

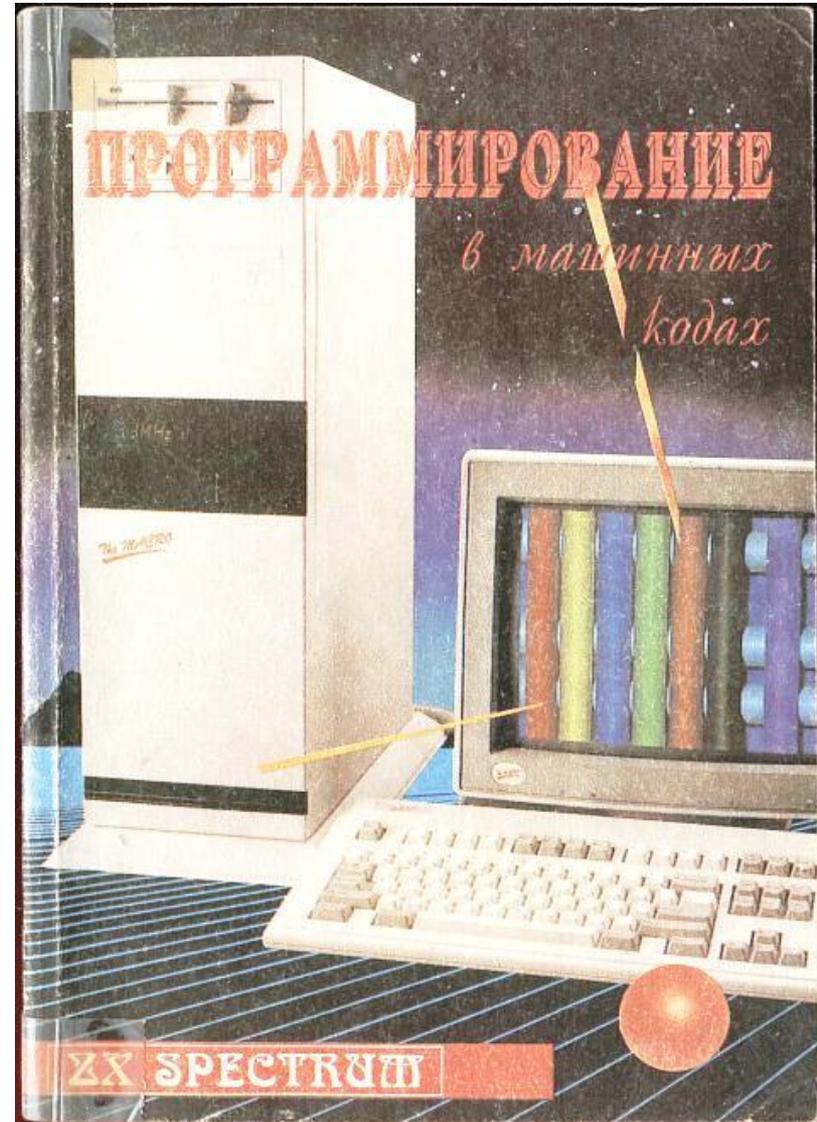


История развития языков программирования

Первые универсальные

Первые ЯЗЫКИ

программы писались на машинном языке. Программисты обязаны были знать архитектуру машины и её машинные коды. Программы были достаточно простыми из-за весьма ограниченными возможностями этих машин и большой



Но такой код разработки давал программисту невероятную власть над системой. Имелись возможности использования хитроумных алгоритмов и способов организации программ.

Например, могли применяться самомодифицирующиеся коды.

Двоичное представление команд позволяло иногда не

Базовые команды:

Арифметические – «сложения» и «вычитания»;

битовые – «логическое и», «логическое или», «логическое не»;

присваивание данных – «переместить», «загрузить», «выгрузить»;

ввода-вывода – обмен данными с внешними устройствами;

управляющие инструкции – «переход», «условный переход»,

«вызов подпрограммы», «возврат из подпрограммы».

Операции над числами (Р, Q, R – порядки чисел X, Y, Z по A ₁ , A ₂ , A ₃)						
Операция	КОП				ω=1	t мксек
	0чН	Б0	БН	Б0-БН		
Сложение	01	21	41	61	Z<0	285
Вычитание	02	22	42	62		
Вычитание модулей	03	23	43	63		
Деление	04	24	-	-	Y>10Q	~107
Умножение	05	25	45	65		51
Извлечение кв. корня	44	64	-	-		272
Выбод младших разрядов произведения	47					24

Первым значительным шагом является переход к языку ассемблера. Программисту не надо было больше вникать в способы кодирования команд на аппаратном уровне для каждого класса машин.

