

## ВВЕДЕНИЕ

С развитием первой компьютерной техники появился специальный машинный язык, при использовании которого программист мог бы применить команды, оперируя с разнообразными ячейками памяти, полностью используя различные возможности машины.

Но частое использование большинства современных персональных компьютеров (ПК) только на уровне машинного языка будет очень затруднительно.

Поэтому от использования в таком его варианте пришлось отказаться. На современном этапе появилось целое множество мощных языков программирования (ЯП), которые позволяют выполнять сложнейшие функции.

В дальнейшем, при развитии высокоуровневых языков программирования, со временем появилась прекрасная возможность как-то адаптировать обработку информации при использовании созданных программных продуктов, являющихся очень удобными, поскольку опытные программисты могут создавать программы прямо для конкретного предметного сектора. Вместе с этим широкой популярностью озаменовались программы с графическим интерфейсом.

Актуальными в наше время является высокая эффективность и надобность применения программ с графическим или же оконным интерфейсом, которые написаны на современных языках программирования высокого уровня.

Целью работы является подробное описание популярных языков для программирования и создания с их помощью оконного графического приложения.

Смотря на поставленную цель надо решить задачи:

– рассмотреть современные теоретические источники литературы;

- подробно описать классификацию ЯП;
- детально раскрыть основные понятия ЯП Java, C#, C++;
- охарактеризовать интегрированную среду по разработке программ C++ Builder;
- создать оконное приложение на языке программирования высокого уровня.

Объектом исследования являются языки программирования.

Предметом исследования являются критерии выбора среды и языка программирования.

Этим занимались талантливые разработчики Б.Стауструп, Ст.Прата, П.Голдинг и другие.

## **1. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ**

### **1.1. Этапы развития языков программирования**

Физические принципы для функционирования электронных устройств таковы, что каждый ПК может воспринимать разные команды, что состоят лишь из определенной последовательности единиц и нулей – машинного кода. На этой стадии такого развития ЭВМ пользователям было необходимо составлять программы на языке понятном ПК. Каждая команда состояла с адресов операндов, кода операций, выраженных в сочетаниях бинарного кода.[5]

Это приводит часто к необходимости найти разные средства, которые позволят еще более просто налаживать корректное общение человека с ПК. И такое средство было найдено: различные ЯП и соответствующие им модули для трансляции (системы программирования).[15]

Языком программирования называется специальный язык, что предназначается для написания команд и управления компьютером. Все современные языки программирования созданы для того, чтоб пользователям было проще выполнять операции с ПК, но они также затем должны транслироваться (интерпретаторами, трансляторами) в машинный код для выполнения компьютером.

ЯП условно можно подразделять на языки высокого и низкого уровня программирования.

Практически всю историю компьютерной индустрии с определенного взгляда легко представить, как становление истории развития ЯП.

Рассмотрим далее историю развития ЯП:

1801 г. – француз И. Жаккард при использовании перфокарт впервые вышивает словосочетание «hello world» на ткани.

1842 г. – первая программист Ада Лавлейс пишет первую свою компьютерную программу. А через почти 160 лет архитекторы по

корпоративным программным продуктам переймут технику Ады при использовании моделей программ, а также назовут этот метод UML.[20]

1936 г. – Алан Тьюринг изобрел специальную теорию языков программирования.

1957 г. – Джон Бакус и корпорация IBM изобретают ЯП под названием Фортран.

1964 г. – Джон Кемни и Том Курц создают на то время революционный ЯП Бейсик.

1970 г. – Н. Вирт создает первый процедурный язык программирования Турбо Паскаль.

1972 г. – Адам Колмеро изобретает современный на то время логический язык Пролог. Сразу начинают его использовать при разработке современных баз данных (БД).

1983 г. – Б. Страуструп на основании языка С, создает новый язык программирования под названием С++.

1990 г. – группа разработчиков: П. Худак, Ф. Водлер, С. Пейтон-Джонс, Э. Катчер создает ЯП Хаскелл.

1996 г. – Дж. Гослинг выполняет проектирование языка Java. Java – довольно типизированный современный объектно-ориентированный ЯП, который описан на основе классов, с современным сборщиком мусора, одиночной диспетчеризацией, одиночными реализациями наследования, а также и расширенным множественным наследованием для любых интерфейсов.

