

Введение

Проблема защиты информации путем ее преобразования, исключающего ее прочтение посторонним лицом волновала человеческий ум с давних времен. История криптографии - ровесница истории человеческого языка. Более того, первоначально письменность сама по себе была криптографической системой, так как в древних обществах ею владели только избранные. Священные книги Древнего Египта, Древней Индии тому примеры.

С широким распространением письменности криптография стала формироваться как самостоятельная наука. Первые криптосистемы встречаются уже в начале нашей эры. Так, Цезарь в своей переписке использовал уже более менее систематический шифр, получивший его имя.

Бурное развитие криптографические системы получили в годы первой и второй мировых войн. Начиная с послевоенного времени и по нынешний день появление вычислительных средств ускорило разработку и совершенствование криптографических методов.

Проблема использования криптографических методов в информационных системах (ИС) стала в настоящий момент особо актуальна.

С одной стороны, расширилось использование компьютерных сетей, в частности глобальной сети Интернет, по которым передаются большие объемы информации государственного, военного, коммерческого и частного характера, не допускающего возможность доступа к ней посторонних лиц.

С другой стороны, появление новых мощных компьютеров, технологий сетевых и нейронных вычислений сделало возможным дискредитацию криптографических систем еще недавно считавшихся практически не раскрываемыми.

Основываясь на этих заключениях, было принято решение проверить а как же в нашем современном обществе происходит кодирование информации, и как это кодирование используется человеком для получения знаний о потребляемых услугах и товарах.

Цель данной работы: создание программы для определения подлинности товара, путем обработки их штрих-кода и его расшифровки.

Достижение этой цели будет реализовано через следующие **задачи**:

- анализ влияния кодирования на деятельность человека;
- исследование известных языков программирования;
- анализ штрих-кодовой информации и ее значения в жизни человека;
- создание программы проверки подлинности товаров, через раскодировку их штрих-кода.

Методы решения основных задач: - анализ, - исследование, - синтез.

ОПРЕДЕЛЕНИЕ ШТРИХ-КОДОВОЙ ПОДЛИННОСТИ ТОВАРА С ПОМОЩЬЮ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

ГЛАВА 1. ИСТОРИЯ КРИПТОГРАФИЧЕСКИХ МАШИН

Читая современные публикации, посвященные криптографии, может сложиться впечатление, что она зародилась вскоре после изобретения компьютера и долгое время использовалась лишь для военных нужд. Действительно, появление компьютеров очень преобразило криптографию, но считать XX век ее началом, безусловно, не стоит. Люди стремились максимально защитить свои секреты всегда, поэтому еще до нашей эры не только придумывались, но и довольно широко применялись разные алгоритмы шифрования.

По прошествии многих лет уже трудно достоверно определить дату зарождения шифрования. Вполне возможно, что она лишь немного "моложе" даты возникновения письменности.

Чем сложнее шифр, тем труднее отправителю секретного сообщения и его получателю пользоваться им (но зато, конечно, такое сообщение труднее и расшифровать). Поэтому, естественно, были созданы специальные устройства для шифрования и расшифровки текста.

В XVI веке итальянец Альберти впервые выдвинул идею "двойного шифрования" — текст, полученный в результате первого шифрования, подвергался повторному. Реализация такого шифра осуществлялась с помощью шифровального диска, положившего начало целой серии так называемых "многоалфавитных шифров". Пять подобных шифровальных дисков использовались правительством и армией конфедератов во время гражданской войны в Северной Америке (*рис. 1*).

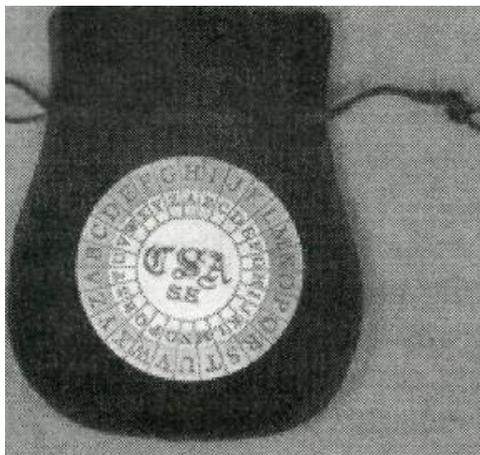


Рис. 1

В начале XIX века криптография обогатилась замечательным изобретением. Его автор — первый государственный секретарь, а затем и президент Соединенных Штатов Америки Томас Джефферсон. Шифр реализовывался с помощью специального устройства в виде цилиндра (см. *рис. 2*), которое впоследствии назвали "шифратором Джефферсона". В XX веке криптоаналитики США признали высокую стойкость шифра Джефферсона. Они даже назвали его автора "отцом американского шифровального дела".

Идея Джефферсона возродилась и в России. В 1916 году подпоручиком Попазовым было изготовлено шифровальное устройство, впоследствии названное «Прибор Вави». Устройство по своей идее было адекватно цилиндру Джефферсона, но вместо дисков на оси имелись 20 колец, надетых на цилиндр вплотную друг к другу, которые могли вращаться на нем.



Рис. 2

В конце XIX века криптография начинает приобретать черты точной науки, ее начинают изучать в военных академиях. Здесь следует отметить роль французской военной академии Сен-Сир, в которой к этому времени был разработан свой собственный военно-полевой шифр и шифровальное устройство, получившее название "Линейка Сен-Сира" (*рис. 3*).



