. В файле Program.cs содержится автоматически сгенерированный код проекта. Для открытия файла, находящегося в «Обозревателе решений», нужно дважды кликнуть по нему, например, выбрать файл Program.cs (рисунок 11).

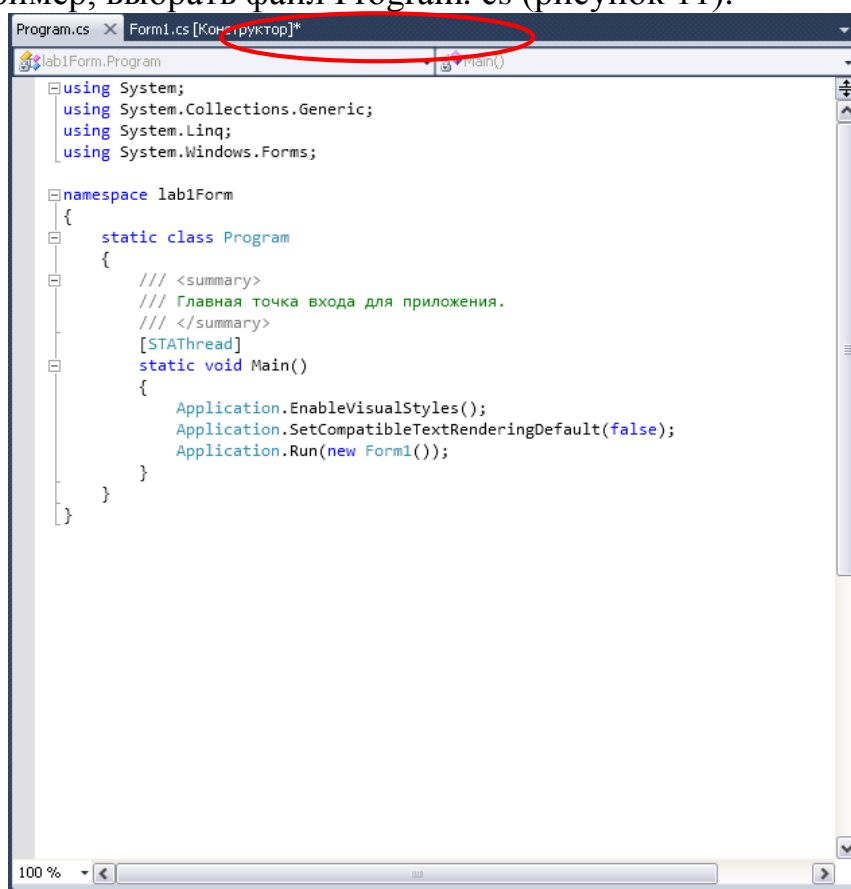


Рисунок 10 - Файл Program.cs

При корректной и полной установке MVS, при наведении на строку, можно получить подсказку о ее назначении.

Для перехода к конструктору форм, просто кликнем по названию вкладки находящейся рядом.

Теперь нужно сделать так, чтобы строка «Выберете действие:» при нажатии на кнопку отобразилась на форме.

Для обработки событий в коде должны присутствовать соответствующие методы. Заготовки этих методов добавляются в текст файла **Form1.cs** с помощью инструментария MVS. Например, выполнив двойной щелчок левой клавишей мыши на соответствующей кнопке в визуальном представлении формы, в исходный код будет добавлен код заготовки метода, реализующего обработчик события, при этом автоматически произойдет переход в режим редактирования кода.

Итак, у Вас уже есть форма. Разместим на ней элемент кнопка (Button1) и метку (Label1). Далее при нажатии кнопки в поле метки должна появиться заданная строка, следовательно, необходимо запрограммировать событие click (нажатие) для кнопки, которое в свою очередь изменит свойство text для метки. Для того, чтобы приступить к написанию кода для события click, нужно дважды щелкнуть мышью по кнопке на форме (или выбрать на форме кнопку и в области «Свойства» нажать на кнопку «События» и щелкнуть напротив Click на пустую строку), этим действием автоматически генерируется и добавляется в файл Form1. cs код обработчика события Click (рисунок 12).

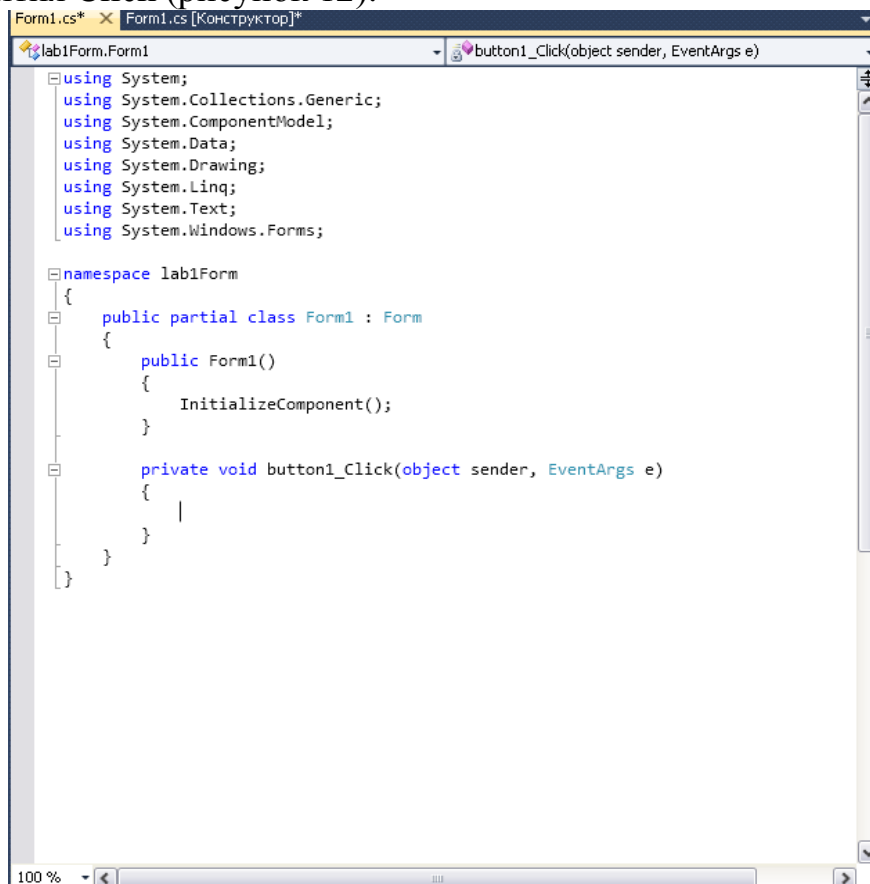


Рисунок 11 - Файл Form1. cs код программы

```
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;

namespace lab1Form
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }
    }
}

```

Обратим внимание на последнюю строку в коде с именем кнопки и названием события для нее, это и есть наш сгенерированный обработчик события. Для изменения свойства text элемента метка между фигурными скобками в обработчике button1_Click

необходимо ввести следующий программный код:

label1. text = “Выберете действие:”

При вводе названия элемента метка (label1) для упрощения ввода система предлагает выбрать из списка существующие названия элементов, процедур, методов и классов (рисунок 13). Необходимо учесть, что если на форме отсутствует элемент, название которого вы вводите (или необъявленный класс, метод, процедура), то его в списке не будет.

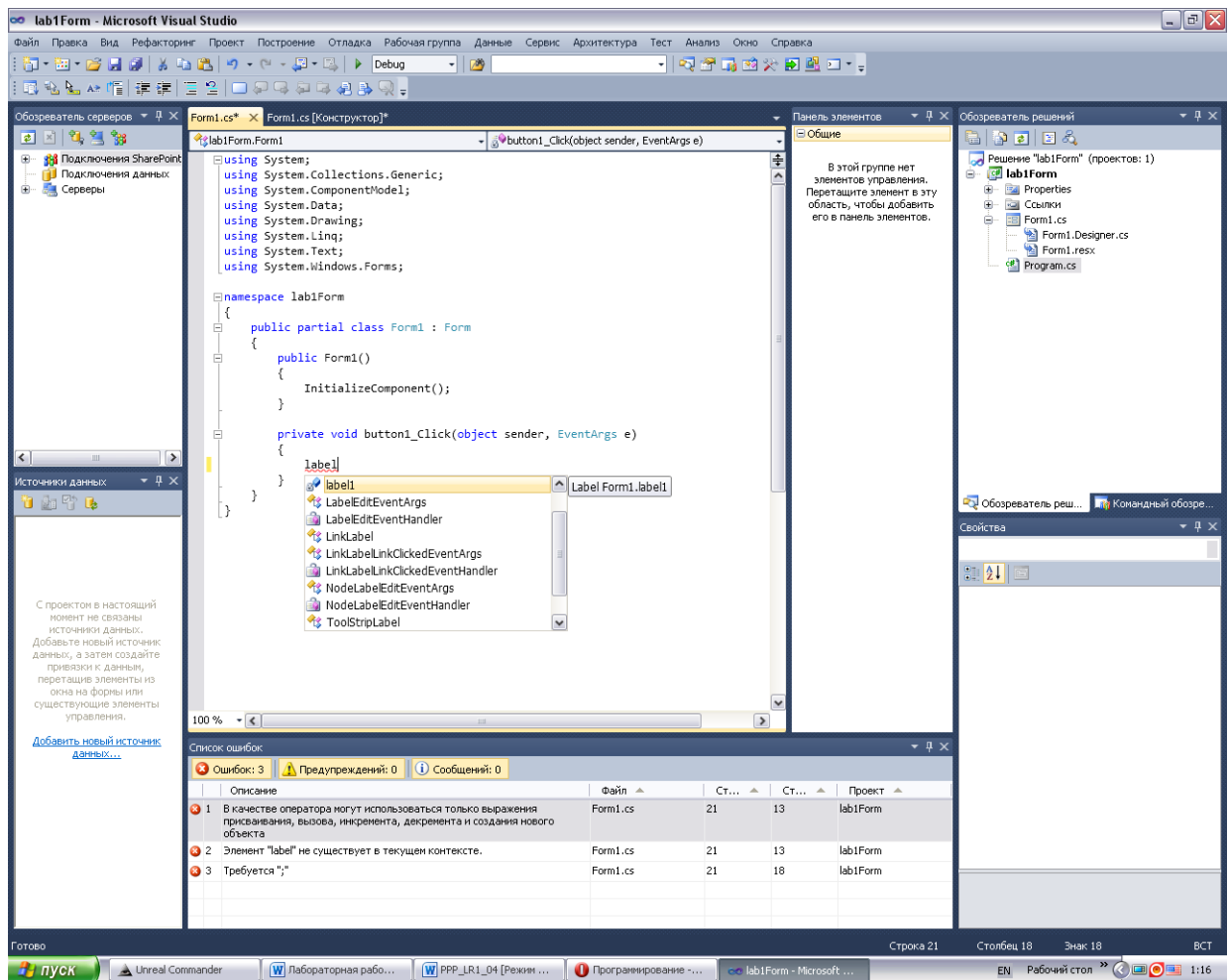


Рисунок 12 - Вывод списка существующих элементов

Теперь посмотрим, что же получилось в результате наших действий (рисунок 14).