## **1.2. Описание классификации языков программирования**

Процессы работы ПК заключаются в выполнении самых разных инструкций, то есть какого-то перечня определённых команд для нужного порядка. Машинный вид имеющейся информации, который состоит только из 1 и 0, часто указывает, какое именно определенное действие должен выполнять центральный процессор.

Чтобы задать для ПК перечень действий на выполнение, необходимо задать какую-то последовательность с бинарного кода. [5]

Программы, изображенные в специальных машинных кодах, состоят с сотен тысяч команд.

Качественное их написание – занятие утомительное и очень сложное. Программисты должны помнить полностью все сочетания 1 и 0 для бинарного кода для всех команд, а также и двоичные коды адресов, что часто используются при выполнении.

Гораздо проще написать программу для ПК на естественном языке, а работу по переводу его в машинный код поручить ПК.

Языком программирования называется формальная знаковая система, что предназначена для записи компьютерных программ. Языком программирования часто определяется весь синтаксический набор лексических, семантических правил, которые задаются для программ и действий.[12]

Все создатели языков рассматривают понятие языка программирования немного по-своему. К самым современным утверждениям, признаваемым практически большинством программистов, относятся:[16]

– язык программирования предназначен для разработки программных продуктов, которые будут применяться в процессе передачи персональному компьютеру некоторых определенных инструкций для разработки некоторого вычислительного процесса.

– ЯП отличаются также от привычных человеческих языков предназначением для трансляции данных от пользователей к ПК. Можно даже обобщить эти все определения: ЯП – это способ передачи приказов, разных инструкций, четкого руководства для действий, тогда как обычные человеческие языки будут предназначены только для обмена некоторой информацией.

ЯП могут также использовать конструкции по выполнению манипуляций и определению разных структур данных, а также процессов вычислений.

ЯП представляется в виде набора чаще всего инструкций, определяющих его понятие семантики, синтаксиса. Для многих ЯП также применяются международные стандарты. В таком случае специальные организации могут проводить систематические обновление спецификаций, разных определений соответствующего ЯП, продолжать модернизацию и их внедрение.

После создания самых первых вычислительных машин все разработчики создали более 2600 ЯП и число их с каждым годом увеличивается. Некоторыми такими языками программируют также небольшое число разработчиков, а иные становятся сверх популярными.

Нет таких программистов, что не мечтали бы разработать свой собственный язык программирования, который был бы самым простым и

функциональным, самым надежным и быстрым. Именно поэтому и за целую компьютерную предысторию люди придумали много таких ЯП.

На современном этапе перед опытными программистами ставится задача для проектирования систем хранения информации, что ещё 30-40 лет назад казались практически невозможными. Появляются новые технологии, требующие новейших подходов к выполнению процессов программирования, а стремительное развитие Интернета также, в своем роде, предоставляет все новые методы, которые не освоены до конца по созданию новейших технологий программирования.

Всё это служит хорошей почвой при создании новейших ЯП, отвечающих всем задачам для нынешнего времени и также используют новые способы, парадигмы программирования, часто позволяющие решать все актуальные проблемы разработки программного обеспечения с графическим интерфейсом.[7]

Но не смотря на такое огромное количество ЯП, лишь некоторые с них имеют признание программистов.

При определении за 2017 год самых популярных ЯП применим статистические данные компании «ITIBE Software».

По информации корпорации «ITIBE Software» самые популярные языки на сентябрь 2017 года представлены ниже таблице 1:

Таблица 1 – Список популярных языков программирования

№	Название языков программирования	Показатель	Последний релиз языка программирования
1	Java	16.99%	15.10.2014
2	C++	15.06%	13.11.2010
3	C	8.03%	26.06.2015
4	PHP	6.56%	23.05.2013
5	C#	5.63%	17.03.2014
6	Python	5.45%	02.05.2013
7	Visual Basic	4.12%	16.03.2014
8	Objective C	3.87%	24.07.2003
9	Perl	2.36%	13.03.2014

Все такие языки являются очень разными, ведь каждый имеет уникальное назначение и принципы работы с ним.

Сравнительный анализ всех ЯП по возможностям, а также способам реализации или по сложности освоения является очень актуальной задачей. Оценивать удобство семантических схем только можно на каком-то примере и для всех современных ЯП можно указать те задачи, для которых будут пользоваться намного лучше, чем остальными.

