

Основные понятия и определения

Предпосылки перехода к информационному обществу:

- Совершенствование средств накопления и обработки знаний;
- 50-60-е годы XX века – «информационный взрыв»;
- Появление и развитие Интернета;
- Простота, доступность, демократичность средств создания и распространения информации.

Последствия «информационного взрыва»

- Стремительное увеличение объемов информации, как нужной, так и ненужной;
- Увеличение объемов некачественной информации;
- Трудности поиска релевантной информации среди и ориентации в информационном пространстве.

Информация

Самые разнообразные сведения, сообщения, известия, знания и умения, которые человек получает из окружающего мира

Особенно быстро ее роль возросла после изобретения в середине XX века КОМПЬЮТЕРА

Компьютер

Аппаратно-программный комплекс, осуществляющий ввод, хранение, обработку и вывод информации, в соответствии с АЛГОРИТМОМ

Алгоритм

Точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи

Название "алгоритм" произошло от латинской формы имени среднеазиатского математика Аль-Хорезми — Algorithmi

Алгоритм — одно из основных понятий информатики и математики

**С появлением компьютеров
сформировалась наука
ИНФОРМАТИКА**

Информатика

Наука об общих свойствах и закономерностях информации, методах ее поиска, передачи, хранения, обработки и применения в различных сферах деятельности человека

Термину **ИНФОРМАТИКА** предшествовал
термин **КИБЕРНЕТИКА**

Кибернетика (от греческого **kibernetike** - искусство управления) - название книги великого математика XX века **Норберта Винера**

Кибернетика - часть более общей науки **ИНФОРМАТИКИ**

Термин Информатика (informatique) происходит от французских слов:

- ***information*** (информация)

- ***automatique*** (автоматика)

дословно - ***"информационная автоматика"***

Англоязычный вариант информатики Computer Science

от английского слова ***compute*** – вычислять



Информатика

техническая наука, систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими

```
graph TD; A[Информатика] --- B[Практическая информатика]; A --- C[Теоретическая информатика]; A --- D[Техническая информатика];
```

Информатика

**Практическая
информатика**

**Теоретическая
информатика**

**Техническая
информатика**

Информатика

- Теоретическая информатика (brainware, "мозговое" обеспечение) изучает теоретические проблемы информационных сред
- Практическая, прикладная информатика (software, "гибкое", программное обеспечение) изучает практические проблемы информационных сред
- Техническая информатика (hardware, "тяжелое", аппаратное обеспечение) изучает технические проблемы информационных сред

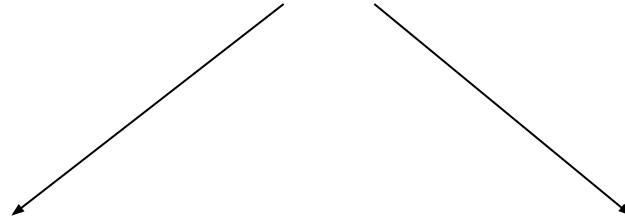
Пример применения информатики

- Задача построения математической модели прогноза кредитного риска банка – это задача теоретической информатики и экономики
- Построение алгоритма прогноза по этой модели – задача теоретической информатики
- Разработка компьютерной программы (комплекса программ) для прогноза риска – задача практической информатики
- Подготовка технического оснащения для отладки и обработки компьютерной программы - это задача технической информатики

Информатика включает в себя

- **теоретическую информатику (в том числе математическую логику и теорию информации);**
- **кибернетику;**
- **программирование;**
- **информационные системы;**
- **вычислительную технику;**
- **средства телекоммуникации, в том числе, глобальные компьютерные сети;**
- **проблемы создания искусственного интеллекта**

ИНФОРМАТИКА



Технические средства
Hardware - "твёрдые изделия".

Программные средства
Software - "мягкие изделия"



Предмет информатики

- **Аппаратное обеспечение средств Вычислительной Техники**
- **Программное обеспечение средств ВТ**
- **Средства взаимодействия аппаратного и программного обеспечения**
- **Средства взаимодействия человека с аппаратными и программными средствами**



Интерфейс

Средства взаимодействия в информатике принято называть интерфейсом

- средства взаимодействия аппаратного и программного обеспечения называют аппаратно-программным интерфейсом
- взаимодействие человека с аппаратными и программными средствами называют интерфейсом пользователя

Направления практического применения информатики

- **архитектура вычислительных систем** - приемы и методы построения систем, предназначенных для автоматической обработки данных;
- **интерфейсы вычислительных систем** - приемы и методы управления аппаратным и программным обеспечением;
- **программирование** - приемы, методы и средства разработки комплексных задач;
- **преобразование данных** - приемы и методы преобразования структур данных;
- **защита информации** - обобщение приемов, разработка методов и средств защиты данных;
- **автоматизация** - функционирование программно-аппаратных средств без участия человека;
- **стандартизация** - обеспечение совместимости между аппаратными и программными средствами, между форматами представления данных, относящихся к разным типам вычислительных систем.

Основная цель работы любого человека — максимально возможное повышение эффективности своего труда, труда своих коллег и подчиненных в той или иной прикладной области

- ***Прикладная область*** (предметная область) есть сфера практической деятельности, включающая в себя материальные объекты, информационные ресурсы и технологии

Примеры: предоставление образовательных услуг, медицинское обслуживание, услуги в области страхования, производство и реализация товаров и т.п.