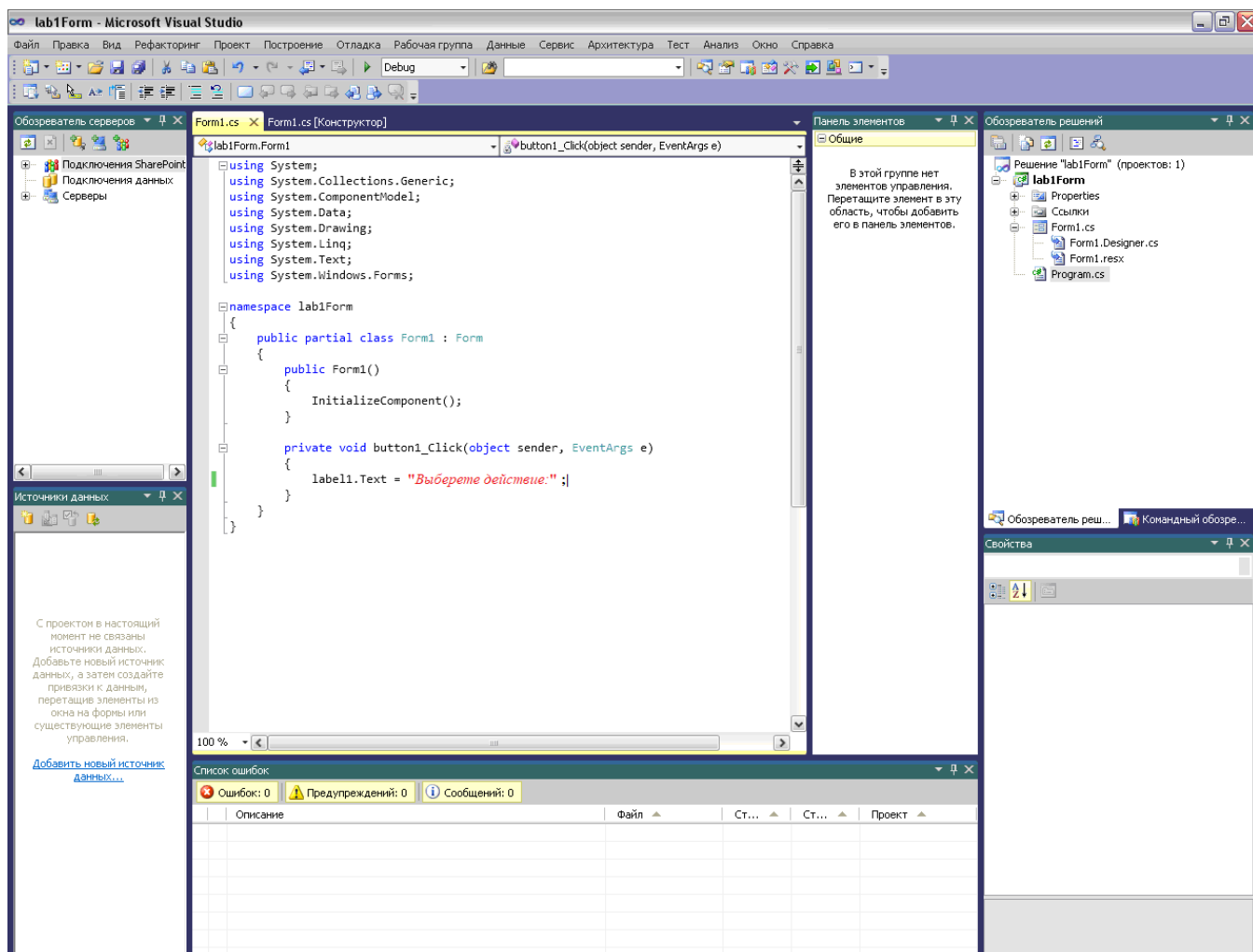


Рисунок 13 - Готовый код программы

Рассмотрим подробнее строку `label1.Text = "Выберете действие:"`;

Label1 - название элемента, который отображает надпись, text – свойство элемента label1, хранящее эту надпись, знак = это знак присвоения, “Выберете действие:” – текст, который будет отображаться в label1 по причине изменения свойства text этого элемента, т. е. новое значение свойства.

Наконец, для просмотра результата выполнения написанной программы необходимо запустить ее на отладку и выполнение, нажав на клавиатуре кнопку F5 (рисунок 15, 16).

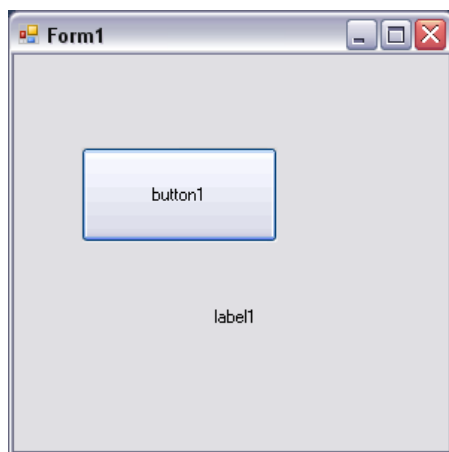


Рисунок 14 - Окно запущенной программы

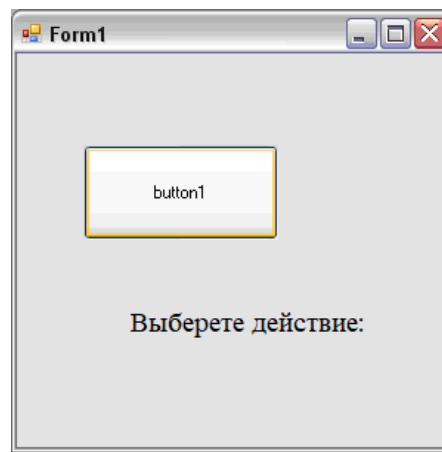


Рисунок 15 - Результат нажатия кнопки

Убедившись в том, что программа разработана, верно, и без ошибок, необходимо сделать так, чтобы пользователю было понятно, как с ней работать. Для этого обратимся к свойствам формы и тех элементов, что размещены на ней.

Для начала, необходимо изменить название формы и кнопки. Изменим название формы Form1 на «Menu». Обратимся к свойствам этой формы, кликнем левой кнопкой мыши по форме и посмотрим на область «Свойства», где необходимо изменить свойство text. Справа от названия свойства, в строке будет содержаться текст Form1, удаляем его и пишем «Menu», далее нажимаем Enter (рисунок 17).

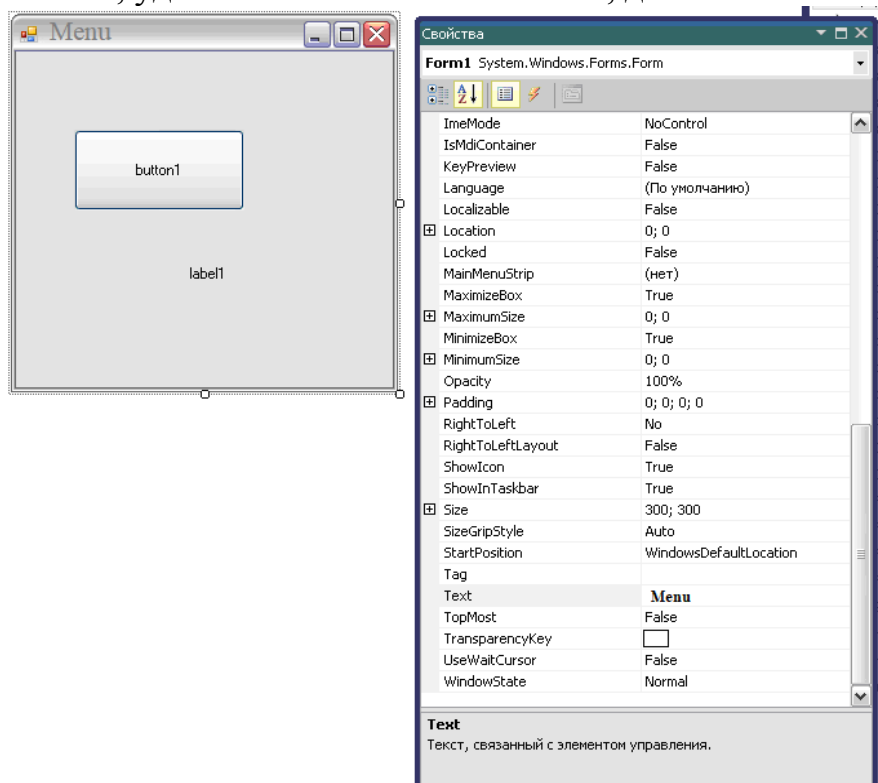


Рисунок 16 - Результат изменения свойства text объекта Форма

Таким же образом необходимо изменить надпись на кнопке. Например: «Нажми меня» (рисунок 18).

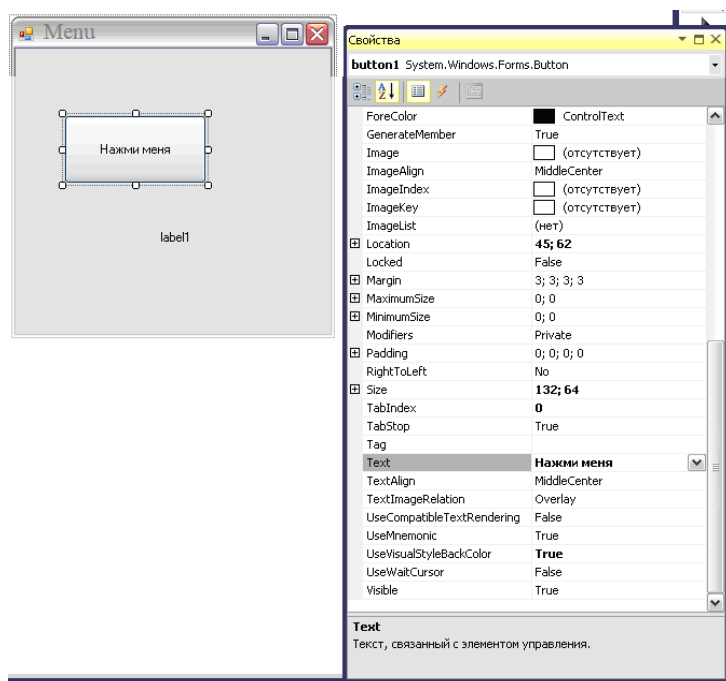


Рисунок 17 - Результат изменения свойства text элемента кнопка

Теперь можно изменить свойства текста, который будет отображаться в элементе label1. В области «Свойства» элемента label1 находим строку font, это свойство служит для настройки отображаемого текста. Нажмите на знак перекрестия рядом с названием свойства или на знак многоточия в строке справа от названия свойства, тем самым, получив инструменты для изменения шрифта (рисунок 19). В данном случае свойство является составным. Измените параметры шрифта по своему усмотрению.

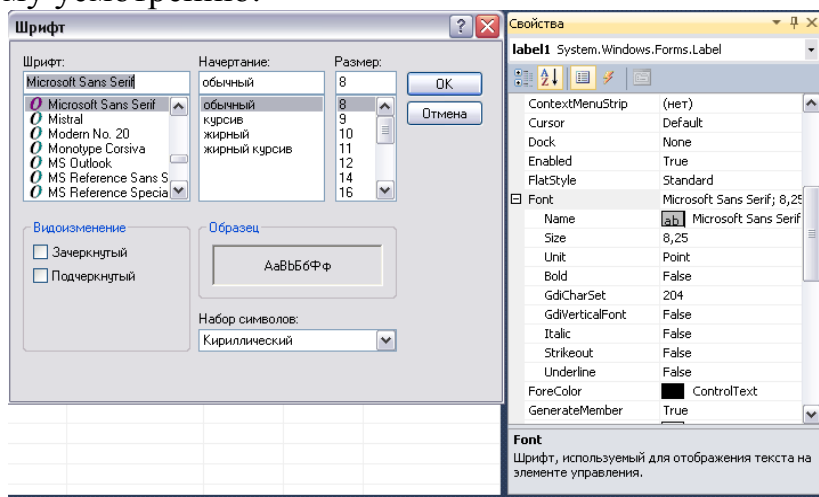


Рисунок 18 - Инструмент изменения шрифта отображаемого текста

Далее сохраните проект и запустите его на компиляцию и выполнение. Убедитесь в произошедших изменениях (рисунок 20, 21).

