

В широком смысле  
**информатика** есть наука о  
вычислениях, хранении и  
обработке информации.



Она включает дисциплины, так или иначе относящиеся к **вычислительным машинам**: так и абстрактные, вроде анализа алгоритмов, так и довольно конкретные, например, разработка языков программирования.

Отдельной наукой информатика была признана лишь в 1970-х;

до этого она развивалась в составе **математики**, электроники и других технических наук. Некоторые **начала информатики** можно обнаружить даже в **лингвистике**.

С момента своего признания отдельной наукой информатика разработала собственные **методы и терминологию**.



# Разделы информатики

Математические  
основы

Теория  
вычислений

Алгоритмы и  
структуры  
данных

Языки  
программирования  
и  
трансляторы

Базы данных

Конкурентные,  
параллельные  
и распределенные  
системы



## **Информатика – отнюдь не только “чистая наука”.**

У нее, безусловно, имеется научное ядро, но важная особенность информатики – широчайшие приложения, охватывающие почти все виды человеческой деятельности: производство, управление, науку, образование, проектные разработки, торговлю, финансовую сферу, медицину, криминалистику, охрану окружающей среды и др.

И, может быть, главное из них – совершенствование социального управления на основе новых информационных технологий.



# Готфрид Вильгельм фон Лейбниц

- Лейбниц также описал двоичную систему счисления с цифрами 0 и 1, на которой основана современная компьютерная техника.
- ***Можно условно сказать, что с него началась наука информатика.***



# Некоторых из ученых, чьи мысли явно присутствуют в современной информатике

- **А.А. Марков** (1903-1979) - основатель алгоритмики,
- **Э. Пост** (1879-1954) - американский логик, соединивший основания математики и основы новой техники вычислений,
- **Алан Тьюринг** (1912-1954) - английский математик, обосновавший возможности вычислительной техники,
- **Ада Лавлейс** (1816-1852) - первая в мире программистка,
- **Джон фон Нейман** (1903-1957) - основатель вычислительных структур, живущих по сей день,
- **С.А. Лебедев** (1902-1974) - русский основоположник всей вычислительной техники и создатель первой ВМ...

# Андрей Андреевич Марков (1903-1979)

- Основные труды по теории динамических систем, топологии, топологической алгебре, теории алгоритмов и конструктивной математике.



Рис. 28. А. А. Марков

# Эмиль Пост (1879-1954)



- Американский математик и логик. Читал лекции по математике и логике в Колумбийском, Нью-Йоркском и др. университетах США.
- Им получен ряд фундаментальных результатов в математической логике.



# Августа Ада Кинг (урожденная Байрон), графиня Лавлейс (1815-1852)

- Английский математик. Известна прежде всего созданием **описания вычислительной машины**, проект которой был разработан Чарльзом Бэббиджем. Составила **первую в мире программу** (для этой машины). Ввела в употребление термины **«цикл»** и **«рабочая ячейка»**.



# Сергей Алексеевич Лебедев (1902 -1974)



- **Основоположник вычислительной техники в СССР.**
- Российская академия наук учредила премию имени С. А. Лебедева — за выдающиеся работы в области разработок вычислительных систем.

## Информатика не более наука о компьютерах, чем астрономия — наука о телескопах.



**Эдсгер Дэйкстра**  
выдающийся нидерландский  
учёный

программирования Известность Дейкстре принесли его работы в области применения математической логики при разработке компьютерных программ. Он активно участвовал в разработке языка программирования Алгол и написал первый

компилятор Алгол-60. Будучи одним из авторов концепции структурного программирования, он проповедовал отказ от использования инструкции GOTO Известность Дейкстре принесли его работы в области применения математической логики при разработке компьютерных программ. Он активно участвовал в разработке языка программирования Алгол и написал первый компилятор Алгол-60. Будучи одним из авторов концепции структурного программирования, он проповедовал отказ от использования инструкции GOTO. Также ему принадлежит идея применения «семафоров Известность Дейкстре принесли его работы в области применения математической логики при разработке компьютерных программ. Он активно участвовал в разработке языка программирования Алгол и написал первый компилятор Алгол-60. Будучи одним из авторов концепции структурного программирования, он проповедовал отказ от использования инструкции