Любая прикладная область включает в себя ***информационные технологии***

Информационная технология

Есть **процесс**, использующий совокупность **средств** и **методов сбора, обработки и передачи данных** (первичной информации) для **получения информации нового качества** о состоянии объекта, процесса или явления (информационного продукта)

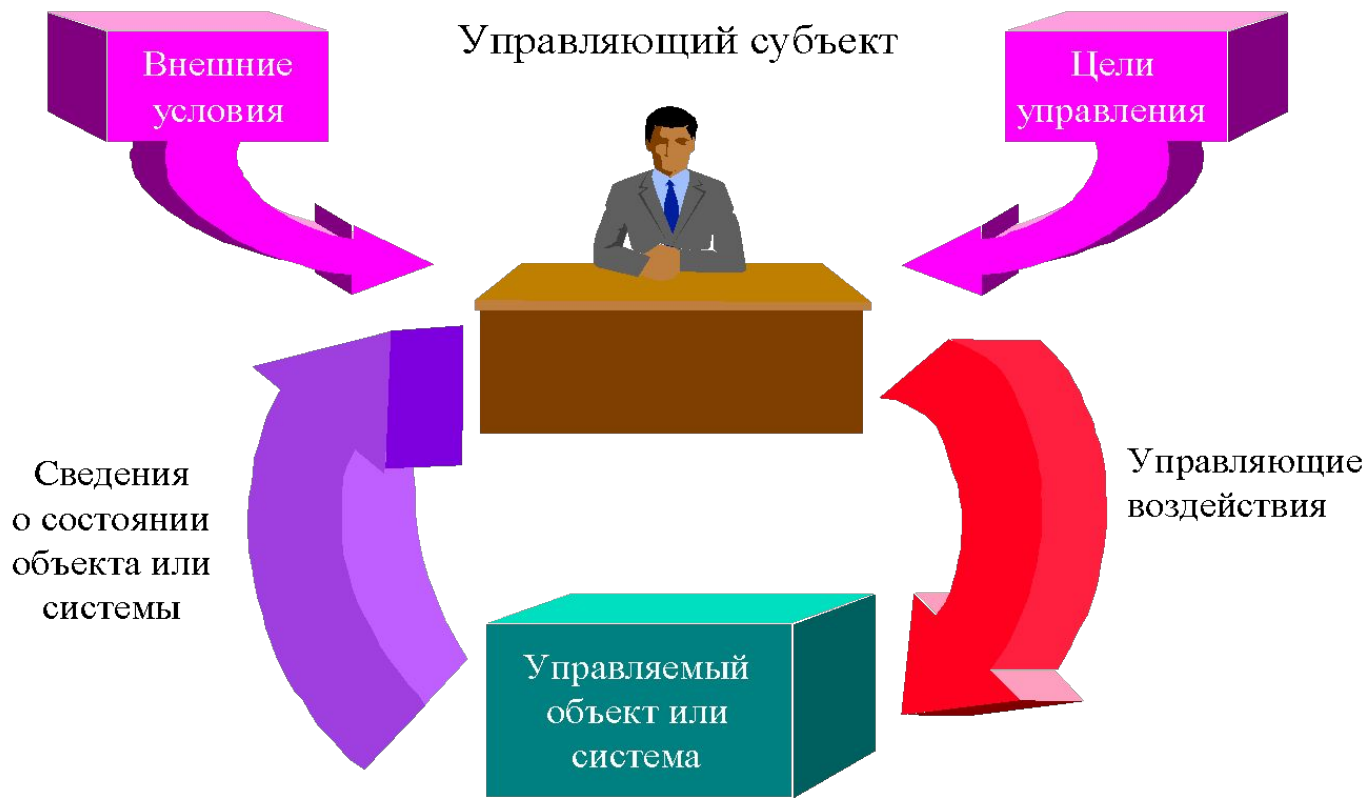
Примеры: ведение бухгалтерского учета, учет успеваемости студентов, делопроизводство и т.п.

Важнейшей характеристикой любого технологического процесса — и материального, и информационного, — является его **производительность**

Производительность технологического процесса

Производительность технологического процесса (информационной технологии) есть величина, определяющая **отношение величины полученного продукта к затратам времени на его производство**

Примеры: рассылка писем традиционным способом или с помощью электронной почты, создание баланса вручную или с помощью компьютерных технологий и т.п.



Эффективность технологии есть отношение стоимости информационного продукта к накладным затратам на его производство

Особенность современных информационных технологий заключается в том, что в них предметом и продуктом труда является информация, а орудиями труда служат средства вычислительной техники и связи

Информационные технологии дают возможность человеку получать сведения о событиях не только в данном месте и настоящем времени, но и в других местах и прошлом времени

- **Первое - информацию о событиях в других местах обеспечивают средства связи**
- **Второе - информацию о событиях в прошлом времени - физические тела - носители информации или устройства памяти (камень, бумага, книга, грампластинка, фотография, киноплёнка, магнитная плёнка, дискета, компакт-диск, карта флэш-памяти и др.), в которые эта информация вносится и сохраняется во времени (запоминается с целью последующего воспроизведения)**

Информационные технологии

- **Ручные** – инструментарий: перо, чернила и т.п.
- **Механические** - инструментарий: пишущая машинка, телефон и т.п.
- **Автоматизированные** - инструментарий: использование средств вычислительной техники и человека
- **Автоматические** - инструментарий: использование средств вычислительной техники без участия человека

Информационные технологии делятся на аналоговые и цифровые

- **Аналоговые технологии** основаны на способе представления информации в виде какой-либо непрерывной (аналоговой) физической величины

Например, напряжение или амплитуда электрического тока, величина которых (сигнал) является носителем информации

- **Цифровые технологии** основаны на дискретном (от лат. discretus - разделенный, прерывистый) способе представления информации в виде чисел (обычно с использованием двоичной системы счисления)

- В стремительном развитии радиотехники и вычислительной техники сыграли главную роль два изобретения:
 - вакуумных электронных ламп в 1905-1907 гг.
 - полупроводникового транзистора в 1948 году
- В результате изобретения электронных ламп сформировалась технология приборов вакуумной электроники, появились заводы по производству таких приборов, положившие начало развитию электронной промышленности. До 1960-х гг. вакуумная электроника представляла практически всю электронику
- Изобретение полупроводникового транзистора вызвало бурный рост микроэлектроники, отказ от использования электронных ламп
- Изобретение жидкокристаллических дисплеев и светочувствительных приборов с зарядовой связью (ПЗС) в 1970-х гг. позволили создать целую серию современных портативных устройств. В результате созданы цифровые наручные часы, мобильные телефоны, цифровые фото- и видеокамеры, ноутбуки, карманные компьютеры и т.п.