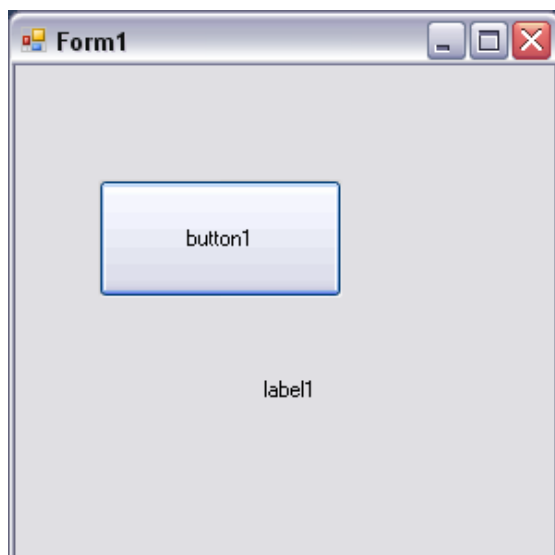


Рисунок 19 - До изменения свойств элементов

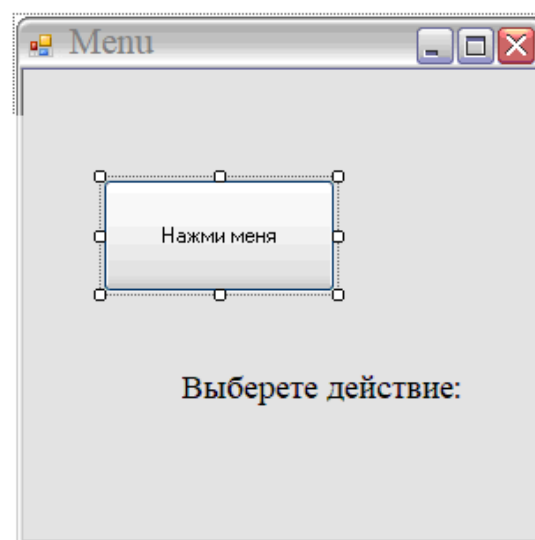


Рисунок 20 - После изменения свойств элементов

Свойства можно менять как программно, так и при разработке дизайна формы. Если добавить в обработчик событий строку (рисунок 22):

button1.Text = "Меня уже нажимали";

то в результате, после выполнения программы, название кнопки изменится (рисунок 23).

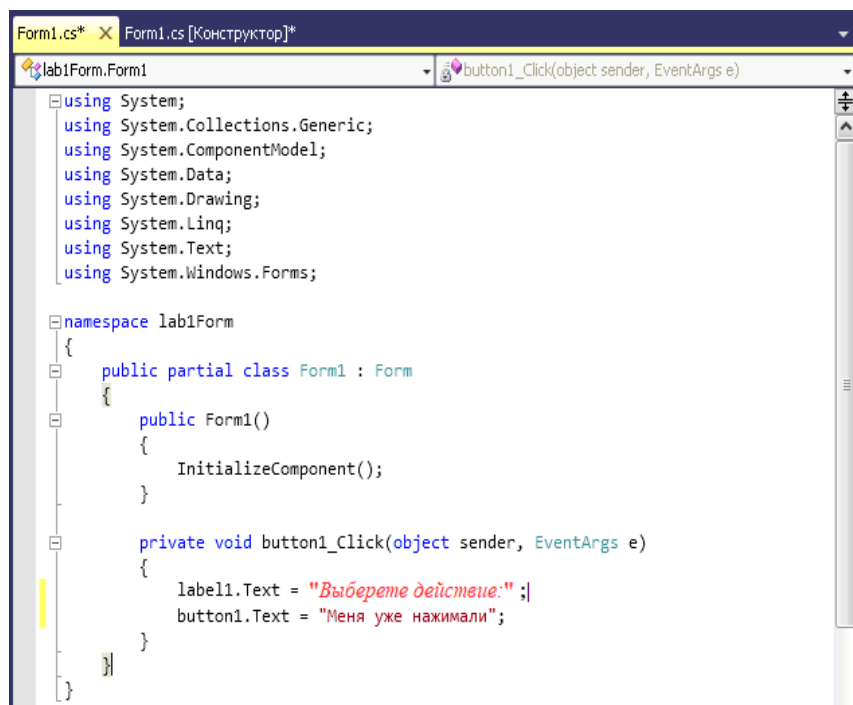


Рисунок 21 - Добавление строки в обработчик событий

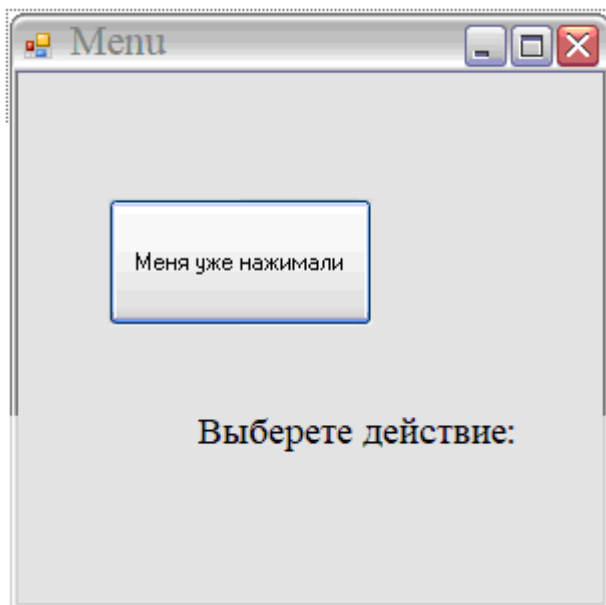


Рисунок 22 - Результат изменений

Рекомендации по разработке программы

Процесс разработки программы можно представить себе в виде схемы, приведенной на рисунке 24. Нужно помнить, что время, потраченное еще до начала фактического программирования на всесторонний анализ поставленной задачи, относится к числу полезных затрат. Эти затраты более чем окупят себя экономией времени, которое будет потрачено на исправление ошибок.

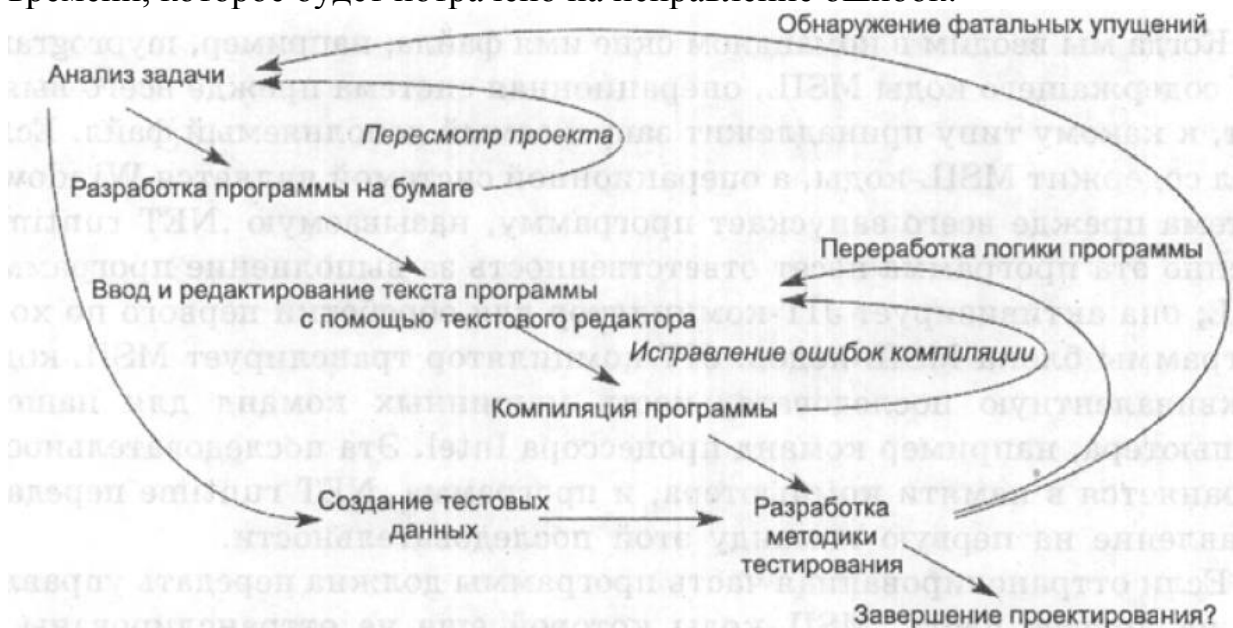


Рисунок 23 - Схема процесса разработки программы

1.6. Что лучше Delphi или Visual Studio?

Когда говорят о преимуществах той или иной программной среды перед другой всегда вспоминается софизм: «Что лучше, левый глаз или правый глаз? Каким глазом смотреть? А какой можно выкинуть на помойку?» ;)))

Вообще, Delphi конечно легкий для понимания, но VS как-то по умолчанию считается более прогрессивной. Все-таки Delphi - это наследие Паскаля, а Паскаль в первую очередь язык для обучения программированию, нежели для практического написания ПО.

Visual Studio сможете использовать бесплатную версию Express или Community Edition, Delphi же все платные.

Перечисление остальных бесспорных преимуществ VS перед Delphi в присутствии дельфийцев перерастёт в холивар Delphi vs C#.

в VS .NET есть:

- представление исходника в виде иерархического дерева с возможностью скрытия отдельных участков кода;
 - продвинутый синтаксический анализатор, показывающий ошибки непосредственно после ввода текста;
 - динамическая справка;
 - поддержка диаграмм UML и средств работы с XML;
- и это далеко не все.

Потом в VS .NET можно работать не только на C++, но и на C#, VB .NET --> универсальная среда.

К VS .NET прилагается море документации. Чего стоит один MSDN.

У каких IDE кроме VS .NET есть все это и еще что-то более продвинутое? На мой взгляд, если в ближайшие несколько лет не нужно выпускать коммерческих продуктов, а программировать только в собственное удовольствие, тогда лучше VS.NET. Т.к. .net еще не очень распространен. Хотя с другой стороны .net продуктов мало и спрос будет постоянно расти, есть над чем задуматься.

VS позволяет работать и с WPF и с Silverlight и с ASP .NET и с XNA и еще многим интересным и полезным. Не хочется говорить что язык между прочим как и Java межплатформенный поддерживает многопоточность и еще много чего интересного. Забудьте о функциях и процедурах переходите на методы и классы. А по поводу Студии скажу так один из самых мощных инструментов какие я встречал работай хоть один хоть 100 человек всем удобно все работает настраиваемо. Так же отдельно хочется сказать как выразились выше графическую прогу на C# не напишешь. А на кой ляд придумали WPF после его обработки простым художником или дизайнером программист просто вбивает свой текст после чего программа не только работает, но и на неё очень даже хочется взглянуть, а тормоза если то скорее всего что-то где то неоптимизированно.

Но конечно и у среды Delphi есть свои плюсы:

- работает на реально древних машинах (я лично запускал на первом пентиуме, 166МГц + 48ОЗУ, и мелкие проги можно делать и компилировать).
- занимает немного места (вроде мегабайт 200-300 на диске).
- не требует .NET.
- работает на Windows 2000 и старше.
- список багов давно известный. Система предсказуемая.

- в большинстве случаев не нужно никаких дополнительных dll/манифестов.
- можно не устанавливать, а просто распаковать с архива и чуточку поднастроить и можно работать.