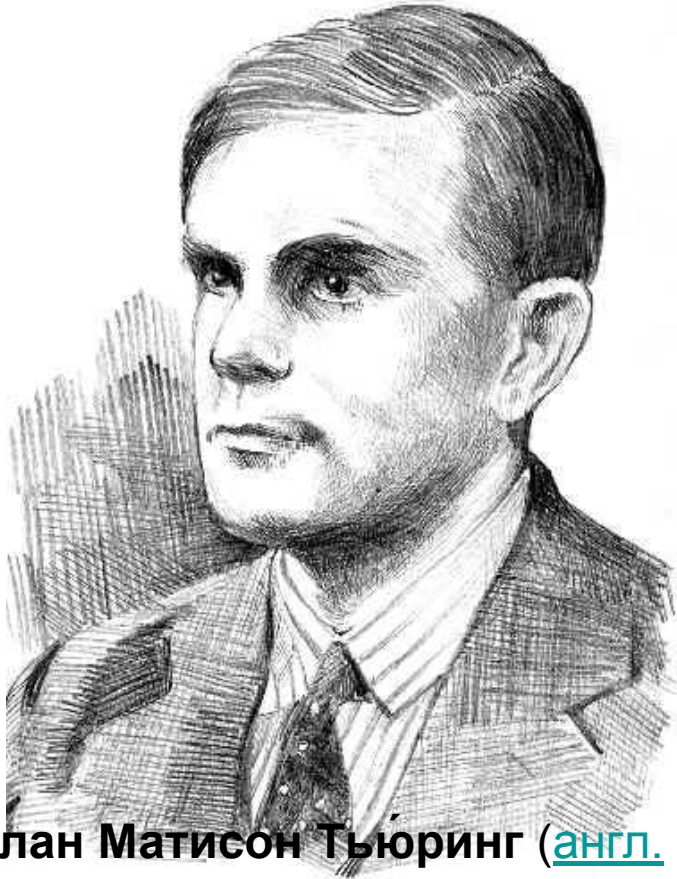
**Машина Тьюринга (МТ)** — абстрактный исполнитель (абстрактная вычислительная машина). Была предложена

Аланом Тьюрингом Аланом Тьюрингом в 1936 году Аланом Тьюрингом в 1936 году для формализации понятия алгоритма. Тьюринг является основателем

**теории искусственного интеллекта.**

Было обнаружено, что компьютеры всё-таки могут решить не любую математическую задачу. Алан Тьюринг доказал в 1936 году Было обнаружено, что компьютеры всё-таки могут решить не любую математическую задачу. Алан Тьюринг доказал в 1936 году, что общий алгоритм для решения проблемы остановки для любых возможных входных данных не может существовать.

Любая интуитивно вычислимая функция является частично рекурсивной, или, эквивалентно, может быть вычислена с помощью некоторой машины Тьюринга.



**Алан Матисон Тьюринг** (англ.

*Alan*) — английский математик) — английский математик, логик) — английский математик, логик, криптограф) — английский математик, логик, криптограф,

изобретатель машины Тьюринга

Высшей наградой за заслуги в области информатики является [премия Тьюринга](#).