Появились возможности использования макросов (наборов отдельных команд) и меток (точек перехода), что упрощало создание, редактирование и отладку программ.





Однако переход к этому языку имел и отрицательные стороны:

- Возможности программистов сильно сократились;
- Впервые в истории развития программирования появились два представления программы:
 - в исходных текстах;
 - в откомпилированном виде.

К концу ассемблерной эпохи возможность автоматической трансляции в обе стороны практически исчезла. В связи с этим было разработано большое количество специальных программ-дизассемблеров, осуществляющих обратное преобразования, однако в большинстве



Следующий шаг был сделан в 1954 году, когда был создан первый язык высокого уровня — Фортран. Программист мог по-настоящему отойти от особенностей машинной архитектуры. Синтаксическая структура языка была достаточно сложна для машинной обработки в первую очередь из-за того, что пробелы как синтаксические единицы вообще не использовались. Это породило массу возможностей для скрытых ошибок.

Программы записывались на перфокартах и перфоленте, проводилось многократное исправление программ и их проверка.

Язык Фортран использовался для научных вычислений, компилятор практически никак не



Появление Фортрана вызвало еще больше критики, чем внедрение ассемблера. Со временем пришло понимание того, что реализация больших проектов невозможна без применения языков высокого уровня. Мощности вычислительных машин росли, требовались новые системы программирования ориентированные на различные области применения. В этих условиях преимущества языков высокого уровня стали настолько очевидными, что побудили разработчиков к созданию новых более совершенных языков :

- 1960 г. – создание языка Cobol
- 1960 г. – Петер Наур создал язык программирования Algol.

Бейсик (от BASIC, сокращение от англ. Beginner's All-purpose Symbolic Instruction Code — универсальный код символических инструкций для начинающих) — семейство высокоуровневых языков программирования.

Был разработан в 1964 году профессорами Дартмутского колледжа





Язык создавался как инструмент, с помощью которого студенты-непрограммисты могли самостоятельно создавать компьютерные программы для решения собственных задач. Получил широкое распространение в виде различных диалектов, прежде всего как язык для домашних компьютеров.

Это был язык нового типа – интерпретатор, который не создавал всю программу в машинных кодах, а интерпретировал лишь текущую строку программы, что делало отладку программ более простой.

На основе данного языка работает пакет MS Office



В 1970 году Никлаусом Виртом был создан язык программирования Pascal. Язык замечателен тем, что это первый язык с приемами структурного программирования.

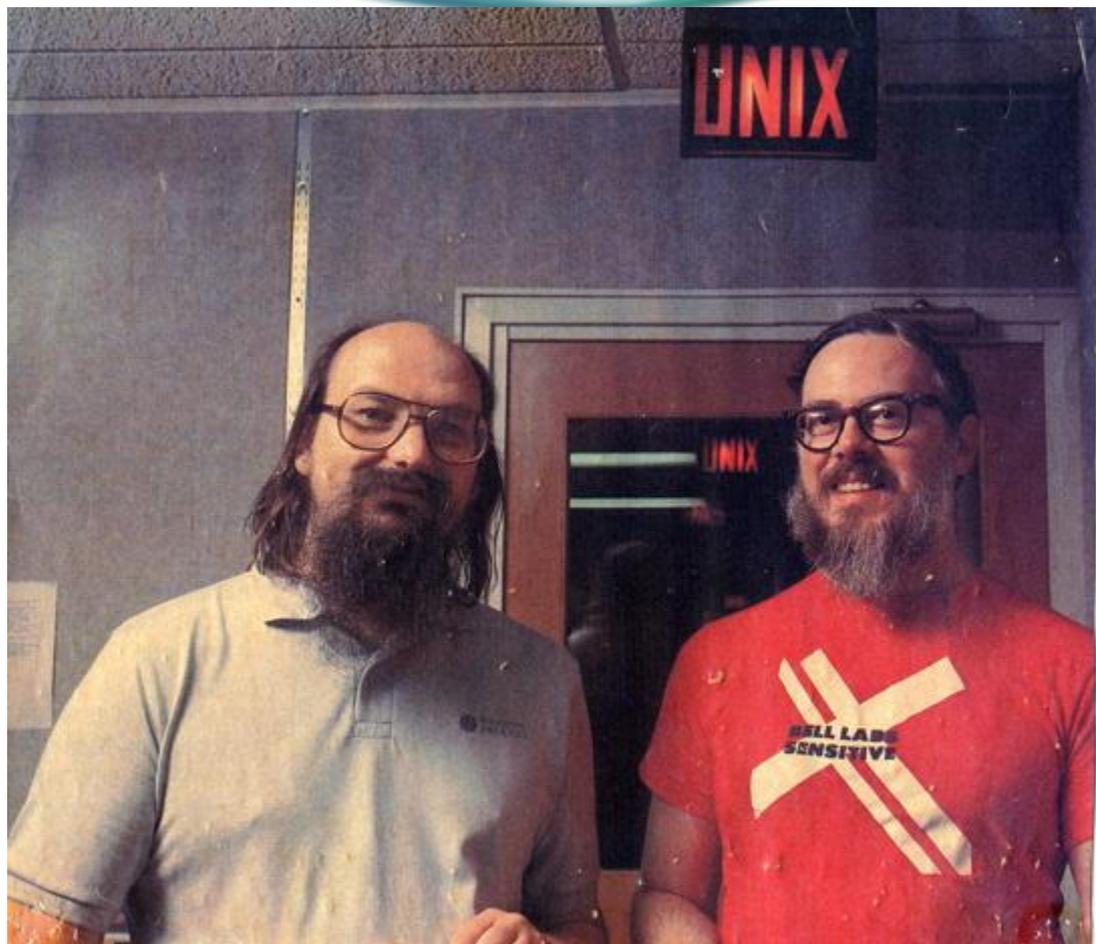
В языке внедрена строгая проверка типов, что позволило выявлять многие ошибки на этапе компиляции.