- **Появление компьютеров - новая эра информационных технологий: цифровая**
- **Внедрение компьютеров практически во все сферы жизни послужило появлению термина "информационные технологии"**
- **Философу Френсису Бэкону принадлежит высказывание:
«Кто владеет информацией - владеет миром»**

В наши дни это высказывание становится все более актуальным

Информационный кризис

- в 80-х годах XX века набрал полную силу информационный кризис
- **Информационный кризис** — явление, выражающееся в постоянно нарастающей мощности информационных потоков на фоне фиксированной производительности ИТ
- На первых порах для борьбы с последствиями информационного кризиса предлагались автоматизированные ИТ. Инструментарием таких ИТ были вначале малые, а затем — персональные ЭВМ, программный инструментарий составляют прикладные программы общего назначения

- **Программный инструментарий ИТ** — один или несколько взаимосвязанных программных продуктов для определенного типа компьютеров, работая с которыми пользователь может реализовать ИТ
- Автоматизированная ИТ не заменяет, а дополняет традиционную ИТ
- Пример: при внедрении автоматизированной технологии ведения деловой переписки с помощью текстового процессора механизм подготовки писем и их рассылки не претерпевает существенных изменений.
- В подавляющем большинстве случаев производительность и эффективность технологии тем выше, чем выше степень ее специализации

Информационная система

- Для обозначения инструментария сравнительно узко специализированной ИТ, ориентированной на выпуск четко определенной номенклатуры информационных продуктов часто используют термин **информационные системы**
- **Информационная система** — *человеко-компьютерная система для производства информационных продуктов, использующая компьютерную информационную технологию*
- **Информационная система** - взаимосвязанная совокупность средств, методов и персонала, используемая для хранения, обработки и вывода информации с целью решения конкретной задачи

Свойства основных понятий информатики

Информация

- **Информация** есть сведения, сообщения, известия, знания и умения, которые человек получает из окружающего мира
- **Информация** есть характеристика не сообщения, а соотношения между сообщением и его потребителем
- Без наличия потребителя, хотя бы потенциального, говорить об информации бессмысленно

Формы существования информации

- **в виде текстов, рисунков, чертежей, фотографий;**
- **в виде световых или звуковых сигналов;**
- **в виде радиоволн;**
- **в виде электрических и нервных импульсов;**
- **в виде магнитных записей;**
- **в виде жестов и мимики;**
- **в виде запахов и вкусовых ощущений;**
- **в виде хромосом, посредством которых передаются по наследству признаки и свойства организмов и т.д.**

Информационные объекты

- **Информационные объекты** (источник информации) есть предметы, процессы, явления материального или нематериального характера, рассматриваемые с точки зрения их информационных свойств



Свойства информации

- **достоверность;**
- **полнота;**
- **точность**
- **ценность;**
- **своевременность**
- **понятность;**
- **доступность;**
- **краткость**
- **объективность и субъективность;**
- **адекватность;**
- **актуальность и т.д.**

- **Информация достоверна, если она отражает истинное положение дел.** Недостоверная информация может привести к неправильному пониманию или принятию ошибочных решений
- **Достоверная информация со временем может стать недостоверной,** так как она обладает свойством **устаревать**, то есть **перестаёт отражать истинное положение дел**
- **Информация полна, если её достаточно для понимания и принятия решений.** Как неполная, так и избыточная информация **сдерживает принятие решений** или может повлечь **ошибки**
- **Точность информации** определяется степенью ее близости к реальному состоянию объекта, процесса, явления и т.п.

- **Ценность информации зависит от того, насколько она важна для решения задачи, а также от того, насколько в дальнейшем она найдёт применение в каких-либо видах деятельности человека**
- **Только своевременно полученная информация может принести ожидаемую пользу. Одинаково нежелательны как преждевременная подача информации (когда она ещё не может быть усвоена), так и её задержка**
- **Если ценная и своевременная информация выражена непонятным образом, она может стать бесполезной**

Действия, применяемые к информации

- **сбор данных** - накопление информации с целью обеспечения достаточной полноты для принятия решения;
- **формализация данных** - приведение данных, которые поступают из разных источников к единой форме;
- **фильтрация данных** - устранение лишних данных, которые не нужны для принятия решений;
- **сортировка данных** - приведение в порядок данных по заданным признакам с целью удобства использования;

Действия, применяемые к информации

- **архивация данных** - сохранение данных в удобной и доступной форме;
- **защита данных** - комплекс мер, направленных на предотвращение потерь и модификации данных;
- **транспортировка данных** - прием и передача данных между отдаленными пользователями информационного процесса. Источник данных принято называть сервером, а потребителя - клиентом;
- **преобразование данных** - преобразование данных из одной формы в другую, или из одной структуры в другую, или изменение типа носителя.

Единицы измерения информации

- Возможно ли объективно измерить количество информации?
- Важнейшим результатом теории информации является **вывод**:

В определенных, весьма широких условиях можно

- ✓ **пренебречь качественными особенностями информации**
- ✓ **выразить её количество – числом**
- ✓ **сравнить количество информации, содержащейся в различных группах данных**

Американский инженер Р. Хартли (1928 г.)