Год	Лауреат	Причина присуждения
<a href="#">1966</a>	<a href="#">Алан Дж. Перлис</a>	<i>За его влияние в области обобщенных техник <a href="#">программирования</a> За его влияние в области обобщенных техник программирования и построения <a href="#">компиляторов</a></i>
<a href="#">1967</a>	<a href="#">Морис Винсент Уилкс</a>	<i>Профессор Уилкс более всего известен как проектировщик <a href="#">EDSAC</a> Профессор Уилкс более всего известен как проектировщик EDSAC, первого <a href="#">компьютера</a> Профессор Уилкс более всего известен как проектировщик EDSAC, первого компьютера, допускающего внутреннее хранение <a href="#">программ</a> Профессор Уилкс более всего известен как проектировщик EDSAC, первого компьютера, допускающего внутреннее хранение программ. Построенный в <a href="#">1949</a> Профессор Уилкс более всего известен как проектировщик EDSAC, первого компьютера, допускающего внутреннее хранение программ. Построенный в 1949. <a href="#">EDSAC</a> Профессор Уилкс</i>

<b><u>2002</u></b>	<u>Рональд Л. Ривест</u> <u>Шамир. Ади</u>	<i>За уникальный вклад по увеличению практической пользы <u>систем шифрования с открытым ключом</u></i>
<b><u>2003</u></b>	<u>Алан Кэй</u>	<i>За многочисленные новаторские идеи, которые легли в основу современных <u>объектно-ориентированных языков</u></i> <i>За многочисленные новаторские идеи, которые легли в основу современных объектно-ориентированных языков, руководство командой разработчиков языка <u>Smalltalk</u></i>
<b><u>2004</u></b>	<u>Винтон Серф</u> и <u>Роберт Кан</u>	<i>За пионерскую работу по проблеме межсетевого обмена включая разработку и реализацию основных <u>Интернет-протоколов</u></i> <i>За пионерскую работу по проблеме межсетевого обмена включая разработку и реализацию основных Интернет-протоколов, <u>TCP/IP</u></i>
<b><u>2005</u></b>	<u>Питер Наур</u>	<i>За фундаментальный вклад в проектирование языков программирования и создание языка <u>Алгол 60</u></i> <i>За фундаментальный вклад в проектирование языков программирования и создание языка Алгол 60, а также в проектирование <u>компиляторов</u>,</i>
<b><u>2006</u></b>	<u>Френсис Э. Аппен</u>	<i>За новаторский вклад в теорию и практику оптимизации компьютерных программ, послуживший основой для современных оптимизирующих компиляторов и</i>

# Как стать изобретателем?

Тщательный анализ жизненного пути многих изобретателей позволяет выделить шесть качеств творческой личности – минимально необходимый "творческий комплекс".

**Прежде всего нужна достойная цель – новая**  
(еще не достигнутая),

значительная общественно полезная

**Нужен комплекс реальных рабочих планов**  
достижения цели и регулярный контроль за  
выполнением этих планов

**Высокая работоспособность в выполнении**  
намеченных планов

**Хорошая техника решения задач**

**Способность отстаивать свои идеи -**  
"умение держать удар"

**Результативность**

# Теория Решения Изобретательских Задач ТРИЗ



Генрих Саулович  
Альтшуллер

ТРИЗ— теория решения изобретательских задач, основанная **Генрихом Сауловичем Альтшуллером** и его коллегами в **1946** году, и впервые опубликованная в **1956** году— это технология творчества, основанная на идее о том, что «изобретательское творчество связано с изменением техники, развивающейся по определенным законам»



# Как решать задачи с помощью ТРИЗ?

**Определите тип задачи**

```
graph TD; A[Как решать задачи с помощью ТРИЗ?] --> B[Определите тип задачи]; B --> C[Изобретательская задача]; B --> D[Исследовательская задача];
```