Подобные сравнения часто переходят иногда и в самую настоящую «битву» между программистами. Тут также каждая с сторон яростно выполняет оборону «своего» языка, не принимая доводы иной.

Самым первым шагом для разработки языков, которые содержат основные понятия, которые так близки человеку, стали языки, которые переводят некоторые символические обозначения в машинный код (Ассемблер).[14]

ЯП низкого уровня – это язык программирования, предназначенный для определенного типа компьютера и отражающий его непосредственный машинный код; языки низкого уровня часто программисты называют машинно-ориентированными ЯП. Их сложно как-то конвертировать для реализации их на ПК с самыми разными центральными микропроцессорами, также довольно тяжело выполнить изучение, поскольку требуется очень хорошо знать все принципы внутренней схемы ПК.[3]

К ЯП, так называемого, низкого уровня принадлежат такие специальные языки – ассемблеры – это специальные языки машинно-зависимого плана, что описывают разные действия, команды в описании процессора.

Для каждого типа процессоров существует только свой определенный язык, а о портативности таких программ в другую платформу говорить вовсе не приходится.

Немного позже были созданы компьютерные программы для трансляции самых разных арифметических выражений, а в 1955 г. был запущен первый функционирующий транслятор для ЯП Фортран.

Язык высокого уровня предназначен для удовлетворения самых разных требований для программиста; он зависит как от внутренних машинных кодов, так и ПК любого типа.

ЯП высокого уровня используют для решения современных и сложных проблем и их часто также называют по этой причине проблемно-ориентированными ЯП.

Все команды ЯП высокого уровня эквивалентны сразу нескольким командам, что написаны в машинных кодах, а поэтому такие программы, намного компактны.[4]

Машинно-ориентированными ЯП называются языки, наборы разных типов операторов и средства, которые сильно могут зависеть от ЭВМ (внутреннего машинного языка).

Эти языки также позволяют использовать возможности:

- высокое качество программ;
- предсказуемость кода;
- возможность разработки с помощью конкретных ресурсов;
- для описания оптимальных программ нужно также знать команды и все особенности функционирования ЭВМ;
- очень низкая скорость создания программ;
- трудоемкость процесса составления кода, плохо защищенного от наличия ошибок;
- невозможность использования программ, составленных на этих ЯП, в других ЭВМ.

Машинно-независимые языки – это средства описания алгоритмов для выполнения задач, подлежащей обработке. Они также очень удобны при использовании для широкого круга программистов.

Стоит отметить, что разные командные последовательности (или подпрограммы), часто используемые в обычных машинных программах, представляются в современных высокоуровневых языках только как отдельные операторы.

В первом разделе курсового исследования рассматривались теоретические понятия, связанные с языками программирования. Отметим, что все языки программирования нынешнего этапа развития являются очень популярными.

Для обеспечения удобства проектирования программных средств создаются также специальные среды для интегрированного программирования – это специальные системы, что объединяют трансляторы, редакторы текстов, отладчики (к примеру, языки программирования серии Turbo).

Получили распространение системы для визуальной разработки программ – это специальные средства, при помощи которых создаются ПО путем визуального проектирования.

## **2. Описание средств разработки ПО**

### **2.1. Проблема выбора методологии и средств разработки**

В настоящее время существует множество программных продуктов, что позволяют в сжатые сроки качественно разработать программные продукты для разных предметных областей.

К таким относятся следующие программные средства: Delphi, Visual C++, Visual Basic, C++ Builder, Java Builder и многие другие.

Использование средств такого типа оправдано, когда нужно в сжатые сроки создавать приложение с понятным и удобным и графическим интерфейсом.[3]

Приняв во внимание все вышеперечисленные аргументы, для создания программных продуктов целесообразно использовать средства быстрой разработки RAD.

Для функционирования любого программного комплекса, необходима некоторая программная среда, а в простейшем случае представленная ОС. В сложных случаях, если система обрабатывает большое количество данных, что необходимо поддерживать и контролировать в актуальном состоянии, присутствует некоторая СУБД.

Для обоснованного и правильного выбора RAD-средства нужна оценка продуктов за определенным критериям экспертами.

Непосредственно получить оценку таких продуктов можно с специальных источников. Хотя эта оценка дается, имея специфику разработки некоторого приложения.