Рис.3

Линейка Сен-Сира позволила существенно повысить эффективность труда шифровальщика, облегчить алгоритм реализации шифра. Именно в этой механизации процессов шифрования-дешифрования и заключается вклад авторов линейки в практическую криптографию. Это изобретение получило достаточно широкое распространение.

Яркой фигурой в криптографии XIX века является англичанин Чарльз Бэббидж. Потомкам он известен как "изобретатель вычислительных машин". Бэббидж еще в юности увлекся криптографией. С годами он приобрел славу лучшего в Англии эксперта по тайнописи. К нему обращались с просьбой о дешифровании самых разных документов: от записок первого британского придворного астронома Дж. Флэмстида и жены короля Карла I Генриетты до писем, фигурировавших в уголовных делах. По каким-то причинам Чарльз Бэббидж не публиковал своих работ ни по криптографии, ни по созданию вычислительных машин. Ученый мир об успехах Бэббиджа узнал только в XX веке, когда исследователи принялись разбирать его обширные архивы. Чарльз Бэббидж впервые дал строгую математическую формализацию основных, понятий криптографии. Он предложил алгоритм решения многоалфавитных шифров, многие из которых до этого времени считались "нераскрываемыми". Ему же принадлежит одна из идей раскрытия систем с использованием "самоключа", т.е. ключ зависит от открытого текста. Также Бэббидж одним из первых математиков начал применять алгебру в области криптографии.

В XIX веке значительный вклад в криптографию внесла целая плеяда французских специалистов. Офицер французской армии маркиз де Виари изобрел шифровальное устройство с печатающим механизмом. Этьен Базери увлекался историческими шифрами. Ему удалось почти полностью восстановить несложный код Наполеона I, который тот использовал в низовых звеньях своей армии. Также этот код был раскрыт и русскими специалистами.

В 1924 году изобретатель Александр фон Крыха создал механическую криптографическую машину (рис. 4). Она широко использовалась дипломатами и банкирами разных стран до 1950-х годов. Выпускались три модели: стандартная, карманная, названная "Лилипут", и электрическая "Крыха" (рис. 5), которая могла кодировать 360 символов в минуту. К ней можно было подключать и телеграфное оборудование фирмы Siemens.

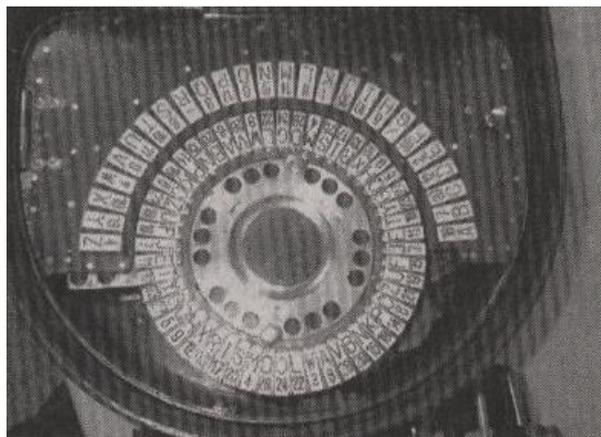


Рис. 4

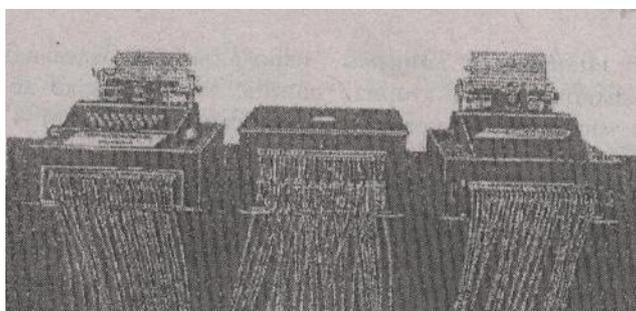


Рис. 5

Непосредственной предшественницей современных криптографических машин была роторная машина, изобретенная американцем Эдвардом Хепберном в 1917 году и названная впоследствии "Энигмой" (в переводе "загадка") (рис. 6). "Энигма" работала совместно с электрической пишущей машинкой. Э.Хепберн продолжал проектировать и строить подобные электромеханические роторные машины до окончания Второй мировой войны.

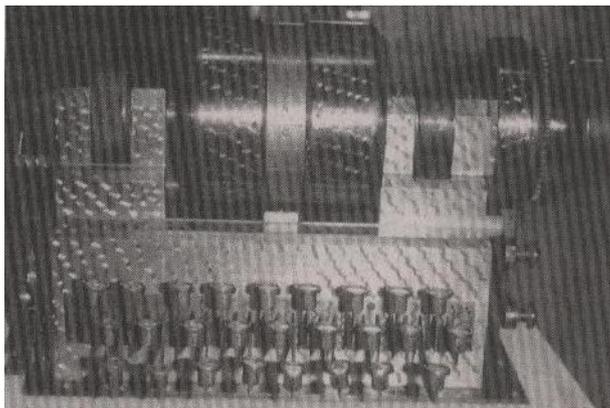


Рис. 6

Независимая промышленная версия "Энигмы" создана чуть позже (в 1923 г.) берлинским инженером Артуром Шербиусом (рис. 7). Поразившись ее надежности, которая основывалась на статистических исследованиях, германское правительство сохранило все права на нее за собой и начало использовать для нужд своих вооруженных сил. Она стала стандартной шифровальной машиной немецкой армии, разведки и полиции. Также она использовалась в высших эшелонах командования армией.