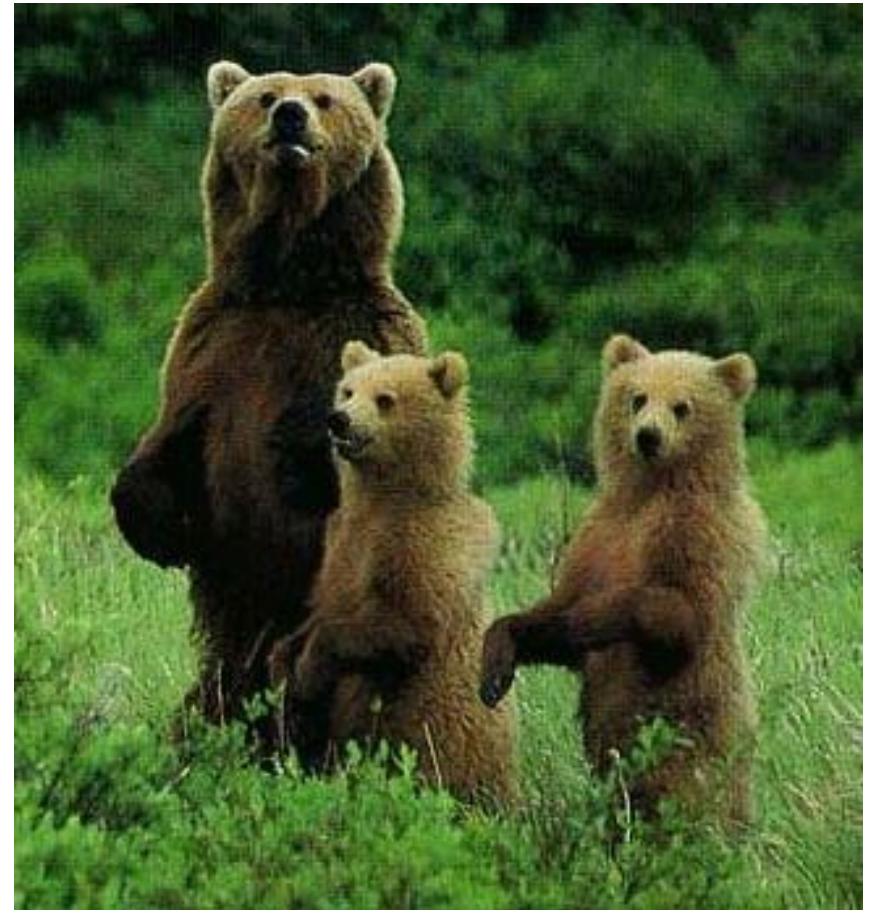
**Изобретательская задача**

**Исследовательская задача**

# Исследовательская задача

перед Решателем возникает вопрос:  
**«ПОЧЕМУ? КАК ПРОИСХОДИТ?»**

- Отправляясь на охоту, медведица оставляет своих малышей одних. А при ее возвращении медвежата ведут себя очень странно: едва завидев приближающуюся маму, они залезают на тонкие деревца. Почему?



# Изобретательская задача

перед Решателем возникает вопрос:  
**«КАК БЫТЬ?»**



- Медвежата плохо видят и не сразу узнают маму, возвращающуюся с охоты. Дождаться пока она приблизится — опасно, а вдруг это чужой взрослый медведь. Он ведь и обидеть может. Как быть медвежатам?

# Как решать задачи с помощью ТРИЗ?

Сформулируйте к задаче Противоречие,  
Идеальный конечный результат (ИКР)  
Идеальный конечный результат — это  
ситуация, когда нужное действие получается  
без каких-либо затрат (потерь), усложнений и  
нежелательных эффектов.

# Как решать задачи с помощью ТРИЗ?

## Выявите Ресурсы

*Принято классифицировать ресурсы следующим образом:*

- Материально-вещественные (вещества, предметы, товары, деньги, оборудование и т.д.);
- Информационные (каналы и носители информации).
- Ресурсы времени;
- Ресурсы пространства (площадь, объем и т.д.);
- Энергетические ресурсы и поля (тепловая, электрическая, электромагнитная, атомная энергия, звуковые сигналы и т.д.).

# Решить задачу

## Подсказка 1 [Идеальный результат]

Женщины сами снимают паранджу.

Издать указ, что паранджу можно носить только некрасивым и старым женщинам.