Рациональный выбор средства для разработки приложения можно выполнить только в контексте определенного проекта и конкретной организации, которая ведет разработку.[6]

Поэтому для правильной оценки всех средств разработки приложения очень нужна оценка экспертов, которые ознакомлены с спецификой разрабатываемого приложения и вопросами его модификации и сопровождения. [9]

Во внимание принимаются различные критерии по оценке качества программного продукта, а именно такие, что учитывают аспекты всех разрабатываемых программных продуктов:

- доступность программных средств для реализации и разработки;
- соответствие выбранных программных средств уровню программиста;
- возможности программных средств при разработке профессиональных приложений;
- перспективность и жизнеспособность компаний изготовителей программных средств, а также возможность обновления, наличия новых версий программных продуктов;
- скорость работы приложения;
- простота языка программирования;
- наличие интегрированного отладчика;
- обработка исключительных ситуаций.

## **2.2. Обзор языка программирования C++**

В нынешнее время ЯП C++ считают одним из самых популярных и основных языков, которые могут использоваться для разработки программ. В последние годы господство ЯП C++ слегка было поколеблено, ведь широкое развитие уже получили такие языки, как Java, Python, C#, хотя маятник мнения опытных программистов уже качнулся на обратную сторону, большинство квалифицированных программистов, что покинули программирование на ЯП C++, поспешили сразу возвратиться к своей прежней привязанности. [7]

ЯП C++ – это язык для общего назначения, что очень часто применяется для написания кода для системного программирования, понимаемом в очень широком смысле слова.

Кроме того, язык C++ успешно может применяться для написания приложений, далеко выходящих за рамки классического системного кода. Реализации языка C++ также присутствует полностью во всех машинах, от самых малофункциональных – и до супер-ПК. [1]

Б. Страуструп является первым создателем и разработчиком для языка C++, создателем транслятора для него. Он является авторитетным и опытным

сотрудником научно-исследовательского центра корпорации AT&T в Нью-Джерси (США).

Страуструп получил престижное звание почетного магистра с вычислительной техники в университете города Арус (Дания), а имеющееся докторское звание – в университете города Кембридж. [17]

Он специализируется также в сфере операционных систем (ОС), разных распределенных ИС, программирования, моделирования, является автором по руководству С++.

ЯП С++ многим обязан безусловно ЯП С [18], который сохраняется как некоторое его подмножество. Сохранены все свойственные Си средства и методы для низкого уровня, предназначенные для решения задач по системному программированию. [7]

Название С++, было придумано Р. Маскитти еще в 1982 г. Оно имело возможность показать эволюционный характер нового ЯП С++. А обозначение «++» определяет инкремент операции наращивания.

Изначально ЯП С++ спроектировали также для того, чтобы Б. Страуструпу и его команде разработчиков не надо было выполнять создание программы на языках ассемблера или других языках низких уровней. [3] Основным его предназначением было сделать более приятным именно процесс программирования, а также упрощать его для некоторых отдельных программистов.

До какого-то времени не было вовсе определенного какого-то графика разработки для языка С++. Реализация и его документирование для всех его средств шли практически параллельно.

Поэтому данный язык продолжает развиваться так, чтобы быстро преодолеть некоторые проблемы, возникающие для пользователей.

Примерно в 1984 году стало очевидным, что некоторые работы по стандартизации С++ являются неизбежными и надо незамедлительно приступить к проектированию базиса для нее.

Организация AT&T также внесла свой вклад в этот этап работы. Больше 100 представителей выполняли приемы изучения и комментировали все аспекты языка, что и стали современной версией руководства и материалами стандарта С++.

В процессе разработки С++ самым важным фактором была его простота. Ведь при возникновении вопросов, что именно надо упростить: руководство

или документацию, то выбирали всегда первое. Огромное значение также разработчики придавали совместимости C++ с ЯП Си, что мешало удалить весь синтаксис.[11]

Пример кода на языке C++ показан на рисунке 1:

```
typedef enum class color {
    red, blue, green, yellow
} color_t;

void ColorCheck(int flag) {
    color c;

    switch (flag) {
        case 0: c = color::red; break;
        case 1: c = color::blue; break;
        default: c = color::green;
    }

    switch (c) {
        case color::red: cout << "red"; break;
        case color::blue: cout << "blue"; break;
        case color::green: cout << "green"; break;
        case color::yellow: cout << "yellow"; break;
        default: cout << "err";
    }
}
```

Unreachable code [more...](#) (⌘F1)

Рисунок 1 – Пример кода языка C++

В ЯП C++ нет различных типов данных, а также операций высокого уровня. К примеру, не существует одного типа «матрица» с операциями обращения, или «строка» с операцией встроенной конкатенации.