- Процесс получения информации есть выбор одного сообщения из конечного наперёд заданного множества N равновероятных сообщений
- Количество информации I , содержащееся в выбранном сообщении, определяет как двоичный логарифм N
- Формула Хартли: $I = \log_2 N$

нужно угадать одно число из набора чисел от 1 до 100 для этого требуется количество информации : $I = \log_2 100 = 6,644$. То есть сообщение о верно угаданном числе содержит количество информации, приблизительно равное 6,644 единиц информации
примеры **равновероятных сообщений**:

- при бросании монеты: "выпала решка", "выпал орел";
- на странице книги: "количество букв чётное", "количество букв нечётное"

Являются ли равновероятными сообщения ?

- "первой выйдет из дверей здания женщина"
- "первым выйдет из дверей здания мужчина"

Для задач такого рода **американский учёный Клод Шеннон предложил в 1948 г. другую формулу определения количества информации, учитывающую возможную неодинаковую вероятность сообщений в наборе.**

Формула Шеннона:

$$I = - (p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N),$$

где p_i — вероятность того, что именно i -е сообщение выделено в наборе из N сообщений.

Если вероятности p_1, \dots, p_N равны, то каждая из них равна $1/N$, и формула Шеннона превращается в формулу Хартли

Единицы измерения информации

- В качестве единицы информации условились принять один бит (англ. bit — binary, digit — двоичная цифра)
- **Бит в теории информации — количество информации, необходимое для различения двух равновероятных сообщений**
- В вычислительной технике битом называют наименьшую "порцию" памяти, необходимую для хранения одного из двух знаков "0" и "1", используемых для внутримашинного представления данных и команд

Единицы измерения информации

- 1 Байт = 8 бит
- 1 Килобайт (Кбайт) = 1024 байт
- 1 Мегабайт (Мбайт) = 1024 Кбайт
- 1 Гигабайт (Гбайт) = 1024 Мбайт
- 1 Терабайт (Тбайт) = 1024 Гбайт
- 1 Петабайт (Пбайт) = 1024 Тбайт
- 1 Экзабайт (Эбайт) = 1024 Пбайт
- 1 Зеттабайт (Збайт) = 1024 Эбайт
- 1 Йоттабайт (Йбайт) = 1024 Збайт

$$2^{10} = 1024$$

Алгоритм

- **Алгоритм есть точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи**
- **Исполнитель алгоритма есть некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом**

Исполнителя характеризуют:

- **Среда**
- **Система команд**
- **Элементарные действия**
- **Отказы**

Обычно исполнитель ничего не знает о цели алгоритма, выполняет все полученные команды, не задавая вопросов "почему" и "зачем"

- **Среда** (или обстановка) есть "место обитания" исполнителя

Например, для исполнителя "Водитель автомобиля" среда есть автомобиль. Дорога на которой расположен автомобиль тоже часть среды. А положение автомобиля на дороге задаёт конкретное **состояние среды**.

- **Система команд**. Каждый исполнитель может выполнять команды только из некоторого строго заданного списка — системы команд исполнителя
- Для каждой команды должны быть заданы **условия применения** (в каких состояниях среды может быть выполнена команда) и описаны **результаты выполнения команды**

Например, команда "движение вперёд" может быть выполнена, если перед автомобилем есть свободное пространство и горит зеленый свет светофора. Ее результат — движение в прямом направлении

- **Элементарное действие** исполнитель совершает после вызова команды
- **Отказы** исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды

Свойства алгоритма

- **Понятность** - все действия должны входить в систему команд исполнителя.
- **Дискретность** - алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов
- **Определенность** - каждый шаг алгоритма должен быть однозначным. Благодаря этому свойству выполнение алгоритма носит автоматический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче
- **Результативность** - алгоритм должен приводить к решению задачи за конечное число шагов
- **Массовость** - алгоритм выполняется для любых значений исходных данных. Алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применения алгоритма

Способы представления алгоритма

- **Словесный** – представление на естественном языке
- **Графический** - представление в виде блок-схемы
- **Псевдокод** - полужформализованное описание алгоритма на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и т.п.
- **Программный** - текст на языках программирования

Словесный способ представления алгоритма

- **Алгоритм представляет собой описание последовательных этапов обработки данных, изложенных на естественном языке**

Записать алгоритм нахождения **наибольшего общего делителя (НОД)** двух натуральных чисел.

Алгоритм может быть следующим:

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
3. определить большее из чисел;
4. заменить большее из чисел разностью большего и меньшего из чисел;
5. повторить алгоритм с шага 2

Словесный способ

Недостатки:

- описания строго не формализованы;
- имеют многословность записей;
- допускают неоднозначность толкования отдельных предписаний;
- не может быть выполнен компьютером

Преимущества

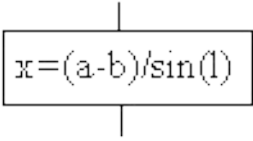
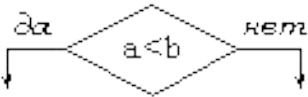
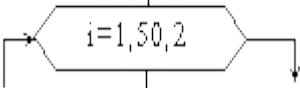
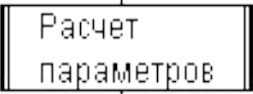
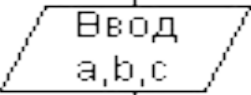
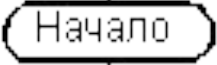
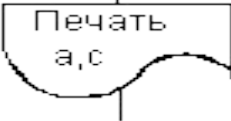
- Понятен человеку