Язык получился очень популярным и широко используется до настоящего времени, как система обучения приемам программирования

В 1972 году Керниганом и Ритчи был создан Си.

Через 14 лет Бьярн Страуструп создал первую версию языка С++, добавив в язык объектно-ориентированные черты. Язык стал

основой для проектов. В 1999–2000 годах в корпорации Microsoft была разработана и создана версия языка С#. Он схож с Java (задумывался как современная альтернатива ему), но имеет и отличительные особенности. Ориентирован, в основном, на разработку мультимедийных Интернет





Возникаю и специализированные языки:

- **SETL** — язык для описания операций над множествами. Основной структурой данных в языке является множество, а операции аналогичны математическим операциям над множествами.
- **Perl** – язык создавался в помощь системному администратору операционной системы Unix для обработки различного рода текстов и выделения нужной информации. Развился до мощного средства работы с текстами.
- **Python** – интерпретируемый, объектно-ориентированный язык программирования. По структуре и области применения близок к Perl,

Структуры языков

- Алфавит языка – английские и иногда национальные символы;
- Специальные знаки – знаки арифметических и логических операций, работы с символьной информацией и т.д.
- Служебные слова – набор зарезервированных слов;
- Структуры – синтаксис описания структурных операторов;
Все это усложняет понимание языков программирования и основная современная задача
- Директивы компиляции – служебные символы, которые приводят к логическим трюкам, способам написания кода и использованию традиционных средств программ.



Все современные языки программирования структурированы и построены на объектах, что упрощает управление последними.

Например, нам не надо управлять экраном монитора, а достаточно послать туда нужную информацию в объект Screen, хотя мы имеем возможность и управлять им на уровне отдельных точек, но это потребует больших затрат при создании программы.

MS Office так же построен на основе объектов, что позволяет проще работать как с самим объектом в целом, так и с его отдельными компонентами.

Application (MS Excel)

Properties

Вид окна
Вид ленты
Настройки
И т.д.

Events

OnKey
OnTime
Quit
...

Methods

SheetActivate
WorkbookOpen
...

Collections

WorkBooks

Workbook1

Workbook2

Properties

Число листов
Диаграммы
Стили
И т.д.

Events

Open
Close
NewSheet
...

Methods

BeforeOpen
AfterOpen
Activate
...