Но иногда пользователю понадобится некоторый тип, то он может определить с легкостью его непосредственно посредством самого языка. Программирование в C++ также сводится к определению зависящих параметров от типов или области написания кода. [10]

Язык C++ создавался также для использования его для традиционной среды, а именно в системе для программирования C в ОС Linux. Но также есть уже обоснованные методы по применению C++ для этой богатой программной

среды. К примеру, все развитые системы трансляции, динамическая загрузка и БД, можно применять и в Linux.[17]

Язык C++ проектировался в свое время в качестве объектно-ориентированного и поддержки абстракции в C.

### **2.3. Обзор языка программирования Java**

ЯП Java – это кросс-платформенный язык программирования, что был разработан фирмой Sun Microsystems.

Все приложения для Java обычно будут также компилироваться в специальный стандартный байт-код, работать поэтому они могут практически на любой виртуальной Java-машине быть выполнены вне зависимости от любой компьютерной архитектуры.

Сегодня популярная технология Java достигает некоторого совершенства, что она может предоставлять средства по превращении любых статических Web-страниц в полные и интерактивные динамические документы, что используются для создания распределенных не зависящих от конкретных платформ приложений.

Чтобы успешно конкурировать с производителями в сфере бытовой электроники, любая компания также должна рассматривать процессоры в качестве товара, который в некоторый момент заменятся на более дешевые, обеспечивать совместимость работы и соблюдать все возможные существующие стандарты.

Java – это не лишь ЯП, а и специальная платформа для проектирования приложений.

Изначально этот язык назывался Oak, был разработан Джеймсом Гослингтом для программирования разных популярных в то время бытовых электронных устройств.

Но, его переименовали, в последствии, в Java и стал он использоваться для создания разных клиентских приложений или же серверного программного обеспечения.

Язык был назван в честь специального вида кофе Java, что любили некоторые из программистов, поэтому на официальном его логотипе ЯП изображена специальная чашка с парящимся кофе.

Также существует также другая из версий происхождения этого названия, Java – это сленговое обозначение кофе с иллюзией на специальную

кофе-машину, как пример для некоторого бытового устройства или для выполнения программирования которых он изначально создавался.

Чтобы не связывать эту всю разработку программного обеспечения с конкретной платформой, Дж. Гослингт начал использовать расширение компилятора C++.

Но, стоит отметить, со временем он понял, только один C++ никак не сможет удовлетворять всем потребностям, как бы его ни расширять. Поэтому еще в середине 1990 г. был спроектирован язык под названием Oak.

Программы на Java транслируются в байт-код, что выполняется на виртуальной машине Java – специальной программой, что обрабатывает транслированный байтовый код и сразу передает инструкции оборудованию посредством интерпретатора.

Достоинство метода выполнения программ находится также в полной независимости от этого байт-кода и оборудования, которое имеет возможность выполнять все имеющиеся Java-приложения практически на любых устройствах, для которых и существует уже соответствующая им виртуальная машина.

Другой специфической особенностью технологии Java они считают гибкую систему для безопасности благодаря тому, что выполнение программы полностью контролируется виртуальной машиной.

Практически все операции, которые превышают некоторые установленные полномочия программ (попытка несанкционированного доступа или соединения с некоторым ПК) вызывают немедленное их прерывание программным способом.

Часто к недостаткам концепции виртуальной машины можно отнести и то, что выполнение специального байт-кода конкретной виртуальной машиной значительно будет снижать производительность программ или же алгоритмов, что реализованы на Java.