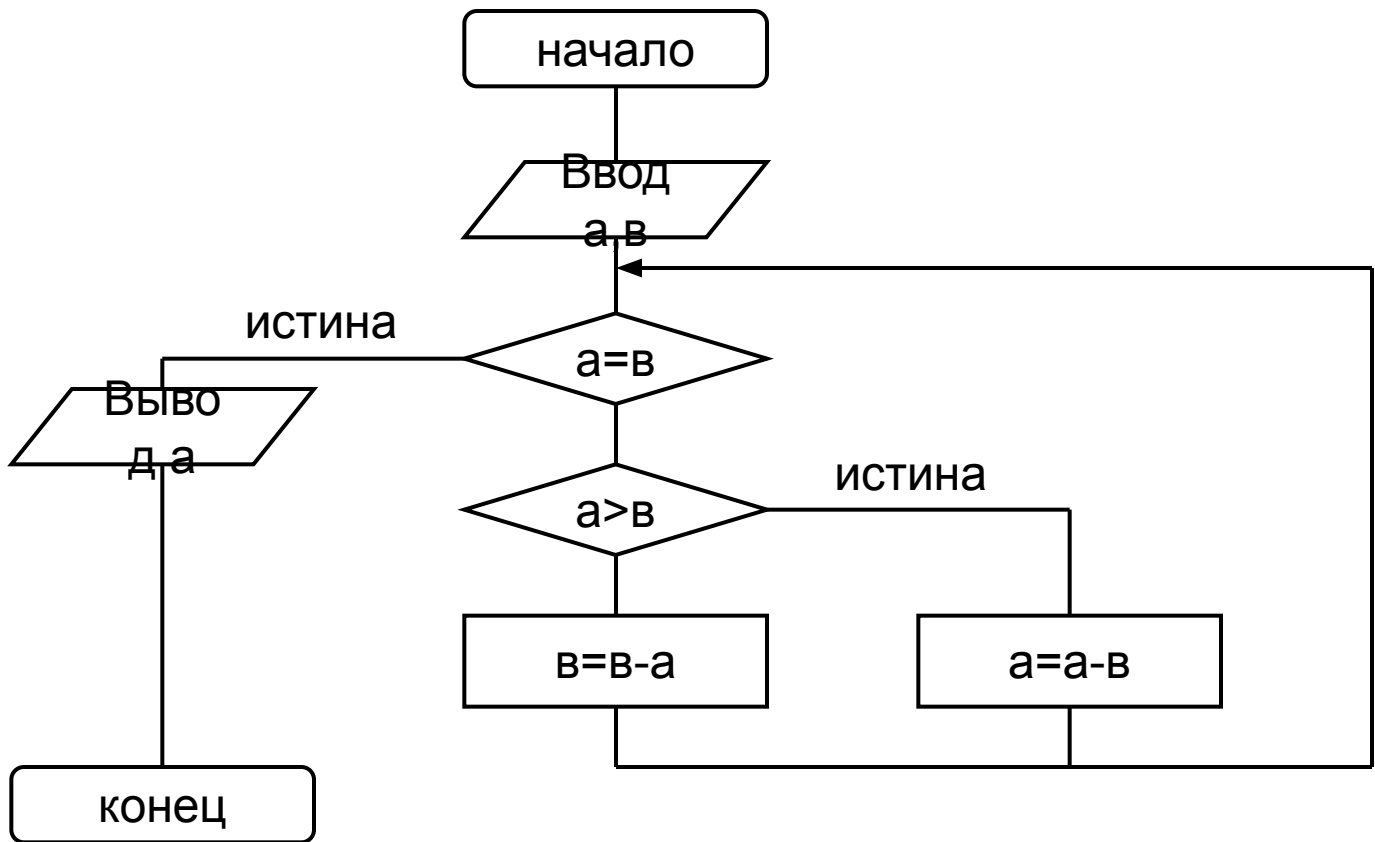
Графический способ

Алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий

Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий

Графическое представление называется схемой алгоритма или блок-схемой

Название блока	Обозначение	Пояснение
Процесс		последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Подпрограмма
Ввод-вывод		Ввод-вывод
Пуск-останов		Начало, конец алгоритма
Документ		Вывод на печать



Графический способ

Достоинства:

- Однозначность
- Точность
- Наглядность

Недостатки:

- Требуется подготовка для понимания
- Не может быть выполнен компьютером

Псевдокод

- Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов
- В псевдокоде не приняты строгие синтаксические правила для записи команд, а имеются некоторые конструкции (служебные слова), присущие формальным языкам

Примером псевдокода является школьный алгоритмический язык в русской нотации

Программный способ

Программа есть представление алгоритма на одном из языков программирования

В настоящее время в мире существует несколько сотен реально используемых языков программирования

Для каждого языка есть своя область применения

Программный способ

В зависимости от степени детализации предписаний определяется уровень языка программирования — чем меньше детализация, тем выше уровень языка

По этому критерию можно выделить следующие уровни языков программирования:

- машинные
- машинно-ориентированные (низкого уровня)
- машинно-независимые (языки высокого уровня)

Машинные языки

Каждый компьютер имеет свой машинный язык (машинных команд) - набор операций, которые может выполнить машина

Команды отличаются

- количеством операндов в команде
- назначением информации, задаваемой в операнде

Каждая команда при этом имеет вид последовательности нулей и единиц

Машинные языки

Достоинство

- Эффективное использование ресурсов компьютера
- Может быть выполнен компьютером

Недостатки

- Процесс написания программы на машинном языке очень трудоемкий и утомительный
- Программа получается громоздкой, труднообозримой, ее трудно отлаживать, изменять и развивать