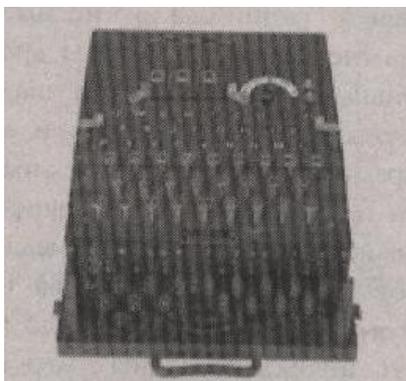


Рис. 7

Иногда для «Энигмы» использовались отдельная ламповая индикаторная панель, что создавало дополнительный уровень секретности. «Энигма М 4» имела дополнительное устройство печати. Оно представляло собой принтер с печатающим колесом, на котором размещалось 26 букв.

Во время Второй мировой войны было построено около 100 000 машин «Энигма», которые использовались вооруженными силами Германии для

шифрования секретнейших приказов верховного главнокомандования и аппарата полиции. Германские эксперты объявили криптографическую машину «Энигма» верхом совершенства, а закодированные ею тексты – не поддающимися расшифровке разведкой противника.

Однако, как показала практика, они все-таки поспешили с этим заявлением. В канун Второй мировой войны с помощью разведки англичане сумели заполучить ключевую информацию о секретах «Энигмы» и создали ЭВМ «Колосс». С ее помощью только в период с 1941 по 1945 годы английская разведка расшифровала 338 тысяч перехваченных немецких радиogramм.

ГЛАВА 2. ЭВМ, ПРОГРАММА, ПРОГРАММИРОВАНИЕ

1.1 Из истории создания ЭВМ

С древних времен люди создавали различные механизмы для облегчения своего труда. Но эти машины облегчали только физический труд человека. Деятельность связанная с умственным трудом, обходилась без серьезного участия машин, пока не были созданы электронно-вычислительные машины (ЭВМ). Первые настоящие ЭВМ появились во время Второй мировой войны в Англии. Они были строго засекречены и применялись для расшифровки вражеских шифров и расчета траекторий полета снарядов. После войны использовать компьютеры могли только крупные корпорации и правительственные организации из-за высокой стоимости их обслуживания. В этих компьютерах использовались стеклянные электронные лампы (почти 20 тыс. шт.), поэтому они имели огромные габариты, весили несколько десятков тонн и потребляли много энергии. Лампы часто выходили из строя, и для устранения неисправностей требовались бригады инженеров. Огромные ламповые компьютеры относятся к первому поколению. ЭВМ второго поколения были построены на транзисторах. Компьютеры на интегральных схемах (ИС) относятся к третьему поколению, а изобретение микропроцессора и дальнейшая микроминиатюризация привели к рождению четвертого поколения ЭВМ. Непрерывно развиваясь и наращивая свою вычислительную мощность, ЭВМ сравнительно долго оставались уникальными инструментами. Они занимали огромные площади, потребляли очень много энергии, были ужасно дороги в эксплуатации. Например, в 70-х годах уже, правда, прошлого века час машинного времени ЭВМ, с трудом умещавшейся в одной учрежденческой комнате, стоил несколько сот советских рублей. Поэтому эти инструменты были мало доступны и применялись для решения государственных, промышленных и научных задач. Теперь термин ЭВМ встречается все реже. Все чаще – компьютер. А совсем часто- персональный компьютер (ПК). Они всюду. Они охватывают все сферы нашей жизни и деятельности.

Персональные компьютеры пока еще не стали нам братьями по разуму. Это очень сложные программируемые универсальные цифровые автоматы. Они предназначены для накопления, хранения и обработки информации, для управления различными объектами и процессами. Но все это компьютеры делают под управлением программ, создаваемых человеком.

1.2. Что такое программа?

С этим понятием мы сталкиваемся практически ежедневно. Хотим мы написать книгу, или хотя бы сочинение, начинаем с разработки плана, перед генеральным выходом за покупками – составляем список, собрались приготовить новое блюдо – ищем его рецепт. Программа – это и есть план, список последовательных действий, которые необходимо выполнить для достижения поставленной цели. Точность и подробность плана-программы зависит от вашей цели.

Покупки, например, можно совершать в произвольном порядке, а при приготовлении блюда важно не только строго следить за последовательностью действий, но и за дозировкой продуктов. Вид и детализация программы зависит и от ее предполагаемого исполнителя. Если Вы хотите чаю, то можете просто попросить кого-нибудь из домочадцев вскипятить чайник. Гостю уже потребуются дополнительные указания: где взять чайник, воду, спички. А если гость-иностранец? Придется применить свои лингвистические способности или объясниться на пальцах. Для обучения ребенка понадобятся еще более подробные инструкции.

1.3. Структурное программирование

Это и есть то самое создание программ. Компьютерное программирование часто рассматривается как вид искусства. Однако многие институты преподают программирование как науку. Исследовательские учреждения давно ищут пути увеличения эффективности труда и уровня профессиона-

лизма разработчиков программ, т.е. интеллектуальной собственности, которая имеет свой особый статус и свое правовое поле. Существуют различные подходы к программированию.