Delphi, минусы:

- вопросы по ней на форумах нужно задавать.
- BDE (но многим нравится)
- большой размер exe (да, пару сотен килобайт).
- нельзя писать на c++ .

VS, плюсы:

- новомодная система. на форумах выглядишь крутом ;))).
- навороченный дизайн, вызывающий восторг (капс в меню).
- тонны примочек, одни названия которых вызывают экстаз.
- можно писать на куче языков. Чего только стоит c++ в двух модификациях.
- поддержка последнего стандарта c++0x
- поддержка 8 винды с метро стилем.

VS, минусы:

- только одна инсталляция на пару гигабайт.
- после инсталляции обнаруживается куча мусора, начиная с эмуляторов устройств и заканчивая серверами, которые так и хотят процессорного времени.
- в очередной раз переработанный интерфейс.
- новые, неизведанные баги.
- не поддерживает Delphi. ;)))

ГЛАВА 2. ТЕОРИЯ ДЕЛИМОСТИ МНОГОЧЛЕНОВ

2.1. Многочлены в школьной программе.

Впервые с понятием многочлена учащиеся сталкиваются на уроках алгебры в 7 классе. Типовая учебная программа 7 класса имеет следующие разделы и темы базового (A2) и углублённого базового (A2+) уровней:

III РАЗДЕЛ. ОДНОЧЛЕНЫ И МНОГОЧЛЕНЫ (23 часа, А2+:10 часов)

Темы 17-19: Степень с натуральным показателем. Свойства степени с натуральным показателем. (3 часа)

Темы 20-21: Одночлен и его стандартный вид. Стандартный вид числа. (2 часа)

Темы 22-23: Умножение одночленов. (2 часа)

Контрольная работа. (1 час)

Работа над ошибками. Решение задач и тестов. (1 час)

Темы 24-26: Многочлены. Приведение подобных членов. (3 часа)

Темы 27-29: Сложение и вычитание многочленов. (3 часа)

Темы 30-31: Умножение одночлена на многочлен. (2 часа)

Темы 32-33: Умножение многочлена на многочлен. (2 часа)

Темы 34-35: Деление одночлена и многочлена на одночлен. (2 часа)

А2+: Задачи о кроликах. Принцип Дирихле и его обобщение. (2 часа).

А2+: Принцип Дирихле и деление целых чисел (2 часа).

А2+: Решение задач на принцип Дирихле (2 часа).

А2+: Сумма нескольких слагаемых (2 часа).

Контрольная работа (1 час, А2+: 1 час)).

Работа над ошибками. Решение задач и тестов (1 час, А2+: 1 час).

IV РАЗДЕЛ. РАЗЛОЖЕНИЕ МНОГОЧЛЕНОВ НА МНОЖИТЕЛИ (18 часов, А2+: 6 часов)

Темы 36-38: Вынесение общего множителя за скобки. (3 часа)

Темы 39-40: Метод группировки. (2 часа)

Темы 41-42: Квадрат суммы. (2 часа)

Контрольная работа (1 час)

Работа над ошибками. (1 час)

Темы 43-44: Квадрат разности. (2 часа)

Темы 45-47: Формула разности квадратов. (3 часа)

Темы 48-49: Применение нескольких способов разложения. (2 часа)

А2+: Применение в задачах формул сокращенного умножения (4 часа).

А2+: Последовательности: описание и задание (2 часа).

Контрольная работа. (1 час)

Работа над ошибками. Решение задач и тестов (1 час)

В дальнейшем с многочленами ученики встречаются на протяжении всего курса математики (и алгебры, и геометрии) общеобразовательной школы и в дальнейшем, уже не делая на этом акцента, в частности при решении линейных, квадратных и биквадратных уравнений, упрощении рациональных и иррациональных выражений, степеней с рациональным показателем и многих других.

2.2. Общее понятие.

2.2.1 Одночлен.

Одночленом называют алгебраическое выражение, являющееся произведением букв и чисел. Эти буквы и числа являются множителями данного одночлена. Одночлены или Монономы, простой вид математических выражений, прежде всего рассматриваемых и используемых в элементарной алгебре. Произведение, состоящее из числового множителя и одной или несколько переменных, взятых каждая с той или другой положительной отметкой степени, подразумевается также каждое отдельное число без буквенных множителей. Примеры О.: $x^{*}(-3)*y*1*x$, $1*a^{*}(-1)*b$, $a*0*b^{*}(-1/3)*a$, $3abc$.

2.2.2 Многочлен.

Многочленом называют сумму одночленов. Одночлены, входящие в эту сумму, называют членами многочлена. В математике, многочлены или полиномы от одной переменной — функции вида

$$F(x) = c_0 + c_1x + \dots + c_nx^n,$$

где c_i фиксированные коэффициенты, а x — переменная. Многочлены составляют один из важнейших классов элементарных функций. Многочлен (или полином) от n переменных — есть конечная формальная сумма вида

$$\sum c_I x_1^{i_1} x_2^{i_2} \dots x_n^{i_n},$$

где $I = (i_1, i_2, \dots, i_n)$ есть набор из целых неотрицательных чисел (называется мультииндекс), c_I — число (называемое «коэффициент многочлена»), зависящее только от мультииндекса I .

В частности, многочлен от одной переменной есть конечная формальная сумма вида

$$c_0 + c_1x^1 + \dots + c_nx^n$$

Коэффициенты многочлена обычно берутся из определённого коммутативного кольца R (чаще всего поля, например, поля вещественных или комплексных чисел). В этом случае, относительно операций сложения и умножения многочлены образуют кольцо (более того ассоциативно-коммутативную алгебру над кольцом R без делителей нуля) которое обозначается

$$R[x_1, x_2, \dots, x_n].$$

Например: $a^2+2ab+b^2$.

2.2.3 Стандартный вид многочлена.

Говорят, что многочлен имеет стандартный вид, если все его члены записаны в стандартном виде и среди них нет подобных.

Например: $2, a, a-b, a^2+2ab^2+b^2, 0$.

Многочлен стандартного вида, состоящий из двух одночленов, называют двучленом; многочлен стандартного вида, состоящий из трёх одночленов, называют трёхчленом и

т. д. Например:

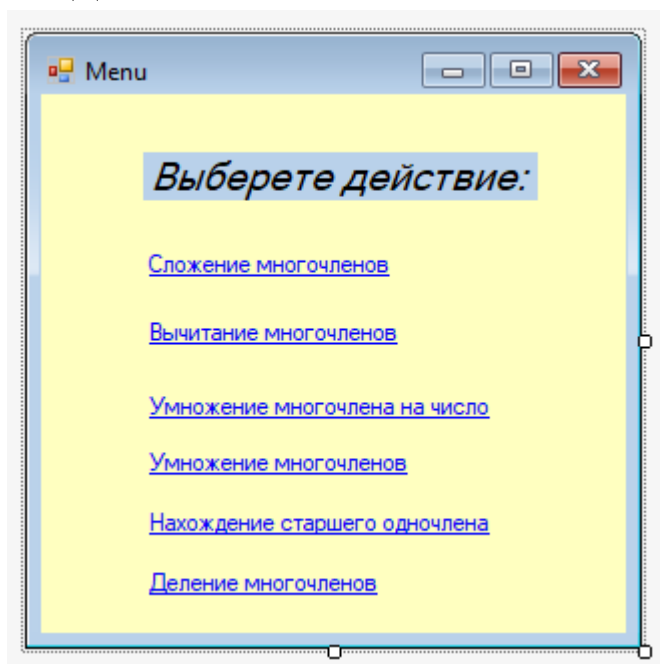
двучлен: $ab-cd, 0,7a^2-2b$;

трёхчлен: $3a-2b-7, x+yz-2z^2$;

четырёхчлен: $a+b-c-d, -abc-acd-bcd-abd$

Любой многочлен можно привести к стандартному виду.

2.3. Действия с многочленами.

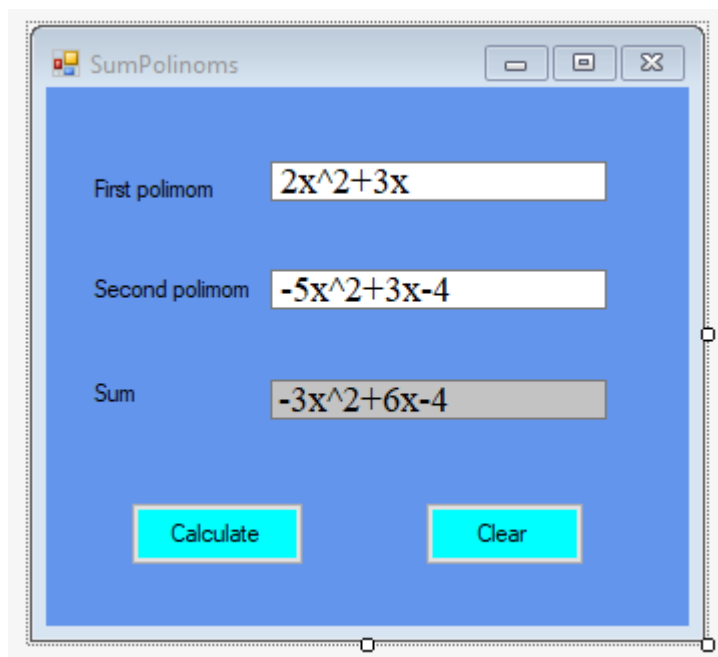


2.3.1 Сложение (вычитание) многочленов.

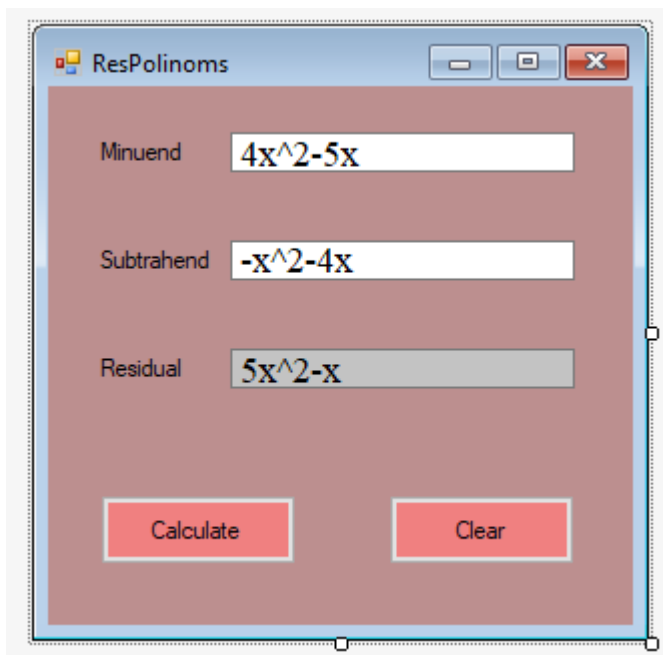
Суммой (разностью) двух многочленов называется многочлен, коэффициенты которого являются суммой (разностью) коэффициентов при подобных членах этих многочленов.

На практике для нахождения суммы и разности многочленов используют правила раскрытия скобок, перед которыми стоит знак плюс (знак минус).