Поэтому в случае, когда нужно иметь эффективную программу, в максимальной степени учитывающую специфику конкретного компьютера, вместо машинных языков используют близкие к ним машинно-ориентированные языки (ассемблеры)

Инструментальные языки и системы программирования

Служат для разработки новых программ на языке, понятном человеку (**инструментальный язык** или **алгоритмический язык программирования**)

Текст программы переводится (транслируется) в машинный код специальной программой, которая называется **транслятором**

Транслятор

- **Интерпретатор** читает один оператор программы, анализирует его и сразу выполняет, после чего переходит к обработке следующего оператора
- **Компилятор** сначала читает, анализирует и переводит на машинный код всю программу и только после завершения всей трансляции эта программа выполняется

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново *переводиться* на машинный язык при каждом очередном запуске программы

Уровни языков программирования

- Языки программирования низкого уровня
- Языки программирования высокого уровня

Языки низкого уровня

Язык программирование ассемблер

Язык ассемблера есть система обозначений, используемая для представления в удобочитаемой форме программ, записанных в машинном коде

Позволяет:

- использовать текстовый мнемонический (то есть легко запоминаемый человеком) код;
- присваивать символические имена регистрам компьютера и памяти;
- выбирать удобные для себя способы адресации;
- использовать различные системы счисления (например, десятичную или шестнадцатеричную) для представления числовых констант;
- использовать в программе комментарии и др.

Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая также называется ассемблером и является простейшим **транслятором**

Языки высокого уровня

- Языки **высокого уровня** намного понятнее человеку, нежели компьютеру
- Особенности компьютерных архитектур не учитываются
- Программы легко переносятся на другие компьютеры
- Разработка программ намного проще

Поколения языков программирования

- **Поколение 1** начало 50-х. первый язык ассемблера. Созданный по принципу “одна инструкция – одна строка”
- **Поколение 2** конец 50-начало 60

Разработан символический Ассемблер, в котором появилось понятие переменной. Это первый полноценный язык программирования

- **Поколение 3** -60-е годы Появились универсальные языки высокого уровня Появилась возможность решать задачи из любых областей. Многие языки применяются до сих пор

Поколения языков программирования

- **Поколение 4** с начала 70-х по настоящее время. Данные языки предназначены для реализации крупных проектов. Обычно ориентированны на специализированные области применения. Позволяют одной строкой создавать такую функциональность, для реализации которой в языках младших поколений требовалось создать тысячи строк исходного кода
- **Поколение 5** середина 90-х Системы автоматического создания прикладных программ с помощью визуальных средств разработки, позволяющие создавать программы пользователям не программистам

Обзор языков программирования высокого уровня

- **Fortran (Фортран)** - первый компилируемый язык , созданный в 50-е годы **Джимом Бекусом**. Удобство создания программ принесено в жертву возможности получения эффективного машинного кода
- **Cobol (Кобол)** -компилируемый язык для применения в экономической области и решения бизнес задач, разработанный в 60-х годах
- **Algol (Алгол)** Компилируемый язык, созданный в 1960 году. Алгол был призван заменить Фортран, но из более сложной структуры не получил широкого распространения.

Обзор языков программирования высокого уровня

- **Язык Pascal (Паскаль)** был разработан в 1970г. Никласом Виртом как язык обучения студентов программированию

Паскаль вырабатывает навыки соблюдения хорошего строгого стиля программирования (называемого структурным программированием), упрощающего разработку сложных программ.

В своем первоначальном виде Паскаль имел довольно ограниченные возможности, но расширенный вариант этого языка – **Turbo Pascal**, является очень мощным языком программирования

Обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью компилятора.

Обзор языков программирования высокого уровня

- **Язык Basic (Бейсик)** был создан в 1965 г. **Дж. Кемени и Т.Курцем** как язык для начинающих, облегчающий написание простых программ. Существуют сотни различных версий Бейсика – от очень простых до усовершенствованных, содержащих множество дополнительных языковых конструкций

Basic создавался как язык для начинающих программистов, для которых строчное выполнение программы имеет неоспоримые преимущества

Имеются и компиляторы и интерпретаторы

Обзор языков программирования высокого уровня

- **Язык C (Си)** (разработан **Деннисом Ритчи** в 1972 г.) соединяет свойства языка высокого уровня с возможностями эффективного использования ресурсов компьютера, которые обычно достигаются только при программировании на языке Ассемблера
Си не очень прост в изучении и требует тщательности в программировании, но позволяет создавать сложные и весьма эффективные программы
- C(Си)** создан в лаборатории Bell и первоначально не рассматривался как массовый. Планировался для замены ассемблера, чтобы иметь возможность создавать эффективные и компактные программы, но независимые от конкретного процессора

Обзор языков программирования высокого уровня

- В последнее время получили распространение системы программирования, ориентированные на создание *Windows-приложений*
- пакет **Borland Delphi (Дельфи)** — блестящий наследник семейства компиляторов **Borland Pascal**, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.
 - пакет **Microsoft Visual Basic** — удобный и популярный инструмент для создания *Windows-программ* с использованием визуальных средств. Содержит инструментарий для создания *диаграмм* и *презентаций*.
 - пакет **Borland C++** — одно из самых распространённых средств для разработки *DOS* и *Windows* приложений

Обзор языков программирования высокого уровня

- **Java (Джава, Ява)** Создан компанией Sun в начале 90-годов на основе C++

Призван упростить разработку приложений на C++ путем исключения низкоуровневых возможностей. Главная особенность компиляция не в машинный код, а в платформонезависимый байт код.