Одним из них является структурное программирование, представляющее попытку превратить искусство программирования в обычную науку. Основной принцип структурного программирования заключается в создании некоторых аналитических моделей решаемой задачи перед началом кодирования. Двумя наиболее используемыми старыми моделями решаемой задачи перед началом кодирования. Двумя наиболее используемыми старыми моделями являются разработки «сверху вниз» и «снизу вверх». Разработка «сверху вниз» означает, что программа создается исходя из принципа первоочередного решения главных задач. Новые более детальные уровни разрабатывают по мере успешной реализации предыдущих более высоких уровней. Идея разработки «сверху вниз» заключается в том, что все хорошие программы состоят из хорошо отлаженных процедур и функций, объединенных логично и разумно. Новые уровни программы разрабатываются с использованием уже готовых функций и процедур, пока программа не будет готова.

1.4. Объектно-ориентированное программирование

В настоящее время значительным сдвигом на пути разработки технологичного программирования является объектно-ориентированное программирование.

Объектно-ориентированное программирование задумывалось как некая мощная концепция, уменьшающая время разработки программ и отладки программ. Объектно-ориентированное программирование задумывалось как некая мощная концепция, уменьшающая время разработки программ и отладки программ. Объектно-ориентированное программирование описывает программы в терминах логических сущностей и взаимодействий между ними. Основным элементом программирования является класс, от которого порождаются многочисленные объекты со своими инкапсулированными данными и интерфейсами, позволяющие выполнять определенные встроенные действия. Отличительной особенностью

стью объектно-ориентированного программирования является мощный механизм эволюционного развития и встраивания новых интерфейсов в последующее поколение классов.

Концепция объектно-ориентированного программирования соответствует внутренней логике функционирования операционной системы Windows. Поэтому приложения, разработанные на основе объектно-ориентированного программирования, приобретают большую надежность кода и возможность широкого использования однажды построенных классов.

1.5. Визуальное программирование

Позволяет разрабатывать все элементы приложения, непосредственно наблюдая результаты своей работы на экране. Иными словами, разработчик увидит, как будет работать его приложение еще до того, как полностью будет написана программа. Реализация виртуального программирования стала возможной в связи с развитием графического интерфейса GUI (Graphical User Interface).

Основным понятием визуального программирования является интерфейсный элемент, который в терминах объектно-ориентированного программирования является объектом. Объекты - кнопки, строки, списки и т.д. – являются кирпичиками, из которых строится приложение. Среда визуального программирования позволяет увидеть объекты на экране монитора до выполнения программы. Изменение внешнего вида объекта выполняется с помощью таких его свойств, как высота, ширина, положение, цвет и др. Свойства – это атрибуты объекта, которые составляют его индивидуальность и помогают описать его. В то же время свойства обеспечивают возможность реагировать на внешние события.

ГЛАВА 3. АНАЛИЗ НЕКОТОРЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

2.1 Язык программирования Pascal

Язык программирования Pascal появился в начале 1970-х годов. Pascal стал “наследником” Algola (**algorithmically language**), который был разработан в 1950-60-х годах. По сравнению с Алголом Паскаль проще и яснее. У него намного лучшие возможности обработки данных и имеются встроенные процедуры ввода-вывода, которых не было в Алголе. Автор языка, швейцарский ученый Николаус Вирт назвал его Паскалем в честь великого французского математика, философа и изобретателя 17 века Блеза Паскаля. Паскаль изобрел вычислительное устройство, именно поэтому новому языку было дано его имя.

Благодаря своей четкости и логичности Паскаль надолго занял свое место среди других языков программирования не только как средство разработки серьезных программ-приложений. Он также очень хорошо подходит для обучения программированию. Впоследствии появились различные версии языка и его расширения, дополненные новыми возможностями. Наиболее известным расширением стал пакет Turbo Pascal от фирмы Borland, появившийся в 1983 году и сразу ставший событием в мире компьютерных технологий. Первое упоминание о нем содержалось в рекламе, опубликованной в журнале “BYTE”, а сам пакет предназначался для операционной системы CP/M. В начале 1984 года он был перенесен в среду операционной системы для персональных компьютеров MS-DOS и приобрел огромную популярность. С тех пор появилось несколько версий Turbo Pascal, последняя – седьмая.

Фирма Borland/Inprise завершила линию продуктов Turbo Pascal (Turbo – торговая марка разработчика системы фирмы Borland International, Inc.(США)) и перешла к выпуску системы визуальной разработки для Windows – Delphi. Несмотря на это, Turbo Pascal продолжает играть роль отличного языка для первого знакомства с миром “серьезного” программирования. Это связано как с его логической структурой, так и с теми возможностями, которые позволяют использовать Turbo Pascal для решения разнообразных задач. Среди них вычис-

ления и обработка данных, компьютерная графика, работа со звуком, системное программирование. Turbo Pascal позволяет применять приемы объектно-ориентированного программирования – одной из ведущих современных компьютерных технологий. Вместе с тем Turbo Pascal прекрасно работает на сравнительно маломощных компьютерах, и поэтому именно этот пакет часто устанавливается в компьютерных классах школ и кружков информатики.