Пример: $(2x^2+3x)+(-5x^2+3x-4)=2x^2+3x-5x^2+3x-4=-3x^2+6x-4$;



$$(4x^2-5x)-(-x^2-4x)=4x^2-5x+x^2+4x=5x^2-x$$



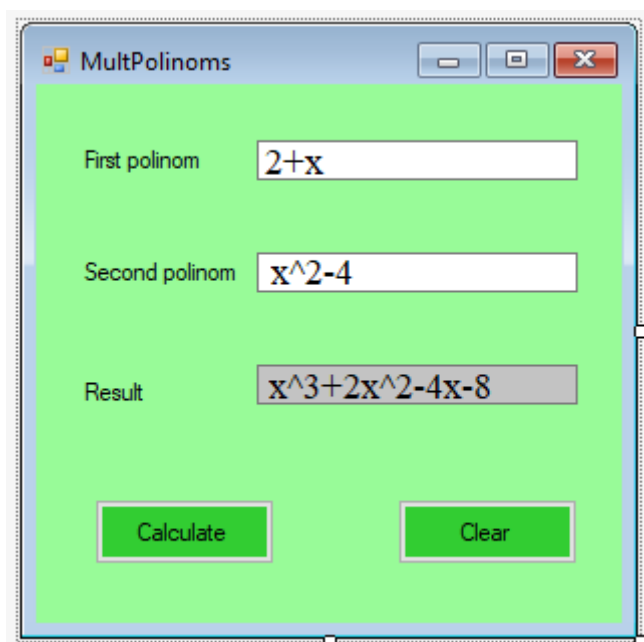
2.3.2 Умножение многочленов.

Чтобы умножить многочлен на одночлен, нужно умножить каждый член многочлена на этот одночлен и сложить полученные произведения.

Чтобы умножить многочлен на многочлен, нужно умножить каждый член первого многочлена на каждый член второго многочлена полученные одночлены сложить.

Пример: $(-5a)(4-b-a^2) = -20a + 5ab + 5a^3$;

$(2+x)(x^2-4) = 2x^2 - 8 + x^3 - 4x$



2.3.3 Деление многочленов

В алгебре, деление многочленов столбиком — алгоритм деления многочлена $f(x)$ на многочлен $g(x)$, степень которого меньше или равна степени многочлена $f(x)$. Алгоритм представляет собой обобщенную форму деления чисел столбиком, легко реализуемую вручную.

Для любых многочленов $f(x)$ и $g(x)$, $g(x) \neq 0$, существуют единственные полиномы $q(x)$ и $r(x)$, такие что

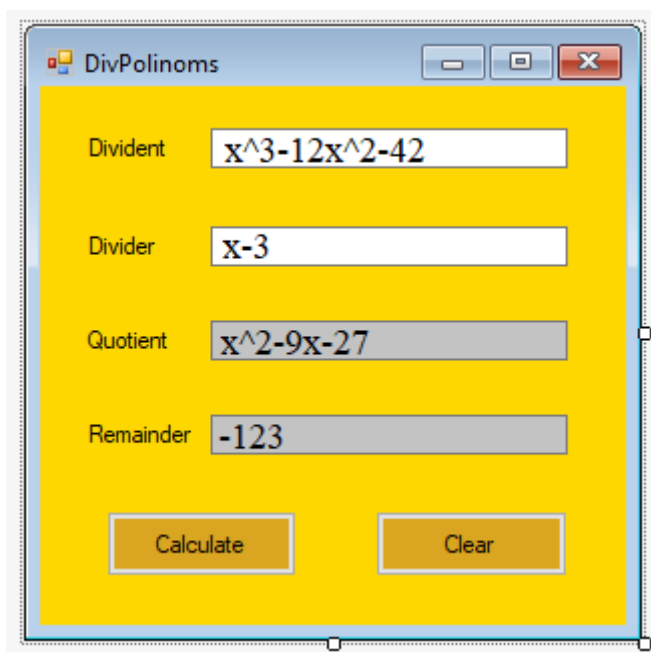
$$\frac{f(x)}{g(x)} = q(x) + \frac{r(x)}{g(x)},$$

причем $r(x)$ имеет более низкую степень, чем $g(x)$.

Целью алгоритма деления многочленов в столбик является нахождение частного $q(x)$ и остатка $r(x)$ для заданных делимого $f(x)$ и ненулевого делителя $g(x)$.

Пример:

Покажем, что $\frac{x^3-12x^2-42}{x-3} = x^2 - 9x - 27 - \frac{123}{x-2}$



Частное и остаток от деления могут быть найдены в ходе выполнения следующих шагов:

а). Делим первый элемент делимого на старший элемент делителя, помещаем результат под чертой $(x^3/x = x^2)$.

$$\begin{array}{r} x^3 - 12x^2 + 0x - 42 \overline{) x - 3} \\ \underline{x^2} \end{array}$$

б). Умножаем делитель на полученный выше результат деления (на первый элемент частного). Записываем результат под первыми двумя элементами делимого $(x^2 \cdot (x - 3) = x^3 - 3x^2)$.

$$\begin{array}{r} x^3 - 12x^2 + 0x - 42 \overline{) x - 3} \\ x^3 - 3x^2 \overline{) x^2} \end{array}$$

в). Вычитаем полученный после умножения многочлен из делимого, записываем результат под чертой

$$(x^3 - 12x^2 + 0x - 42 - (x^3 - 3x^2) = -9x^2 + 0x - 42)$$

$$\begin{array}{r} x^3 - 12x^2 + 0x - 42 \mid x - 3 \\ x^3 - 3x^2 \mid x^2 \\ \hline -9x^2 + 0x - 42 \end{array}$$

г). Повторяем предыдущие 3 шага, используя в качестве делимого многочлен, записанный под чертой.

$$\begin{array}{r} x^3 - 12x^2 + 0x - 42 \mid x - 3 \\ x^3 - 3x^2 \mid x^2 - 9x \\ \hline -9x^2 + 0x - 42 \\ -9x^2 + 27x \\ \hline -27x - 42 \end{array}$$

д). Повторяем шаг 4.

$$\begin{array}{r} x^3 - 12x^2 + 0x - 42 \mid x - 3 \\ x^3 - 3x^2 \mid x^2 - 9x - 27 \\ \hline -9x^2 + 0x - 42 \\ -9x^2 + 27x \\ \hline -27x - 42 \\ -27x + 81 \\ \hline -123 \end{array}$$

е). Конец алгоритма.

Таким образом, многочлен $q(x) = x^2 - 9x - 27$ частное деления, а $r(x) = -123$ — остаток.

2.4. Делимость многочленов

Говорят, что многочлен $P(x)$ делится на многочлен $Q(x)$, если существует многочлен $S(x)$, такой, что $P(x) = Q(x)S(x)$. Многочлен $S(x)$ называется частным от деления $P(x)$ на $Q(x)$.

Теория делимости многочленов имеет много общего с теорией делимости целых чисел. В частности, выполняются следующие свойства:

1) Если $P_1(x)$ и $P_2(x)$ делятся на $Q(x)$; то $P_1(x) + P_2(x)$ и $P_1(x) - P_2(x)$ делятся на $Q(x)$;

2) Если $P(x)$ делится на $Q(x)$; а $T(x)$ — произвольный многочлен; то $P(x)T(x)$ делится на $Q(x)$;

3) Если $P(x)$ делится на $Q(x)$; а $Q(x)$ делится на $H(x)$; то $P(x)$ делится на $H(x)$;

Доказательство этих свойств ничем не отличается от доказательства соответствующих свойств делимости целых чисел. Отметим еще некоторые простые свойства:

4) Если ненулевой многочлен $P(x)$ делится на $Q(x)$; то $\deg P(x) \geq \deg Q(x)$;

5) Если $\deg P(x) = \deg Q(x)$; то $P(x)$ делится на $Q(x)$ тогда и только тогда, когда эти многочлены пропорциональны:

(Многочлены называются пропорциональными, если один из них получается из другого умножением на число, отличное от 0.) Действительно, если $P(x)$ делится на $Q(x)$ и $\deg P(x) = \deg Q(x)$, то частное имеет степень 0, т.е. является числом, отличным от 0. Обратное утверждение очевидно.

2.5. Алгоритм Евклида.

2.5.1 Исторические сведения.

Древнегреческие математики называли этот алгоритм «взаимное вычитание». Этот алгоритм не был открыт Евклидом, так как упоминание о нём имеется уже в Топике

Аристотеля. В «Началах» Евклида он описан дважды — в VII книге для нахождения наибольшего общего делителя двух натуральных чисел и в X книге для нахождения наибольшей общей меры двух однородных величин. В обоих случаях дано геометрическое описание алгоритма, для нахождения «общей меры» двух отрезков.

Историками математики (Цейтен и др.) было выдвинуто предположение, что именно с помощью алгоритма Евклида (процедуры последовательного взаимного вычитания) в древнегреческой математике впервые было открыто существование несоизмеримых величин (стороны и диагонали квадрата, или стороны и диагонали правильного пятиугольника). Впрочем, это предположение не имеет достаточных документальных подтверждений. Алгоритм для поиска наибольшего общего делителя двух натуральных чисел описан также в I книге древнекитайского трактата Математика в девяти книгах.

Ряд математиков средневекового Востока - Сабит ибн Курра, ал-Махани, Ибн ал-Хайсам, Омар Хайям, попытались построить на основе алгоритма Евклида теорию отношений, альтернативную по отношению теории отношений Евдокса, изложенной в V книге «Начал» Евклида. Согласно определению, предложенному этими авторами, четыре величины, первая ко второй и третья к четвёртой, имеют между собой одно и то же отношение, если при последовательном взаимном вычитании второй величин в обеих парах на каждом шаге будут получаться одни и те же неполные частные.

2.5.2 Обобщённый алгоритм Евклида для многочленов.

Рассмотрим пример использования алгоритма Евклида для многочленов.

Найдём наибольший общий делитель многочленов

$$A = x^3 + 3x^2 + 3x + 2 \text{ и } B = x^3 + 2x^2 + 2x + 1.$$

Применим алгоритм Евклида:

	$x^3 + 3x^2 + 3x + 2$	$x^3 + 2x^2 + 2x + 1$
--	-----------------------	-----------------------

	x^3+2x^2+2x+1		1	
	$\underline{x^3+2x^2+2x+1}$	x^2+x+1		
	$x^3+ x^2+ x$	$x+1$		
	$\underline{x^2+x+1}$			
	x^2+x+1			
	0			

2.5.3 Ускоренные версии алгоритма.

Одним из методов ускорения целочисленного алгоритма Евклида является использование симметричного остатка:

$$r_i \equiv r_{i-2} \pmod{r_{i-1}},$$

где

$$-\frac{r_{i-1}}{2} \leq r_i \leq \frac{r_{i-1}}{2}.$$

Одна из наиболее многообещающих версий ускоренного алгоритма Евклида для полиномов основывается на том, что промежуточные значения алгоритма в основном зависят от высоких степеней. Применение стратегии Разделяй и Властвуй позволяет уменьшить асимптотическую сложность алгоритма.

2.6. Применение теории делимости.

2.6.1 Разложение на множители.

$f(x):(x-1/2)$ Разделим.

$2x^3+7x^2-28x+12$	$x-1/2$
$2x^3-x^2$	$2x^2+8x-24$
$8x^2-28x$	
$8x^2-4x$	
$-24x+12$	
$-24x+12$	
0	
$x=-6$	
$x=2$	

Значит $2x^2+8x-24=0$, т.е. $x^2+4x-12=0$

Ответ: $2(x-1/2)(x+6)(x-2)$

2.6.2 Сокращение дробей.

$\frac{2x^3+7x^2-28x+12}{x-1/2}$	=	$\frac{2(x-1/2)(x+6)(x-2)}{(x-1/2)}$	=	$\frac{2(x+6)(x-2)}{1}$	=	$2(x+6)(x-2)$
----------------------------------	---	--------------------------------------	---	-------------------------	---	---------------

Ответ: $2(x+6)(x-2)$

2.6.3 Решение уравнений.

1) $2x^2-3x-5=0$; $f(x)=2x^2-3x-5$

$$f(-1)=2(-1)^2-3(-1)-5=0, \text{ значит}$$

$$f(x):(x+1) \text{ (:-символ кратности).}$$

Разделим уголком: а) $2x^2:(x)=2x$ поставим под уголок

$2x^2-3x-5$	$x+1$		Умножим $2x$ на $(x+1)$
	$2x$		

б) $2x(x+1)=2x^2+2x$ подставим под выражением $2x^2-3x-5$.