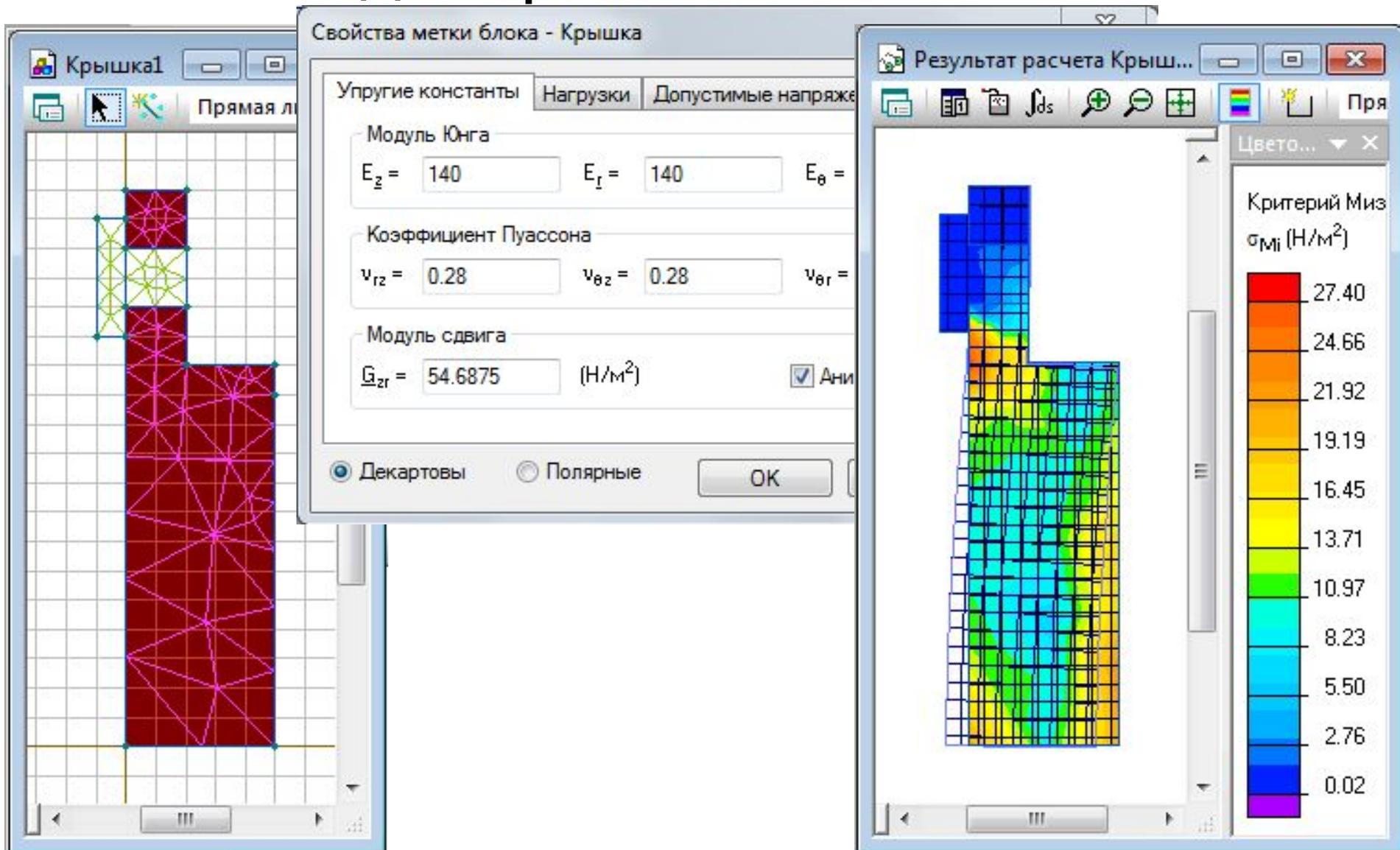


Сейчас возникают системы визуального программирования – способа создания программы путём манипулирования графическими объектами вместо написания её текста. Визуальное программирование часто представляют как следующий этап развития текстовых языков программирования.

Имеется много специализированных пакетов, где создание программы реализуется средствами построения графической схемы и заданием основных параметров для элементов этой схемы.

Например, любая SCADA система (от англ. supervisory control and data acquisition, диспетчерское управление и сбор данных) требует

EICut – моделирование с





Имеется целый класс языков, ориентированных на базы данных, которые используются в экономике, делопроизводстве, управлении и т.д.

Эти языки работают с данными типа записи (блоки одной длины, содержащие поля, которые имеют определенные типы данных) и позволяют вводить, редактировать, удалять записи, создавать запросы (выдирать наборы записей по определенным ключам), готовить отчеты (заполнять шаблоны отчетов необходимой информацией из полей отобранных записей).

Сейчас хорошо известны – Oracle, FoxPro, SQL, MS Access, есть русская версия Предприятие С1.