Пример кода на Java показан на рисунке 2:

```

7  /**
8   *
9   * @author <i>Irina</i>
10  */
11  public class Student extends Teacher {
12  // fields
13     private int id;
14     private String name;
15     private int age;
16     private int unusedField; //unused field
17
18     public Student(int id, int age, String name) {
19         this.id = id;
20         this.name = name;
21         this.age = age;
22     }
23     @Deprecated
24     public int getAge() {
25         return age;
26     }

```

Рисунок 2 – Пример кода на Java

В последнее время был также внесен ряд новых и прогрессивных усовершенствований, которые увеличивали несколько скорость выполнения некоторых программных продуктов:

- широкое использование кода для платформенно-ориентированного вида,
- применение технологий трансляции специального байт-кода в стандартный машинный код прямо за время работы программы с возможностью сохранения промежуточных версий класса,
- аппаратные средства, обеспечивающие ускоренную обработку конвертируемых байт-кодов.

Идеи, которые заложены в такую концепцию, а различные среды для реализации виртуальной машины в Java, значительно вдохновили энтузиастов для расширения некоторого перечня языков, что также могли бы быть использованы для создания программных средств.

Такие идеи нашли свое выражение в общезыковой структуре CLI.[1]

## 2.4. Обзор языка программирования C#

В последнее время ЯП под названием C и C++ являются одними из наиболее используемых для разработки разного рода коммерческих программ, а также бизнес-приложений. Данные языки могут устраивать очень многих

разработчиков, хотя они не обеспечивают в действительности требуемой продуктивности разработки.

Процесс для написания приложения с помощью ЯП C++ занимает зачастую намного больше времени, чем разработка эквивалентного приложения на C#.

Уже сейчас существуют разные языки, которые увеличивающие в несколько раз продуктивность разработки за счет потери в гибкости, которая привычная в C и C++.

Традиционным средством для отладки программных продуктов на стадии выполнения этапов разработки в C++ является маркировка обширных частей кода программ директивами типа `#ifdef`.

В языке C#, применяя такие атрибуты с ориентацией на условные слова, программист может быстрее написать свой программный код.

Все это позволяет очень сильно упрощать работу, вместо того чтобы создавать автоматизированный инструмент по проверке всех классов или интерфейсов, он действительно является частью некоторого абстрактного объекта, которым можно воспользоваться просто с помощью сообщений основанных на локальных атрибутах объектов.[1]

ЯП C#, являясь самым последним с широко используемых ЯП, должен впитать полностью весь имеющийся опыт, лучшие стороны всех ЯП, при этом являясь созданным для программирования в среде .NET.

Архитектура.NET продиктовала ему перспективную объектно-ориентированную направленность. [7]

Его введение позволило применить новейшие приемы для корректного программирования. [6]

Проанализировав материалы второго раздела сделаем вывод, что язык C++ в нынешнее время является самым наилучший из языков высокого уровня.

В C# отсутствуют некоторые факторы, необходимые для проектирования некоторых высокопроизводительных приложений, подставляемые функции, деструкторы, выполнение которых гарантируется в самых разных точках кода. А язык Java не может также обеспечивать полноценное низкоуровневое программирования.

## **ЗАКЛЮЧЕНИЕ**

В нынешнее время, при усилении связей между миром коммерции с миром разработки ПО, все корпорации тратят очень много денег и своих усилий на планирование бизнеса, ощущается острая надобность в соответствии с абстрактными бизнес-процессами их программным или функциональным реализациям.

Стоит отметить, что огромное большинство языков какого-то прямого пути для обеспечения связи кода и логики, к сожалению, не имеют.

К примеру, сегодня разные программисты часто комментируют свои созданные программные продукты для объяснения, какие классы реализуют какой-то именно абстрактный бизнес-объект. Например, ЯП С# позволяет применять масштабные типизированные, расширяемые метаданные, что часто могут быть использованы к некоторому объекту.

Архитектурой всякого проекта могут также определяться и локальные программные атрибуты, что будут как-то связаны с всяческими элементами языка.

В процессе выполнения исследования были получены, закреплены практические навыки в разработке программ при применении различных компонентов в написании программ для среды визуального программирования С++ Builder.

В процессе написания курсовой работы были реализованы задачи:

- рассмотрена классификация ПО, а именно инструментальное ПО;
- описаны теоретические основы теории языков программирования;
  - охарактеризованы этапы развития языков программирования;
  - освоено выполнение выбора программных средств для реализации конкретного задания;
  - продемонстрировано практическое применение выбора программных средств для создания конкретного программного продукта.