2.2. Язык программирования C++

Язык программирования C++ был создан как усовершенствованная версия языка C. C был создан Брайаном Керниганом и Деннисом Ритчи в Bell Labs в период с 1969 по 1973 год. Он первоначально был предназначен для программирования служебных программ низшего уровня типа операционных систем (Керниган и Ритчи разрабатывали Unix). Предполагалось, что он заменит ассемблер. Программы, написанные на ассемблере было очень трудно читать. Кроме того, на ассемблере очень трудно создавать программы, которые можно было бы легко разделить на отдельные модули. C получил широкое распространение и стал ключевым языком для Unix и в конечном счете для Windows.

C самого начала C создавался для написания высокоэффективных программ, и эта же цель преследовалась и при создании C++. При создании программ на C приходится применить процедурный стиль программирования. Процедурное программирование позволяет создавать программы, которые являются набором функций или процедур, обрабатывающих данные, чтобы получить определенный результат. Функции могут обращаться к другим функциям для получения услуг и помощи, что позволяет применить стратегию “разделяй и властвуй” и тем самым упростить решение задач. Кроме того, C – язык со строгим контролем типов. Это означает, что каждый элемент данных в C имеет тип и может использоваться с другими частями данных только так, как определено в их типах. Языки со слабым контролем типов (такие как Basic) либо игнорируют, либо скрывают этот важный принцип. Строгий контроль типов га-

рантирует, что программа правильна, точнее, выполняет осмысленные действия еще до первого его запуска.

Бьярн Страуструп разработал язык C++ в 1983 году как расширение языка C. C++ имеет большинство возможностей C. Фактически многие программы на C++ выглядят точно так же, как и на C. В C++ предусмотрен объектно-ориентированный стиль программирования.

2.3. Язык программирования Delphi

Delphi – это могучая среда разработки самых разнообразных программных приложений, являющаяся одновременно и редактором и компилятором. Данная среда ориентирована на работу, прежде всего, в операционных системах Windows. Однако с помощью Delphi можно разрабатывать приложения для Linux и строить кроссплатформенные приложения для Windows-Linux. Руководитель исследовательской группы по разработке системы Delphi Чак Язджевски вспоминает: “Имя Delphi было предложено Дэнни Торпом во время одной мозговой атаки. Мы хотели, чтобы в имени отразились уникальные способности продукта к работе с базами данных, и Delphi как нельзя лучше перекликалось с таким заслуженным именем в этой области, как Oracle(одна из распространенных программных систем для управления базами данных), по крайней мере, для тех, кому сочетание “дельфийский оракул” о чем-то говорит...”

Система Delphi воплощает в себе лучшие достижения современной теории программирования. Иногда говорят, что Delphi – это интегральная система для разработки программ. Действительно, она объединяет в себе множество полезных инструментов и готовых компонентов, из которых, как дом из кирпичиков, собираются проекты – пользовательские программы. Delphi – визуальная среда разработки программ. Это означает, что внешний вид каждой программы (ее “лицо”, или интерфейс) создается простым перемещением составных элементов из некоторого набора. Такое проектирование по силам даже тем, кто умеет лишь собирать впечатляющие сооружения из деталей детского конструктора.

тора. Конечно, чтобы такая конструкция “оживла” и заработала, необходимо придумать и кое-что еще – написать программу ее поведения.

Базовым языком программирования в Delphi служит язык Object Pascal – объектно-ориентированный Паскаль. Продуктивным ядрышком, из которого буйно разрослось ветвистое дерево Delphi, был старый и добрый (30 лет – это уже история!) язык программирования Turbo Pascal. Принципиальное различие систем программирования Delphi и Turbo Pascal состоит в использовании экранного режима монитора: Turbo Pascal ориентирован на текстовый режим операционной системы DOS, а Delphi, как и Windows, – на графический. Тем не менее, программы, написанные на языке Turbo Pascal и не использующие графические средства, после небольшой настройки могут исполняться в среде Delphi.

Delphi – мощная современная система программирования, имеющая многочисленные приложения везде, где сегодня применяются компьютеры – от инженерных и научных расчетов до автоматизации управленческой деятельности. Прежде всего – это инструмент, инструмент довольно тонкий и универсальный, способный на многое в руках опытного мастера. Сегодня разнообразие Delphi таково, что изучить все возможности этой системы, даже в рамках школьного факультатива, не представляется возможным. В это случае Delphi можно сравнить с горой, путь к вершине которой далек и долог. На Delphi программировать может научиться каждый. Delphi использует наиболее понятный язык, что позволяет без труда создавать сложные программы.

В среде Delphi могут работать не только профессионалы. Среда разработки сохраняет простоту и наглядность процесса создания приложений, основанного на использовании технологий визуального программирования. Объектно-ориентированный компонентный подход позволяет легко и быстро создавать не только интерфейс программ, но и достаточно сложные механизмы доступа к данным, а также повторять и тиражировать удачные программные решения.

ГЛАВА 4. КОДИРОВАНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЯХ

Для общения друг с другом мы используем код — русский язык. При разговоре этот код передается звуками, при письме — буквами. Водитель передает сигнал с помощью гудка или миганием фар. Вы встречаетесь с кодированием информации при переходе дороги в виде сигналов светофора. Таким образом, кодирование сводится к использованию совокупности символов по строго определенным правилам.

Кодировать информацию можно различными способами: устно; письменно; жестами или сигналами любой другой природы. Когда создается информационная модель объекта или явления, то их свойства, параметры, состояния, среда, действия могут отображаться также с помощью различных способов кодирования.