Женщина не должна снимать паранджу, а лицо при этом должно быть открытым. Сделать новые паранджи с вырезами для лиц.



# Решить задачу



Кинотеатр стал прогорать.  
Оказалось, зрители были

## Подсказка 1

Дама САМА хочет снять шляпу.

делала администрация кинотеатра.

## Подсказка 2

Измените ценность одетой во время сеанса шляпы или ценность человека, у которого одета шляпа.

# Решить задачу

Некоторые дети не желают ни учиться, ни гулять - весь день они сидят дома, "прилипнув" к экрану телевизора.

Как заставить детей проводить поменьше времени у «ящика»?





# Решить задачу



Очень часто одно и то же событие наблюдают журналисты различных новостных агентств. Чаще всего тексты передаются в сжатой форме по телефону. Хорошо теперь - под рукой мобильник и ноутбук. А если их нет? Простых телефонов сразу на всех не хватит.

Как раньше конкурентов передать новости? Ведь сенсация - хлеб журналиста.

# Решить задачу

Однажды за обедом кардинал Ришелье заметил, что один из гостей ковыряет в зубах острым кончиком ножа. Боже, как некультурно! Как бы другие не переняли дурную привычку.

Что же предпринял для этого "хранитель этикета"?



Желаю успехов в решении  
изобретательских задач!



# Математические основы

## Системы счисления

Целочисленные: двоичная, троичная, четверичная, ... .

Нецелочисленные: с основанием  $e$ , с другими основаниями.

## Криптография

Алгоритмы для защиты конфиденциальной информации, включают в себя шифрование.

## Теория графов

Основы структур данных и алгоритмов поиска.

## Математическая логика

Булева логика и другие способы моделирования логических запросов.

## Теория типов

Формальный анализ типов данных и использование этих типов для понимания свойств программ, в частности, их безопасности



# Теория вычислений

## Теория автоматов

Разные логические структуры для решения задач.

## Теория вычислений

Что можно вычислить, используя современные модели компьютеров.

## Теория сложности вычислений

Основные ограничения (в особенности время и размер для хранения данных) классов вычислений.



# Алгоритмы и структуры данных

## Алгоритмы

Формальные логические процессы, используемые для вычислений и эффективность этих процессов.

## Структуры данных

Организация и правила управления данными.

## Генетические алгоритмы

Генетические алгоритмы — это способ поиска приблизительных решений задач поиска и оптимизаций.



# Языки программирования и трансляторы

## Трансляторы

Способы трансляции Способы трансляции компьютерных программ, как правило, из языков программирования высокого уровня Способы трансляции компьютерных программ, как правило, из языков программирования высокого уровня в языки низкого уровня.

Компиляторы — трансляторы, которые проводят полную трансляцию всей программы (часто в несколько шагов) и на выходе выдают результат в виде исполняемого файла. После этого исполняемый файл можно запускать без повторной трансляции.

Интерпретаторы — трансляторы, которые проводят трансляцию программы шаг за шагом, и выполняют её также пошагово. При последующем запуске интерпретатору приходится выполнять все шаги интерпретации и исполнения заново.

## Языки программирования

Формальные парадигмы языков для выражения алгоритмов и свойств этих языков (к примеру, на каком языке лучше решить данную задачу).



# Базы данных

## Поиск данных

Изучение алгоритмов для поиска и обработки информации в документах и базах данных; тесно связан с информационным поиском.





# Конкурентные, параллельные и распределенные системы

## Конкурентность

Теория и практика одновременных вычислений; безопасность данных в любых многозадачных или многопоточных средах.

## Распределенные вычисления

Вычисления, которые используют несколько компьютерных приборов через сеть для решения общей задачи.

## Компьютерные сети

Алгоритмы и протоколы для надежной передачи данных через разные разделенные или выделенные ресурсы

## Параллельные вычисления

Вычисление, использующее несколько конкурентных потоков исполнения.