Особое внимание уделяется

- Поддержке всевозможных мобильных устройств и микрокомпьютеров, встраиваемых в бытовую технику
- Созданию платформо независимых программных модулей . способных работать на серверах в глобальных и локальных сетях

Языки программирования для Интернета

- HTML
- Perl
- Java Script
- VB Script

Системы программирования

- Исходный текст программы создается в текстовом редакторе
- Исходный текст с помощью программы компилятора переводится в машинный код. Результат промежуточный объектный код `.obj`
- Редактор связей или компоновщик связывает объектные модули и машинный код стандартных функций, находя их в библиотеках в исполняемый код

Интегрированные системы программирования

- **Текстовый редактор**
- **Компилятор**
- **Редактор связей библиотеки функций**

Turbo Pascal

ABTO

File Edit Search Run Compile Debug Tools Options Window Help

MYDD.PAS

```
[ ] uses crt;  
var  
  s,s1,s2 : String;  
begin  
  clrscr;  
  Write('Enter your name: ');  
  Readln(s1);  
  Write('Enter your family name: ');  
  Readln(s2);  
  s:=s1 + ' ' + s2;  
  Writeln('Good morning ',s);  
  Writeln('OK <Enter> to exit');  
  Readln;  
end.
```

15:37

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Turbo Pascal

ABTO

File Edit Search Run Compile Debug Tools Options Window Help

MYDD.PAS

```

var
  s,s1,s2 : String;
begin
  Write('Enter
  Readln(s1);
  Write('Enter
  Readln(s2);
  s:=s1 + ' +
  Writeln('Good
  Writeln('OK <
  Readln;
end.

```

Compiling

Main file: MYDD.PAS
Done.

Destination:	Disk	Line number:	0
Free memory:	256K	Total lines:	14

Compile successful: Press any key

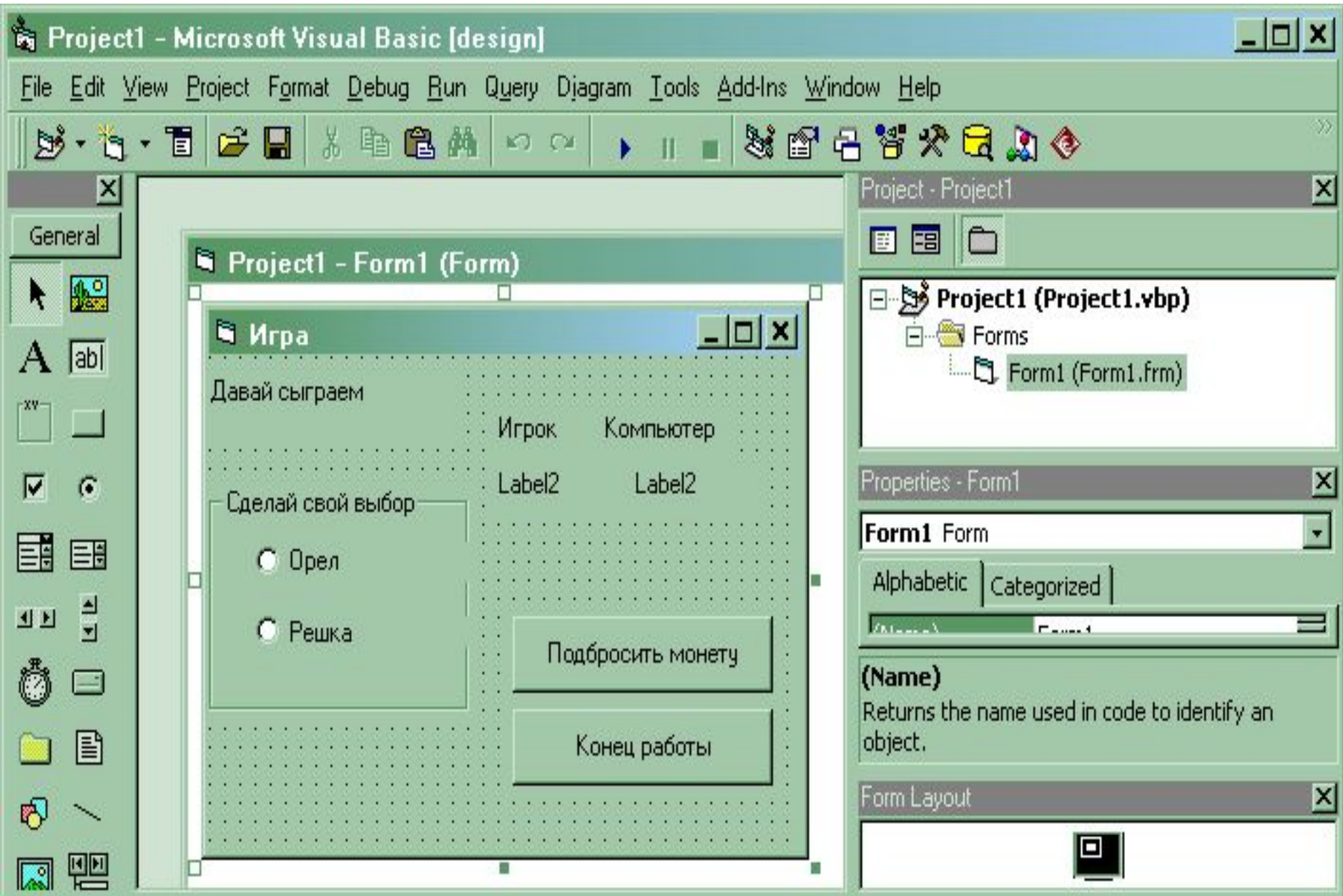
10:16

F1 Help | Compile source file

Turbo Pascal

АВТО

Enter your name: Natasha
Enter your family name: Chistuakova
Good morning Natasha Chistuakova
OK <Enter> to exit



Project1 - Form1 (Code)