Наряду с этим направлением развития языков, ведутся работы по созданию программ и комплексов, которые могут принимать исходную информацию в близкой к обычной форме нашего общения и решать задачи по принципам работы нашего мозга.

Возникают направления в разработке программ искусственного интеллекта

Задачи искусственного интеллекта

ЭС

Решение традиционных задач с использованием традиционного программирования

Искусственный интеллект

Определение искусственного интеллекта (ИИ):

- Научное направление, в рамках которого ставятся и решаются задачи аппаратного или программного моделирования тех видов человеческой деятельности, которые традиционно считаются интеллектуальными.
- Наука под названием «Искусственный интеллект» входит в комплекс компьютерных наук, а создаваемые на её основе технологии к информационным технологиям. Задачей этой науки является воссоздание с помощью вычислительных систем и иных искусственных устройств разумных результатов деятельности.



Свойство интеллектуальных систем выполнять функции (творческие), которые традиционно считаются прерогативой человека. При этом интеллектуальная система — это техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти такой системы.

Структура интеллектуальной системы включает три основных блока — базу знаний, решатель и интеллектуальный интерфейс, позволяющий вести общение с ЭВМ без специальных программ для ввода данных.



Хорошим примером простейших программ ИИ являются экспертные системы (ЭС), примеры программ, литература можно найти на <http://www.aiportal.ru/downloads/>

Хорошая оболочка ЭС должна содержать:

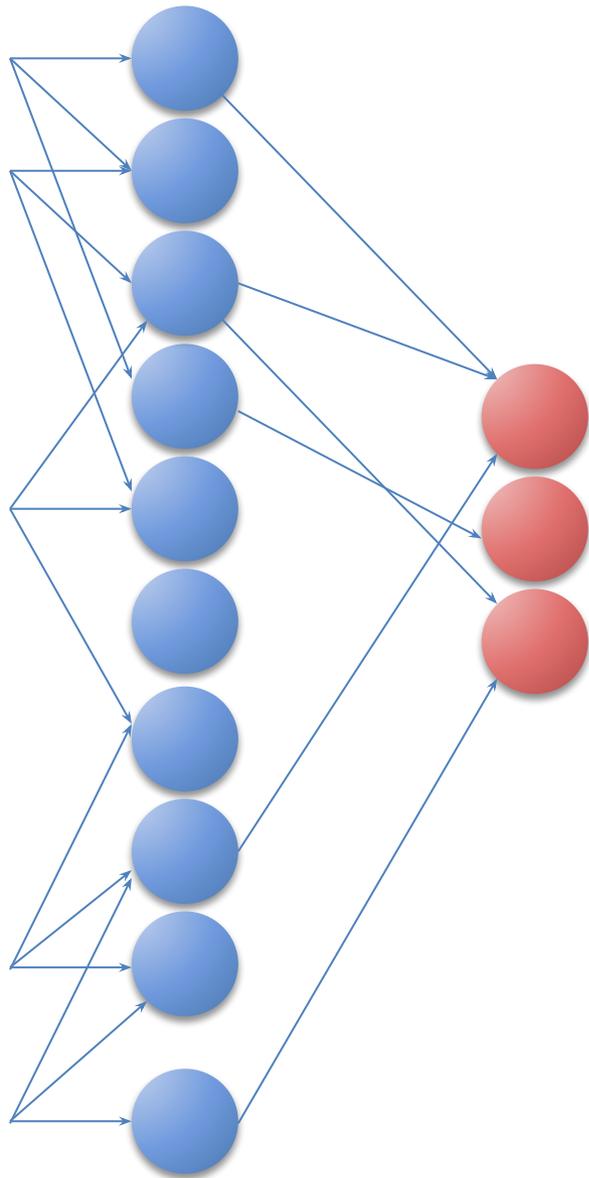
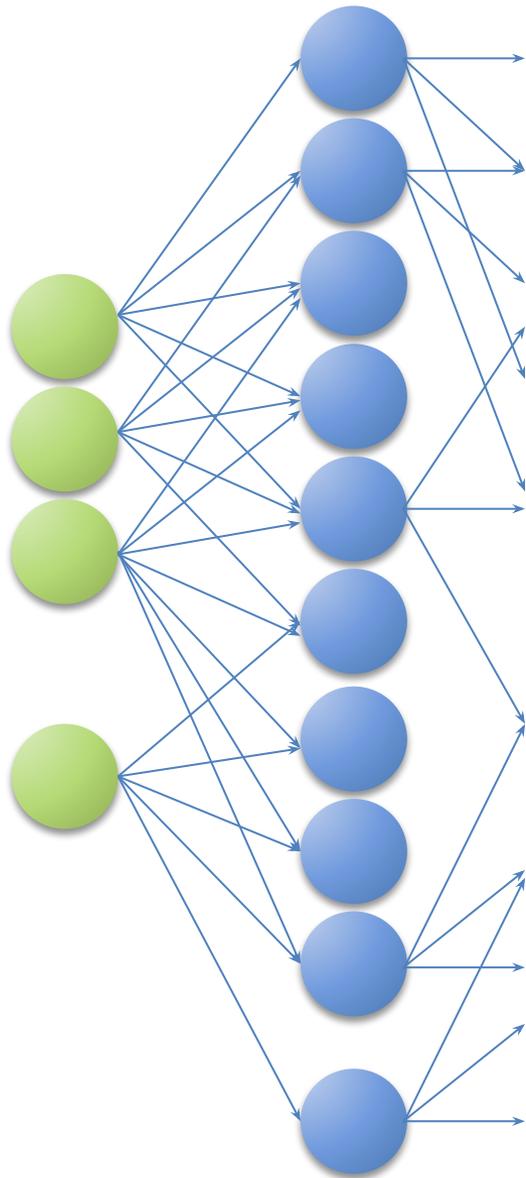
- Базу знаний в виде логических выражений, утверждений (вопросов), фреймов, семантических сетей;
- Систему общения с пользователем;
- Систему поиска результатов, включая эвристики;
- Систему самообучения.