Люди всегда искали способы быстрого обмена сообщениями. Для этого посылали гонцов, использовали почтовых голубей. В древности, а у некоторых народов и сейчас, существовали различные способы оповещения о надвигающейся опасности: барабанным боем, дымом костров, флагами и т. д. Однако использование такого способа представления информации требует предварительной договоренности о понимании принимаемого сообщения. Например, что означает поднятая вверх правая рука регулировщика движения?



По мере развития техники появлялись разные способы кодирования информации. Во второй половине XIX века американский изобретатель Сэмюэль Морзе изобрел удивительный код, который служит человечеству до сих пор. Информация кодируется тремя символами: длинный сигнал (тире), короткий сигнал (точка), нет сигнала (пауза) — для разделения букв.

В качестве источников информации может выступать человек, техническое устройство, предметы, объекты неживой и живой природы. Получателей сообщений может быть несколько или один.

В процессе обмена информацией мы совершаем две операции: *кодирование и декодирование*. Первая связана с переходом от исходной формы представления информации в форму, удобную для хранения, передачи или обработки. А вторая — с обратным переходом к исходному представлению информации.

В настоящее время существуют разные способы кодирования и декодирования информации в компьютере. Выбор способа зависит от вида информации, которую необходимо кодировать: текст, число, графическое изображение или звук. Для чисел, кроме того, важную роль играет то, как будет использоваться число: в тексте, или в вычислениях, или в процессе ввода-вывода. Накладываются также особенности технической реализации.

Несмотря на особенности каждого вида информации, общим для них является использование при кодировании двоичной системы счисления, основанной на двух цифрах — 0 и 1. Эти два символа принято называть двоичными цифрами или битами. С помощью двух цифр 0 и 1 можно закодировать любое сообщение.

Инженером такой способ кодирования привлек простотой технической реализации — есть сигнал или нет сигнала. Эти состояния легко различать. Недостаток двоичного кодирования — длинные коды. Но в технике легче иметь дело с большим числом простых однотипных элементов, чем с небольшим числом сложных.

ГЛАВА 5. ШТРИХ-КОДОВОЕ КОДИРОВАНИЕ ИНФОРМАЦИИ

Все видели на упаковке многих товаров ряд вертикальных полосок различной толщины, разделенных пустыми интервалами, под которыми написано число:



Такое изображение называется штриховым кодом, или штрих-кодом. Что же это такое и зачем это нужно? Начнем с числа.

В свое время производители товаров и торговые фирмы столкнулись с серьезной проблемой: товаров много (например, средний универмаг оперирует с десятью тысячами наименований), и к каждому - длинный сертификат - документ, в котором расписано, где сделан товар, на какой фирме, сколько весит, какие габариты и т.д. Поэтому придумали систему кодирования этой информации в виде последовательности цифр (и штрихового кода). Более 20 лет назад была создана глобальная международная организация - система товарных номеров EAN/UCC, образованная на основе Европейской (European Article Numbering Association – EAN International) и Северо-Американской (Uniform Code Council - UCC) ассоциаций товарной нумерации. В настоящее время система EAN/UCC объединяет национальные организации в 94 странах мира. Каждая страна имеет свой номер. Чаще всего коды стран - двузначные, но могут быть и трехзначными (например, страны СНГ имеют номера с 460-ю по 469-й, в т.ч. Россия - 460). Внутри каждой страны проводится нумерация предприятий - изготовителей товаров. В Российской Федерации национальной организацией товарной нумерации - членом EAN International является Ассоциация автоматической идентификации ЮНИСКАН/EAN РОССИЯ, которая насчитывает около 5000 предприятий-членов. Всем им присвоены уникальные идентификационные номера, которые начинаются с цифр 460 (EAN РОССИЯ). Регистрационный номер предприятия отображается на упаковке продукции в виде первых цифр кода EAN (например, 4600952). Следующие 5 цифр - это закодированная информация о товаре (имя товара, масса, состав, цвет и т.п.). Итак, всего всю необходимую информацию отражают 12 цифр. Но внимательный читатель, ко-

нечно, обнаружит, что на приведенном выше рисунке в числе не 12, а 13 цифр. Дело в том, что последняя, тринадцатая, цифра - контрольная.

При наличии на упаковке товаров закодированной информации о них можно автоматизировать процесс распознавания этой информации, если считывать ее специальным устройством - сканером. Причем можно, конечно, использовать для распознавания информации о товаре указываемые на упаковке цифры. Но это потребовало бы применения сложной компьютерной технологии распознавания символов. Проще и надежнее это делать с использованием двоичного кодирования этой информации. Нет, речь не идет о том, чтобы представлять число-код в виде цифр двоичной системы счисления. Просто десятичный номер товара изображается на упаковке в виде тех самых вертикальных полосок различной толщины и интервалов между ними, а эта информация является двоичной, хотя на первый взгляд этого и не скажешь.

Если сделать тонкий срез этих полосок, то можно увидеть следующее (в увеличенном масштабе):



Эти полоски и пробелы графического изображения штрихового кода очень хорошо понятны специальным приборам - сканерам. Считывая эту информацию слева направо, сканер присваивает 1 первой встреченной черной полоске и 0 - первому промежутку. Следующие промежутки и штрихи считываются как последовательности одного, двух, трех или четырех нулей или единиц, в зависимости от ширины штриха или промежутка. Следовательно, все изображение может быть представлено как последовательность битов:

101000011001011000010011001...