Command1

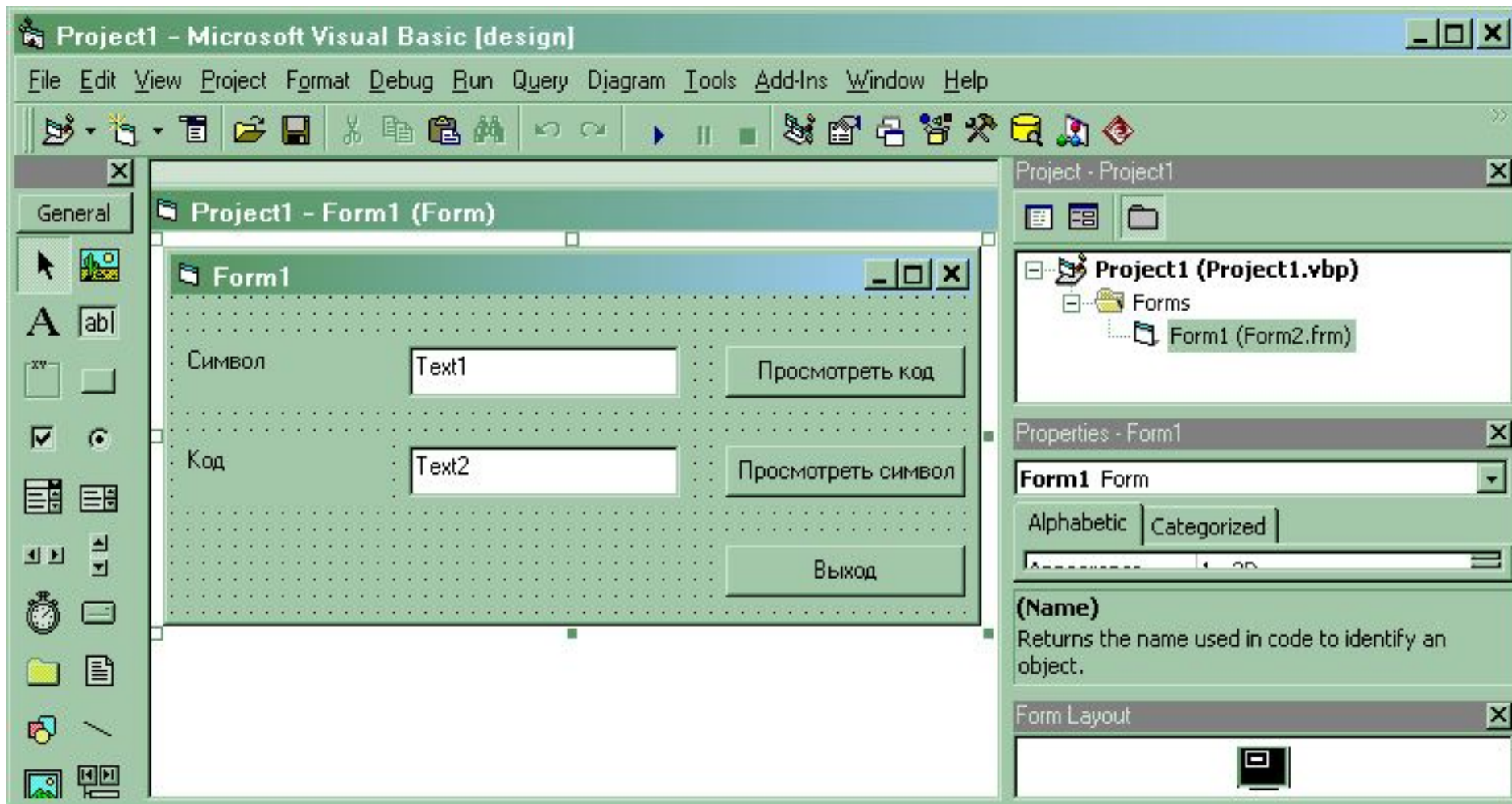
Click

```
Private Sub Command1_Click()  
Dim stra As String  
'stra = Me.Textbox1  
Me.Textbox2 = Asc(Me.Textbox1)  
End Sub
```

```
Private Sub Command2_Click()  
Dim inta As Long  
'inta = CLng(Me.Text2)  
Me.Textbox1 = Chr(Me.Textbox2)  
  
End Sub
```

```
Private Sub Command3_Click()  
'Me.Hide  
Unload Me  
End Sub
```

```
Private Sub Form_Activate()  
Me.Textbox2 = "65"  
Me.Textbox1 = "A"  
  
End Sub
```



Command2

Click

```
Private Sub Command2_Click()  
Randomize  
m = Int(Rnd * 2)  
c = c + 1  
If c >= 6 Then GoTo l11  
'MsgBox m  
If m = 1 Then  
    If Me.Option1.Value = True Then  
        MsgBox "Ура!!! Выигрывает.", vbExclamation, "ОРЕЛ"  
        a = a + 1  
    Else  
        MsgBox "Увы!!! Проигрывает.", vbCritical, "ОРЕЛ"  
        b = b + 1  
    End If  
Else  
    If Me.Option2.Value = True Then  
        MsgBox "Ура!!! Выигрывает.", vbExclamation, "РЕШКА"  
        a = a + 1  
    Else  
        MsgBox "Увы!!! Проигрывает.", vbCritical, "РЕШКА"  
        b = b + 1  
    End If  
End If  
Me.Label2.Caption = a  
Me.Label3.Caption = b  
Exit Sub  
l11:  
MsgBox "Игра окончена.", vbInformation, "Игра ОРЕЛ/РЕШКА"  
If a > b Then  
    Me.Label1.Caption = "Поздравляем с победой."  
Else  
    Me.Label1.Caption = "Повезет в другой раз."  
End If
```