$2x^2-3x-5$	$x+1$
$2x^2+2x$	$2x$

в) Вычтем $(2x^2-3x-5)-(2x^2+2x)=-5x-5$

$2x^2-3x-5$		$x+1$
$2x^2+2x$		$2x$
	$-5x-5$	

г) $(-5x):x=-5$

$2x^2-3x-5$	$x+1$
$2x^2+2x$	$2x-5$
	$-5x-5$

д) $-5*(x+1)=-5x-5$. Подставим под $-5x-5$

$2x^2-3x-5$	$x+1$
$2x^2+2x$	$2x-5$
	$-5x-5$
	$-5x-5$

е) $(-5x-5)-(-5x-5)=0$, значит, остаток равен нулю.

$2x^2-3x-5$	$x+1$
$2x^2+2x$	$2x-5$
	$-5x-5$
	$-5x-5$
	0
	$x+1=0$
	$2x-5=0$

Процесс деления закончен.

Ответ: $\{-1; 2,5\}$

2.6.4 Теорема Безу

Теорема. Остаток от деления многочлена $f(x)$ на многочлен $x - a$ равен $f(a)$.

Доказательство. Степень остатка меньше 1, следовательно, остаток — константа. Пусть r — остаток.

$$f(x) = (x - a)q(x) + r(x).$$

Это равенство верно при любых значениях x . Положим $x = a$:

$$f(a) = (a - a)q(a) + r,$$

$$f(a) = r.$$

Задачи.

1) Проверьте, выполняются ли условия:

а) $x^3 + 3x^2 + 5x - 9$ делится на $x - 1$;

б) $x^{10} + 3x^4 + 7x^2 + 5$ делится на $x^2 + 1$.

2) Докажите, что

$x^3 - 2x^2 - 5x + 6$ делится на $x^2 - 4x + 3$.

3) Найдите значения параметров a и b , при которых

$x^3 + 3ax^2 + bx + 2$ делится на $x^2 - 1$.

4) Найдите все значения параметров a и b , такие, что остаток от деления

$x^3 + 3(a + b)x^2 + (b - a)x + 2$ на $x^2 - 1$ равен $2x + 3$.

5) Найдите все натуральные n , такие, что

$(x^2 + x - 1)^n - (x^2 - x + 1)^{3n}$ делится на $x^2 - x$.

6) Известно, что остаток от деления полинома $P(x)$ на $x - 1$ равен 2, от деления $P(x)$ на $x - 3$ равен 1. Найдите остаток от деления $P(x)$ на $x^2 - 4x + 3$.

7) Найдите остаток от деления многочлена $P(x) = x^{100} - 2x^{51} + 1$ на $Q(x) = x^2 - 1$.

ПРИЛОЖЕНИЯ

Урок 1. Тема: Сложение и вычитание многочленов.

Цели урока: повторить и закрепить понятия одночленов и многочленов; вспомнить правила раскрытия скобок и приведения подобных слагаемых; объяснить правила сложения и вычитания многочленов. В течение урока развивать у учеников умение складывать и вычитать многочлены.

Ход урока:

1. Организационный момент. (2 мин.)

2. Актуализация знаний. (13 мин.)

1) Опросить учащихся определения одночлена и многочлена, их свойства. Для заданных выражений устно перечислить все члены многочленов и указать коэффициенты членов, содержащих буквенные множители:

$$5x^5 - 3x^2 - x + 4; \quad -3x^2y + 4xy - 1; \quad 8n^3 + 7n^2 - 6n + 5;$$
$$2b^2c - 2bc^2 + b - c; \quad 3a^4 - 12a^3 + a.$$

2) Прочитать данные многочлены по убыванию степени переменной:

$$2x^2 + 15x^5 + 71x - 4x^4; \quad 3x - 10x^4 + 2 - 66x^3;$$
$$2a - 42a^2 - 28a^3 - 1.$$

3) На доске выполнить с полным объяснением задание № 729.

3. Новый материал. (5 мин.)

Сначала вспомнить правила раскрытия скобок, если перед скобкой стоят знаки плюс или минус, так же повторить определение подобных одночленов. А после к доске вызвать ученика, который самостоятельно, без помощи учителя, попробует сложить и вычесть данные многочлены:

$3x^2 - 5x$ и $2x^2 + 4$. Затем совместными усилиями выводится правило сложения и вычитания многочленов.

4. Закрепление нового материала. (20 мин.)

Решить задания № 736, 739, 742, 744, 745 и 747.

5. Итоги урока. (3 мин.)

6. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.2.

Решить задания № 737, 740 и 746.

Урок 2. Тема: Сложение и вычитание многочленов.

Цели урока: повторить и закрепить понятия одночленов и многочленов, правила сложения и вычитание многочленов; проверить умение учеников приводить многочлены к стандартному виду. В течение урока развивать у учеников умение складывать и вычитать многочлены, упрощать выражения.

Ход урока:

1. Организационный момент. (2 мин.)

2. Индивидуальная работа. (8 мин.)

К доске вызывается четыре ученика, для выполнения заданий по карточка. Кроме письменного выполнения, требуется еще умение объяснить по правилам ход действий.

Карточка 1. Упростить: $2x^2y - 4y^2x + 5yx^2 - 7x^2y$; $(5ab - 6b) + (8b - 4ba)$; $(7ab - 3a^2b) - (8ab^2 + 6ab)$.	Карточка 2. Упростить: $6xy - 7x^2 + 8x - 9yx$; $(3x - 5y) + (6x + 9y)$; $(15xy + 27y) - (8x + 31xy)$.
Карточка 3. Упростить: $9ac - 4ab - 7bc - 17ca$; $(81ac + 4ab) + (41ab - 64ac)$; $(73a - 4b) - (10b - 50a)$.	Карточка 4. Упростить: $67m^2n + 11m^2n - 28nm^2 + n^2m$; $(3xy^2 + 7xy^2) + (46xy^2 - 31y^2x)$; $(25ab + 42ac) - 6 - (68ca - 19ab)$.

В момент выполнения заданий на доске преподаватель проводит фронтальный опрос по определениям и правилам данной темы. Затем проверяется правильность заданий и еще раз проговариваются правила, но теперь те ученики, которые выполняли задания.

3. Актуализация знаний. (10 мин.)

Закрепить правила при решении на доске задания № 726, 734 и 743. Если у кого-нибудь из учеников возникают проблемы, учитель еще раз объясняет у доски.

4. Решение задач. (20 мин.)

Разобрать задание № 749. Затем преподаватель объясняет, как многочлен представить суммой или разность и решаются примеры № 750, 751, 753, 755.

5. Итоги урока. (3 мин.)

6. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.2.
Решить задания № 738, 741 и 757.

Урок 3. Тема: Умножение одночлена на многочлен.

Цели урока: повторить и закрепить понятия одночленов и многочленов, правила сложения и вычитания многочленов; объяснить правило умножения одночлена на многочлен. В течение урока развивать у учеников умение умножать одночлен на многочлен.

Ход урока:

1. Организационный момент. (2 мин.)

2. Актуализация знаний. (8 мин.)

Устно разобрать задание № 748, при этом вспомнить правила, помогающие выполнить это упражнение. Затем разобрать задание № 752.

3. Устная работа. (5 мин.)

Раскрыть скобки в выражения (приготовить или на карточках или на доске):

$$2(a + b); \quad -4(a + b); \quad 3(2a - 3); \quad -7(a + 4b); \quad 5(10a - 8b).$$

Вспомнить название свойства, при помощи которого раскрывались скобки, а затем и правило, выражающее это свойство.

4. Новый материал. (5 мин.)

Предложить ученикам самостоятельно найти произведение

$3a \cdot (4a - 7b)$. Обсудить этот пример и совместными усилиями вывести правило умножения одночленов на многочлены. Если ученикам тяжело выполнять умножение, то вводится понятие «фонтанчик». «Фонтанчик» идет от одночлена к каждому члену многочлена.

5. Закрепление нового материала. (20 мин.)

1) Рассмотреть задания № 764, 766, 771 и 767.

2) Так же умножить данный одночлен $5x^2y$ на каждый из многочленов, в данном задании происходит постепенное усложнение произведения:

а) $7 - x$; б) $2x - 9y$; в) $3x^3 + 1$;

г) $x^4y^2 - 5y^7$; д) $4xy^5 + 3x^7y^5 - 1,2x^8$.

3) Вспомнить правила решения уравнений со знаменателями:

а) $\frac{x}{5} - \frac{x}{2} = -3$; б) $\frac{x-4}{4} - 2 = \frac{x}{2}$.

6. Итоги урока. (3 мин.)

7. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.3.

Решить задания № 765, 768 и 770.

Урок 4. Тема: Умножение одночлена на многочлен.

Цели урока: повторить и закрепить понятия одночленов и многочленов, правила сложения и вычитания многочленов, правило умножения одночлена на многочлен; проверить знания и умения учащихся, по данной теме. В течение урока развивать у учеников умение складывать и вычитать многочлены, умножать одночлены на многочлены.

Ход урока:

1. Организационный момент. (2 мин.)

2. Актуализация знаний. (8 мин.)

На доске упростить следующие выражения:

- $(3ab - 4a^2b) + (5ab - 2ab^2)$;
- $(7x^2y + 2xy^2) - (xy^2 + 3x^2y)$;
- $2a(3 - 4a) - 3a(5a - 2)$;
- $4ab(2 - 3a) + 7a(6a + b)$;
- $12y(1 - 5y) - (20y^2 + 17y)$.

При выполнении вспомнить и повторить все правила по теме.

3. Самостоятельная работа. (30 мин.)

Вариант 1.	Вариант 2.
№ 724(а), 731, 733(а, в), 754(а, в), 772(а, в).	№ 724(б), 732, 733(б, г), 754(б, г), 772(б, г).

Дополнительно для сильных учеников предлагается выполнить задание № 773.

4. Итоги урока. (3 мин.)

5. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.3.

Решить задания № 756 и 775.

Урок 5. Тема: Умножение одночлена на многочлен.

Цели урока: повторить и закрепить правила сложения и вычитания многочленов, правило умножения одночлена на многочлен; рассмотреть решение задачи – шутки для развития у детей интереса к математике. В течение урока развивать у учеников умение складывать и вычитать многочлены, умножать одночлены на многочлены.

Ход урока:

1. Организационный момент. (2 мин.)

2. Анализ самостоятельной работы. (3 мин.)

Выставить оценки и разобрать задания из работы, которые при решении вызвали затруднения.

3. Актуализация знаний. (12 мин.)

Вспомнить и закрепить сложение и вычитание многочленов, упростить следующие выражения:

- $(a^2 - 0,45a + 1,2) + (0,8a^2 - 1,2a) - (1,6a^2 - 2a)$;
- $(y^2 - 1,75y - 3,2) - (0,3y^2 + 4) - (2y - 7,2)$;
- $6xy - 2x^2 - (3xy + 4x^2 + 1) - (-xy - 2x^2 - 1)$;
- $-(2ab^2 - ab + b) + 3ab^2 - 4b - (5ab - ab^2)$.

Вспомнить закрепить умножение одночлена на многочлен, выполнить задание № 769.

4. Решение задач. (25 мин.)

1) Рассмотреть задания № 773, 774(б), 776(б), 777, 778(б), 779.

2) Решить уравнение $\frac{x}{3} - \frac{x+8}{6} = \frac{3x+2}{9} - \frac{x+11}{6}$.