К реальным программам ИИ можно отнести программы для игры в шахматы, шашки и т.д.

Нейронные сети

Искусственная нейронная сеть (ИНС) — математическая модель, а также её программная или аппаратная реализация, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять





Эволюционные

вычисления

Термин, используемый для общего описания алгоритмов поиска, оптимизации или обучения, основанных на формализованных принципах естественного эволюционного процесса.

Эволюционные методы предназначены для поиска предпочтительных решений и основаны на статистическом подходе к исследованию ситуаций и итерационном приближении к искомому состоянию систем.

Применяют следующие

- 1. Эволюционные методы оптимизации.** Основанный на идеях адаптации и эволюции. Степень мутации в данном случае меняется со временем – это приводит к, так называемой, самоадаптации.
- 2. Генетическое программирование.** Применение эволюционного подхода к популяции программ.
- 3. Эволюционное программирование.** Процесс построения программ строится как эволюция в мире программ.
- 4. Генетические алгоритмы.** Генетический алгоритм – это алгоритм, который позволяет найти удовлетворительное решение к аналитически неразрешимым или сложно решаемым проблемам через последовательный подбор и комбинирование искомых параметров с использованием механизмов, напоминающих биологическую эволюцию.
- 5. Теорию муравьиных колоний.** Эти алгоритмы основаны на применении нескольких агентов и обладают специфическими свойствами, присущими муравьям, и используют их для ориентации

Достоинства эволюционных вычислений:

- широкая область применения;
- возможность проблемно-ориентированного кодирования решений, подбора начальной популяции, комбинирования эволюционных вычислений с не эволюционными алгоритмами, продолжения процесса эволюции до тех пор, пока имеются необходимые ресурсы;
- пригодность для поиска в сложном пространстве решений большой размерности;
- отсутствие ограничений на вид целевой функции;
- ясность схемы и базовых принципов эволюционных вычислений;
- интегрируемость эволюционных вычислений с другими неклассическими парадигмами искусственного интеллекта, такими как искусственные нейронные

Недостатки эволюционных вычислений:

- эвристический характер эволюционных вычислений не гарантирует оптимальности полученного решения;
- относительно высокая вычислительная трудоемкость, которая однако преодолевается за счет распараллеливания на уровне организации эволюционных вычислений и на уровне их непосредственной реализации в вычислительной системе;
- относительно невысокая эффективность на заключительных фазах моделирования эволюции.

Генетические алгоритмы

Возникли в результате наблюдения и попыток копирования естественных процессов, происходящих в мире живых организмов, в время эволюции и связанного с ней естественного отбора популяций живых существ. Генетические алгоритмы применяются в системах искусственного интеллекта, оптимизации, искусственных нейронных сетях и в других отраслях знаний. Ранее эти задачи решались с помощью нейронные сети.

Генетическое программирование (ГП) — автоматическое создание или изменение программ с помощью генетических алгоритмов. С помощью этой методологии «выращиваются» программы, всё лучше