Эти биты и есть двоичное представление десятичного числа - кода товара. При считывании штрихового кода сканер из комбинации штрихов восстанавливает закодированный номер. Те, кто бывали в крупных магазинах, видели, как кассир, делая расчет, просто пронесит товар, повернув его штрих-кодом вниз, над кассовым аппаратом, и на экране аппарата мгновенно выскакивает цена. Это происходит потому, что кассы со считывателями штрихового кода

подключены к компьютеру, который обрабатывает считанную информацию. Кроме удобства работы кассира и "быстроты" обслуживания покупателя, такая автоматизированная система может обеспечить и учет объема продаж того или иного товара, уровень спроса на те или иные изделия, заблаговременно сделать заказ на склад для восполнения запасов товаров на полках торгового зала и т.п.

Некоторых покупателей смущает, если на штрих-коде - только собственно штрихи, а цифр нет. Это - не признак подделки. Для кассового аппарата цифры вообще не имеют значения, и, если места на товаре мало, их не ставят.

Не нужно пытаться продавца и в том случае, если штрих-код узкий или короткий, или вообще "какой-то не такой". Обычно так бывает на мелких по размеру товарах. ЮНИСКАН разрешает производителям таковых использовать сокращенные варианты кодировки.

Есть способ, хотя и несколько трудоемкий, узнать по штрих-коду, поддельный ли товар. Используя последнюю контрольную цифру можно проверить правильность кода товара. Итак, если вам крайне важно узнать, с чем вы имеете дело, нужно произвести следующие арифметические действия:

1. Сложить цифры, стоящие на четных позициях; для штрих-кода, изображенного на рисунке в начале статьи: $6+0+5+0+0+1 = 12$.

2. Сумму умножить на 3: $12 * 3 = 36$.

3. Сложить цифры, стоящие на нечетных позициях (не учитывая контрольную цифру): $4+0+9+2+0+0 = 15$.

4. Сложить то, что получилось в результате второго и третьего действий: $36 + 15 = 51$.

5. От результата отбросить первую цифру. Получится 1.

6. И отнять от 10 то, что получилось в пятом пункте: $10 - 1 = 9$.

Этот результат должен совпадать с контрольной цифрой. Если нет - товар поддельный.

Метод, конечно, сложный. Однако, если вы покупаете дорогую вещь или есть сомнения, доброкачественный ли продукт питания перед вами, имеет смысл произвести эти в общем-то элементарные процедуры.

ГЛАВА 6. ПРАКТИЧЕСКАЯ ЧАСТЬ

На основании всего выше изложенного и учитывая цель своей работы, я могу теперь определиться с выбором языка программирования для создания программы по определению подлинности товара по его штрих-коду.

Программа написана на языке программирования Delphi.

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, jpeg, ExtCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
    Edit1: TEdit;
```

```
    Edit2: TEdit;
```

```
    Edit3: TEdit;
```

```
    Edit4: TEdit;
```

```
    Edit5: TEdit;
```

```
    Edit6: TEdit;
```

```
    Edit7: TEdit;
```

```
    Edit8: TEdit;
```

```
    Edit9: TEdit;
```

```
    Edit10: TEdit;
```

```
    Edit11: TEdit;
```

```
    Edit12: TEdit;
```

```
    Edit13: TEdit;
```

```
    Button1: TButton;
```

```
    Image1: TImage;
```

```
    Label1: TLabel;
```

```
    Button2: TButton;
```

```
    procedure Button1Click(Sender: TObject);
```

```
procedure Edit1Change(Sender: TObject);
procedure Edit2Change(Sender: TObject);
procedure Edit3Change(Sender: TObject);
procedure Edit4Change(Sender: TObject);
procedure Edit5Change(Sender: TObject);
procedure Edit6Change(Sender: TObject);
procedure Edit7Change(Sender: TObject);
procedure Edit8Change(Sender: TObject);
procedure Edit9Change(Sender: TObject);
procedure Edit10Change(Sender: TObject);
procedure Edit11Change(Sender: TObject);
procedure Edit12Change(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
  {$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,a,b,d,c,s,f:integer;
begin
  s1:=strtoint(edit1.Text);
  s2:=strtoint(edit2.Text);
  s3:=strtoint(edit3.Text);
  s4:=strtoint(edit4.Text);
  s5:=strtoint(edit5.Text);
  s6:=strtoint(edit6.Text);
```

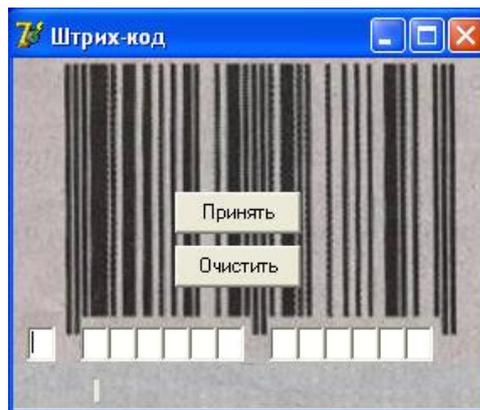
```
s7:=strtoint(edit7.Text);
s8:=strtoint(edit8.Text);
s9:=strtoint(edit9.Text);
s10:=strtoint(edit10.Text);
s11:=strtoint(edit11.Text);
s12:=strtoint(edit12.Text);
s13:=strtoint(edit13.Text);
a:=s2+s4+s6+s8+s10+s12;
b:=a*3;
c:=s1+s3+s5+s7+s9+s11;
d:=b+c;
s:=d mod 10;
f:=10-s;
if f=s13 then
  label1.caption:='Данный продукт лицензированный'else
label1.Caption:='Данный продукт-подделка';
  button2.Enabled:=true;
end;
procedure TForm1.Edit1Change(Sender: TObject);
begin
  Edit2.SetFocus;
end;
procedure TForm1.Edit2Change(Sender: TObject);
begin
  Edit3.SetFocus;
end;
procedure TForm1.Edit3Change(Sender: TObject);
begin
  Edit4.SetFocus;
end;
procedure TForm1.Edit4Change(Sender: TObject);
```