3) Степа получил по математике двойку. Папа, узнав об этом, 15 минут бегал за Степой с ремнем, а Степа убегал со скоростью a метров в секунду. Потом папа схватился за сердце и сев в кресло, стал ждать маму. Когда пришла мама, Степе пришлось скорость своего бега увеличить на 2 метра в секунду, но у мамы силы иссякли за 6 минут. Сколько метров пришлось бегать Степе от своих разгневанных родителей?

5. Итоги урока. (3 мин.)

6. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.3.

Решить задания № 774(а), 776(а) и 778(б, в).

Урок 6. Тема: Умножение многочлена на многочлен.

Цели урока: повторить и закрепить правила умножения одночлена на многочлен; объяснить правила умножения многочленов. В течение урока развивать у учеников умение умножать одночлены на многочлены, находить произведение многочленов.

Ход урока:

1. Организационный момент. (2 мин.)

2. Устная работа. (5 мин.)

Найти произведения:

$$5(2a - 3b); 4a(5a + 6b); -7c(3a - c); ab(4 - a);$$

$$3x^2(xy - x^2); -m^3(3m^4 - 2mn); -5mn(6m^2n + 3n^3).$$

3. Новый материал. (13 мин.)

Учитель объясняет правило умножения многочленов, показывает на доске примеры выполнения:

$$(a + 1)(2a - 3) = 2a^2 - 3a + 2a - 3 = 2a^2 - a - 3;$$

$$(x^2 - y)(x^2 - 4y) = x^4 - 4x^2y - x^2y + 4y^2 = x^4 - 5x^2y + 4y^2.$$

Для упрощения показывается «фонтанчик».

4. Закрепление нового материала. (20 мин.)

1) Рассмотреть задания № 781, 783 и 784.

2) Так же умножить данный многочлен $5x^2 + 3x$ на каждый из многочленов, в данном задании происходит постепенное усложнение произведения:

а) $7 - x$; б) $3x^3 + 2x$; в) $5x^6 + 3x^2 - 4x$.

5. Итоги урока. (3 мин.)

6. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.4.

Решить задания № 780 и 785.

Урок 7. Тема: Умножение многочлена на многочлен.

Цели урока: повторить и закрепить правила умножения одночлена на многочлен, умножение многочленов; проверить умение учеников находить произведение многочленов. В течение урока развивать у учеников умение находить произведение многочленов.

Ход урока:

1. Организационный момент. (2 мин.)

2. Индивидуальная работа. (8 мин.)

Четверо учеников выходят к доске для работы с карточками.

Карточка 1. Выполнить умножение: $(a + b)(x + y);$ $(a - 4)(a + 8).$	Карточка 2. Выполнить умножение: $(a - b)(x + y);$ $(b - 3)(5 - b).$
Карточка 3. Выполнить умножение: $(a + b)(x - y);$ $(2x - 7)(5 + x).$	Карточка 4. Выполнить умножение: $(a - b)(x - y);$ $(1 - 4y)(9 + 5y).$

Остальные ученики выполняют задание № 787.

Сначала учащиеся проверяют правильность примеров на доске, а затем у доски рассматриваются примеры № 787.

3. Актуализация знаний. (10 мин.)

Найти произведение следующих выражений:

$(x^2 + 3y)(x - y^2); (a^2 - 5b)(a^2 + 5b^3);$
 $(4m^2 + 7n^2)(2n^2 - 8m^2); (7x^2 - 4x)(1 - 5x);$
 $(c - 3)(4c^3 - 8c^2); (5x^2 - 7y)(3x^2 + y);$
 $(6 - 3x^3)(6x^3 + 1); (5b^2 - 7b^4)(9b^2 - 1).$

Лучше, если кто-то из учеников будет работать у доски: записать пример, закрыть от класса доску и решить, затем доска открывается и пример проверяется. У каждого ребенка будет возможность проверить себя.

4. Решение задач. (10 мин.)

Рассмотреть задание № 786.

5. Самостоятельная работа. (10 мин.)

Вариант 1.	Вариант 2.
-------------------	-------------------

Найти данные произведения:	
1) $(9 - a)(3a + 5)$;	1) $(8 + 3x)(7x + 1)$;
2) $(3x - 7y)(4y - x)$;	2) $(5a + 6c)(c - 9a)$;
3) $(9b + 1)(2b^2 + 7b^3)$;	3) $(2x - 5x^2)(4 - 7x)$;
4) $(c - 8)^2$;	4) $(6 + b)^2$;
5) $(3m + 11n)(6n^2 + 2mn - m^2)$.	5) $(12m - n)(4m^2 - 3mn + 5m^2)$.

5. Итоги урока. (3 мин.)

6. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.4.

Решить задания № 782 и 791.

Урок 8. Тема: Умножение многочлена на многочлен.

Цели урока: повторить и закрепить правила выполнения действий между одночленами и многочленами. В течение урока развивать у учеников умение упрощать выражения с одночленами и многочленами, использовать свои знания для выполнения разных видов заданий; так же развивается чувство сотрудничества и сопереживания.

Ход урока:

1. Организационный момент. (2 мин.)

Рассадить детей на 5 групп. Рассказать тему урока и задачи урока.

2. Анализ самостоятельной работы. (5 мин.)

Выставить оценки и разобрать задания из работы, которые при решении вызвали затруднения. Так же проводится фронтальный опрос всего класса по определения и правилам.

- Что называется одночленом?
- Что называется многочленом?
- Какие слагаемые являются подобными?
- Что такое стандартный вид одночлена? многочлена?
- Влияет ли коэффициент на подобия слагаемых?
- Что значит сложить подобные слагаемые?
- Расскажите правило раскрытия скобок, если перед скобкой знак плюс.
- Если перед скобкой знак минус.
- Если перед скобкой множитель.
- Расскажите правило умножения многочленов.

3. Актуализация знаний. (13 мин.)

На столе преподавателя лежат три стопки карточек. В первой карточки с выражениями одночленов:

$5x^2y^3$; $-7xy^7$; $3x^2y^2$; $6x^3y$; $-x^3y^2$; $-4x^4y$;

Во второй и третьей карточки с выражениями многочленов:

$(x + y)$; $x^2 - 3y$; $2x + 5y$; $5y - 7x$;

$6x - y$; $x - y$; $9x - y^2$; $2x^2 + 3y^2$;

$5y + 8x$; $2x^2 + y$; $7y^2 - 4x^2$; $4y^2 + 9x$;

В каждой стопке по шесть карточек. Из каждой группы выходит по очереди по одному ученику и для своей группы берет по карточке из каждой стопки. И с данными выражениями группа должна выполнить следующие упражнения:

1. Умножить одночлен на многочлен.
2. Сложить многочлены.
3. Из первого многочлена вычесть второй.
4. Умножить многочлен на многочлен.

Примеры записываются и решаются. Примеры должны быть записаны у каждого в группе и решение должен уметь объяснить каждый из группы.

После выполнения задания, учитель из каждой группы вызывает по одному ученику. Этот ученик должен записать примеры на доске и затем объяснить

решение. Пока задания записываются, остальные устно выполняют и разбирают задание № 796.

4. Решение задач. (13 + 7 мин.)

1) После каждой группе выдается карточка с выражениями для упрощения. Карточки у всех одинаковые. Учащиеся должны обсудить в группе, каким образом выполнять действия и внимательно решить.

1. $(2x - y)(y + 4x) + 2x(y - 3x)$;
2. $(3a - 2b)(2a - 3b) - 6a(a - b)$;
3. $2c(b + 15c) + (b - 6c)(5c + 2b)$;
4. $(3m - 1)(2m + 5) - 6m(m - 2)$;
5. $5m(2n - m) - (8m - n)(2n - m)$.

После выполнения устно проверяются ответы. Каждая группа читает по одному ответу. Если в каком-нибудь примере возникли трудности, то пример обсуждается на доске. (Обратить внимание на пятый пример.)

2) Затем каждой группе раздается по карточке с заданием на доказательство. Если ученики не могут справиться, преподаватель чуть-чуть должен подсказать.

Карточка 1. Докажите, что значение выражения не зависит от переменной x : $(x - 6)(x + 8) - (x + 4)(x - 1)$.
Карточка 2. Докажите, что значение выражения не зависит от переменной x : $x^4 - (x^2 - 7)(x^2 + 7)$.
Карточка 3. Докажите, что при всех целых n значение выражения делится на 6: $n(n - 1) - (n + 3)(n + 2)$.
Карточка 4. Докажите, что при всех целых n значение выражения делится на 7: $n(n + 2) - (n - 7)(n - 5)$.
Карточка 5. Докажите, что при всех натуральных n значение выражения делится на 6: $n(n + 5) - (n - 3)(n + 2)$.

Каждая задача показывается и обсуждается у доски.

5. Итоги урока. (3 мин.)

6. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.4.
Решить задание № 789.

Урок 9. Тема: Умножение многочлена на многочлен.

Цели урока: повторить и закрепить правила выполнения действий с одночленами и многочленами; рассмотреть решение заданий на данную тему повышенной трудности; проверить знания учеников по упрощению выражений, содержащих одночлены и многочлены. В течение урока развивать у учеников умение упрощать выражения с одночленами и многочленами, умение использовать свои знания.

Ход урока:

1. Организационный момент. (2 мин.)

2. Актуализация знаний. (10 мин.)

Вспомнить и повторить все правила, необходимые для упрощения выражений. Выполнить на доске, с полным объяснением номер № 788.

3. Решение задач. (18 мин.)

Рассмотреть усложненные задания на данную тему.

1) Докажите, что при всех натуральных n , больших 2, значение выражения делится на 12: $(n - 1)(n + 1) - (n - 7)(n - 5)$.

2) Представить в виде многочлена выражение $(a + b)(a - b + 1) - (a - b)(a + b - 1)$.

3) Рассмотреть задания № 792, 794, 795(а, в).

4. Самостоятельная работа. (10 мин.)

Вариант 1.	Вариант 2.
Упростить выражения:	
1) $(5 + 6y)(7 - 3y)$;	1) $(5x - 4)(8 - x)$;
2) $(3a - 2)(5 - 2a) + 6a^2$;	2) $(7y - 4)(2y + 3) - 13y$;
3) $-3b^3(b + 2)(1 - b)$;	3) $2a^2(a - 1)(3 - a)$;
4) $(4n^2 - 6nm + 9m^2)(2n + 3m)$;	4) $(25n^2 + 10nm + 4m^2)(5n - 2m)$;
5) $(x + 1)(x + 2)(x + 3)$.	5) $(b - 1)(b - 4)(b + 5)$.

5. Итоги урока. (3 мин.)

6. Домашнее задание. (2 мин.)

Прочитать, разобрать и выучить правила из § 7.4.

Решить задания № 793 и 798.

ВЫВОДЫ

В теории делимости многочленов изучают признаки делимости одного многочлена на другой. Теория делимости многочленов предлагает математический аппарат для описания этих законов. Этот математический аппарат является таким же логически строгим и точным, как математический аппарат в других разделах математики. Рассмотренные понятия позволяют дать определение теории делимости многочленов: теория делимости многочленов - это математическая наука, изучающая деление одного многочлена на другой.

Данная работа помогает разобраться в сущности теории делимости многочленов, научиться решать с помощью нее математические уравнения, понять в каких областях она может применяться. Материалы работы можно использовать на математических кружках, спецкурсах, факультативах.

В данной работе рассмотрены всевозможные программно-педагогические средства, в том числе и приложения, разработанные в среде MS Visual Studio. Рассмотренная в работе программа «Действие над многочленами», является вспомогательной программой и служит для проверки и составления заданий различной степени сложности. Данная программа подходит как для учеников, для проверки правильности результатов выполненного задания, так как программа выдает только результат вычисления, а алгоритм решения остается за учеником. Так и для преподавателей, облегчающий составление заданий разной степени сложности, для учеников с разным уровнем подготовки, что в свою очередь предоставляет возможность своеобразного индивидуального подхода к оцениванию учащихся разного уровня.

Эта программа не требует от пользователя знаний программирования, обладает интуитивно понятным интерфейсом, и минимальным набором функций, что дает возможность ее использования только в узком направлении, при этом не перегружая систему и не вызывая зависания даже самых слабых компьютеров.

В настоящее время использование ИКТ и ППС является неотъемлемой частью процессов обучения и преподавания, не является исключением и преподавание математики. Как известно, математика является одной из наиболее близких дисциплин школьного курса к информатике, не даром большинство программных алгоритмов составлены на основе алгоритмов математических, что в свою очередь, лежит и в основе программных структур. Учащиеся видят перед собой пример реального применения своих знаний и их рациональное решение, проявят наибольший интерес к изучению предмета. Конечно, неукоснительным условием является овладение учеником объемом знаний школьного курса математики и умения применять эти знания на практике, только в этом случае использования ИКТ принесет положительные результаты.

Безусловно, применения программных педагогических средств играет огромную роль в овладении знаний и умений, при этом важно умение педагога донести до учеников предназначение программы как инструмента в достижении определенных результатов, и правильность его использования.

Выводом всей проделанной работы может быть пожелание использовать программно-педагогические средства, как вспомогательное и очень эффективное средство в привлечении интереса учащихся к математике, связь ее с информатикой и даст возможность преподавателю несколько облегчить свой труд.

СПИСОК ЛИТЕРАТУРЫ

1. Послание Президента Республики Узбекистан Шавката Мирзиёева Олий Мажлису от 24 января 2020 года.

2. Государственный образовательный стандарт общего среднего образования, приложение № 1 к Постановлению Кабинета Министров РУз от 6 апреля 2017 года № 187.

3. Постановление Кабинета Министров РУз «О разработке и введении государственных образовательных стандартов для системы непрерывного образования» от 5 января 1998 г. № 5.

4. Указ Президента Республики Узбекистан от 29. 04. 2019 г. № УП-5712 "Об утверждении Концепции развития системы народного образования Республики Узбекистан до 2030 года"

5. Гаюбова, К. А. Использование новых педагогических технологий в преподавании узбекского языка / Молодой ученый. — 2015. — № 2 (82)

6. Байбулатова Г. К. разработка «Новые государственные стандарты в Узбекистане»

7. Ш. А. Алимов, А. Р. Холмухамедов, М. А. Мирзаахмедов Алгебра Учебник для 7 классов школ общего среднего образования Издание четвертое переработанное и дополненное ИПТД „O‘qituvchi“ Ташкент 2017.

8. Мирзаахмедов М. А. , Исмаилов Ш. Н. , Аманов А. К. Математика 11 Алгебра и начала анализа, геометрия часть 1 Учебник для 11 классов средних образовательных учреждений среднего специального, профессионального образования Ташкент 2018 г.

9. Алгебра и теория чисел. Под ред. Н. Я. Виленкина. Москва: Просвещение, 1984.

10. Использование педагогических программных средств в образовательной среде школы для формирования универсальных учебных действий Сборник методических рекомендаций Санкт-Петербург 2016.

11. Архангельский А. Я. Delphi 7. Справочное пособие. Москва: ООО Бином-Пресс, 2004.

12. Попов В. Б. Turbo Pascal. Москва: Финансы и статистика, 2000.

13. Вабищевич, С. В. Технология специальной методической подготовки будущих учителей информатики к осуществлению компьютерного обучения / С. В. Вабищевич // Весці БДПУ. Серыя 3. Фізіка. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – 2010.

14. <http://www.sura.ru/Maxwell/> — Математика, алгоритмы и программирование.

Рецензия

на выпускную квалификационную работу Чепухалина С. А.,

по теме: «Методика применения информационных технологий

в изучении многочленов в общеобразовательных школах».

параграфов отвечают их содержанию. Литература отвечает теме работы. Из рецензируемой работы видно, что Чепухалин С. А. способен вести научную и методическую исследовательскую работу.

Ташкентский Государственный
Педагогический Университет
им. Н. Г. Гусева.

Физико-математический факультет.

Студент направления

"5Н0100 - Методика преподавания
математики".

Рецензия

на выпускную квалификационную
работу студента

Чепуханина Сергея Анатольевича.

Тема работы: Методика применения
информационных технологий
в изучении многоугольников в
образовательных школах.

Значительной частью теоретической
и практической алгебры школьного
курса является изучение многоугольников.
Актуальность темы заключается в
том, что среди многообразных современ-
ных методов преподавания главным
можно считать, — применение ИКТ.
Применение в образовании современных
программ позволяет использовать их
возможности для решения широкого
круга задач.

Из-за огромного количества информации
появляется и множество проблем,
важнейшая из которых — проблема
обучения.

Особый интерес вызывают вопросы, связанные с применением информационных технологий в образовании, так как удовлетворить запросы учащихся в современной мире становится всё труднее.

Самое доступное средство - компьютер. Преимущества компьютеризации:

- упрощается процедура создания электронных учебников и методических пособий;
- автоматизация, хранение данных в любой необходимой форме;
- работа с неограниченным объемом данных;
- изучение сети Интернет, информационные технологии повышают кругозор знаний учащихся;
- повышается интерес к обучению;
- развивается ответственность и самостоятельность.

Работа включает в себя введение, две главы, заключение, список использованной литературы.

В первой главе, "Microsoft Visual Studio" как педагогическое программное средство рассмотрены понятия "педагогическое программное средство"; типы ППС, требования к её разработке, описание среды разработки Microsoft Visual Studio, язык программирования C#, интерфейс Microsoft Visual Studio.

Во второй главе "Теория деятельности многоголосов" приведены основные понятия "одноголос", "многоголос", действия над многоголосами: сложение, вычитание, умножение, деление; применение теории деятельности многоголосов и их реализация в программе, разработанной в среде MS Visual Studio.

В заключение даются выводы по проделанной работе и её значимость. Достоинством работы в том, что студент чётко, последовательно излагает суть тем, логически обосновывая каждое положение.

Рассмотрение типов ППС, классификацию по функциям, по степени активности учащихся, по уровню коммуникативности, роль информационных технологий в образовании, их преимущественные стороны показывает эрудицию, оригинальное мышление, знание математической литературы, высокий уровень теоретической подготовки студента.

Особое внимание заслуживает описание уникальных возможных средств новых информационных технологий (СНИТ); перечисляются сферы использования СНИТ: сеть Интернет, образовательные центры интернета (дистанционное образование), использование средств мультимедиа, электронные учебники в образовании, язык HTML, описание среды MS Visual Studio, индивидуальный стиль кодирования, язык программирования C++ (с++), интерфейс MS Visual Studio.

Вопрос, которые были рассмотрены во второй главе:
многочлен, стандартный вид многочлена, действия с многочленами, делимость, алгоритм Евклида (исторические сведения), обобщенный алгоритм Евклида для многочленов, разложение на множители, сокращение дробей, решение уравнений, имеют практическое значение и могут быть использованы в работе учителя в образовательной школе.

Недостатки в работе не обнаружены. Рекомендую расширить задачи, рассматриваемые во второй главе (их количество)

Общий вывод:

Данная работа помогает разобраться в теории делимости многочленов, применить эту теорию при решении уравнений, сокращении дробей.

Программа "Действия над многочленами" является вспомогательной и служит для составления заданий различной степени сложности, не требует знания программирования, обладает минимальным набором функций.

Квалификационная работа соответствует требованиям, предъявляемым к данному виду работ; заслуживает высокой оценки.

Рецензент: Тиенко Наталья Анатольевна,
учитель математики
школы № 145
Лицея Баевского р-на
20 мая 2020 г. Ташкента.

Заключение

научного руководителя на выпускную квалификационную работу студента 4-курса физико-математического факультета Чепухалина С. А. на тему "Методика применения информационных технологий в изучении многочленов в общеобразовательных школах"

Отличительной особенностью данной работы является, то, что она рассматривает два аспекта, во взаимосвязи методики преподавания алгебры и теории делимости многочленов, в контексте неразрывно друг от друга. Большинство работ рассматривают либо методику преподавания, либо непосредственно математическую теорию не всегда учитывая, как они впоследствии будут взаимодействовать.

Актуальностью работы является то, что в настоящее время информационные технологии развиваются очень быстро и теория преподавания не всегда поспевает за прогрессом. В то время как учащиеся осваивают информационные технологии с малых лет и в их обучении нельзя ограничиваться только методами «старой школы», а чаще применять инновационные методы и заинтересовать учащихся изучать столь важный и нужный предмет как математика.

Тема «Теория делимости многочленом» неслучайно выбрана как основа для применения информационных технологий. Эта теория применяется не только на протяжении всего школьного курса алгебры, но и в дальнейшем и в других направлениях математики и не только. Многочлены играют весомую роль в математике. С помощью теории многочленов, учащиеся могут посмотреть с определенной позиции на многие математические задачи, успешно решить сложные уравнения и неравенства, установить связь между математикой и прикладной математикой.

В алгебре мы встречаемся с многочленами при решении уравнений и неравенств. Будь то линейные, квадратные, уравнения и неравенства высших степеней, рациональные и иррациональные выражения, интегральное и дифференциальное исчисления. В геометрии, при составлении формул и нахождения периметров и площадей, значение элементов плоских фигур, нахождения площадей поверхности и объемов фигур пространственных, нахождений координат и длин векторов в плоскости и пространстве. В физике, экономике, химии, биологии, везде где требуются математические вычисления, работа с формулами, везде мы сталкиваемся с многочленами, даже не задумываясь об этом, потому что мы знаем их с 7 класса, как не замечаем того как пользуемся буквами при составлении слов и используем арифметические действия знакомые нам с начальной школы.

Очень удачно построена структура работы. Данная работа состоит из введения, двух глав, приложения, выводов, списка использованной литературы. Во введении сформулированы основные характеристики

исследования: актуальность, объект, предмет, цель, задачи и методы исследования.

В первой главе «Microsoft Visual Studio как педагогическое программное средство» рассмотрены понятия «Педагогическое программное средство» (ППС). Приведены высказывания, как отечественных, так и зарубежных педагогов-специалистов о достоинствах и недостатках применения ППС в преподавании различных дисциплин, приведена классификация ППС. Рассмотрены наиболее часто применяемые ППС в общеобразовательных школах, подробно рассмотрены возможности и интерфейс Microsoft Visual Studio, синтаксис языка программирования C# (Си шарт), как наиболее «продвинутого» языка программирования семейства Си (следующий после C++), произведено сравнения сред разработки Microsoft Visual Studio и Delphi.

Во второй главе «Теория делимости многочленов» приведены теоретические основы, введены понятия «одночлен», «многочлен», описаны правила сложения, вычитания, умножения одночлена на многочлен и многочлена на многочлен, деления многочлена на многочлен, с остатком и без остатка. А также описан интерфейс программы, разработанный в среде Microsoft Visual Studio, описаны правила записи многочленов в текстовые поля программы, назначения кнопок и переходы между окнами программы, немного затронута программная часть, объясняется, как реализована та или иная функция.

В приложении представлены разработки 9 уроков по действиям над многочленами для 7 класса. В заключении приведены основные выводы по данной работе, также представлен список литературы, состоящий из 14 источников.

Считаю, что выпускная квалификационная работа отвечает всем требованиям, предъявляемым к таким видам работ и ее можно допустить к защите.

Кандидат физико-математических наук, доцент кафедры

«Математика и методика ее преподавания»



Ш. Н. Исмаилов

1