```
begin
Edit5.SetFocus;
end;
procedure TForm1.Edit5Change(Sender: TObject);
begin
Edit6.SetFocus;
end;
procedure TForm1.Edit6Change(Sender: TObject);
begin
Edit7.SetFocus;
end;
procedure TForm1.Edit7Change(Sender: TObject);
begin
Edit8.SetFocus;
end;
procedure TForm1.Edit8Change(Sender: TObject);
begin
Edit9.SetFocus;
end;
procedure TForm1.Edit9Change(Sender: TObject);
begin
Edit10.SetFocus;
end;
procedure TForm1.Edit10Change(Sender: TObject);
begin
Edit11.SetFocus;
end;
procedure TForm1.Edit11Change(Sender: TObject);
begin
Edit12.SetFocus;
end;
```

```
procedure TForm1.Edit12Change(Sender: TObject);
begin
Edit13.SetFocus;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
edit1.Text:="";
edit2.Text:="";
edit3.Text:="";
edit4.Text:="";
edit5.Text:="";
edit6.Text:="";
edit7.Text:="";
edit8.Text:="";
edit9.Text:="";
edit10.Text:="";
edit11.Text:="";
edit12.Text:="";
edit13.Text:="";
label1.Caption:="";
edit1.SetFocus;
end; end.
```

Алгоритм выполнения программы.

Запускаем программу. Изначально она выглядит так



Вводим штрих-код в программу



Нажимаем кнопку “Принять”. Если вы ввели штрих-код правильно, то внизу программы появится надпись либо “Данный продукт лицензированный”, либо “Данный продукт-подделка”.

Если вы допустите хоть одну ошибку при введении штрих-кода(введете цифру, не соответствующую штрих-коду), то программа будет воспринимать данный продукт как подделку.



Убедившись в подлинности продукта, нажимаем на кнопку “Очистить” и вы можете снова вводить другой штрих-код.

Программа проста в использовании и понятна каждому. Весит она всего 532Кб, что повышает ее практичность и возможность носить с собой программу на дискете или на любом другом носителе.



III. ЗАКЛЮЧЕНИЕ

Исследовав некоторые языки программирования и выбрав язык Delphi, мы создали программу для определения подлинности любого товара. Программа проста в использовании и понятна каждому. Весит она всего 532Кб, что повышает ее практичность и возможность носить с собой программу на дискете или на любом другом носителе.

Но и это еще не точка в нашей работе. В данный момент наша программа адаптируется под среду мобильного телефона, что позволит иметь эту программу всегда под рукой.. Научившись ею пользоваться можно обезопасить себя от покупки подделанного товара, тем самым защитить свое здоровье, и не тратить дополнительные деньги на покупки новых более качественных товаров.

Конечно же на этом тоже будет рано останавливаться. Штрих-кодовое обозначение товаров требует серьезной доработки. Более подробную информацию о товаре можно получить по метке радиочастотной идентификации RFID, которая имеет ряд преимуществ перед штрих-кодом.

ЛИТЕРАТУРА

1. Статья «История криптографии», «В мире информатики №15/ Информатика №47/2003»
2. Статья «Первые криптографические машины», «В мире информатики №87/ Информатика №5/2007»
3. Статья «Кодирование информации о товаре», «В мире информатики №55/ Информатика №8/2005»
4. Учебник «Информатика», под ред. Н. В. Макаровой, 7-8 класс, издательство «Питер», 2001
5. Баричев С. «Криптография без секретов»
6. Статья Е. В. Андреевой «Методика обучения основам программирования на уроках информатики», «Информатика №17/2005»

Содержание

I.	Введение	3-4
II.	Определение штрих-кодовой подлинности товара с помощью языков программирования	5
	Глава 1. История криптографических машин	5-10
	Глава 2. ЭВМ, программа, программирование	11
	1.1 Из истории создания ЭВМ	11-12
	1.2 Что такое программа?	12
	1.3 Структурное программирование	12-13
	1.4 Объектно-ориентированное программирование	13-14
	1.5 Визуальное программирование	14-15
	Глава 3. Анализ некоторых языков программирования	15
	2.1 Язык программирования Pascal	15-16
	2.2 Язык программирования C++	16-17
	2.3 Язык программирования Delphi	17-18
	Глава 4. Кодирование информации в компьютерных технологиях	19-20
	Глава 5. Штрих-кодовое кодирование информации	21-23
	Глава 6. Практическая часть	24-29
III.	Заключение	30
IV.	Литература	31