

**Открытая конференция ИСП РАН  
им. В.П.Иванникова, 2018**

**Доклад**

**«Информатика и ЭВМ-70лет. Аспекты  
развития»**

**Е.М.Лаврищева  
д.ф.-м.н., профессор, гнс. ИСП РАН**

## Информатика и Кибернетика-70 (1948)

В 1948 вышла книга Н. Винера «Кибернетика, или управление и связь в животном и машине» и ввела **кибернетику и информатику**. До этого уже сформировались идеи ЭВМ

- аналитическая машина Ч. Бэббиджа (1812) с арифметическими операциями над **информацией** (данными и словами).
- машина для **решения дифференциальных уравнений** в частных производных (1930) А. Розенблют, В. Буш.
- **ЭВМ ЭНИАК** (1946) Дж. Фон Нейман ввел понятие программы, в 1939г. Н. Атанасов из штата Айова сделал первый макет ЭВМ на электронных схемах и лампах, К. Цузе (1942г.) разработал проект ЭВМ, Г. Лейбниц усовершенствовал арифмометр Б. Паскаля и определил комбинаторику.
- **Аристотель** (IV век) сформулировал логику, закон исключения третьего и др.
- К. Шенон определил базовые элементы ЭВМ - электрические схемы из реле и математическую логику,
- **А. Никасима** и физики В. Шестаков, М. Гаврилов (1930) создали математический аппарат синтеза схем ЭВМ в разные структуры.

# Кибернетика

Н. Винер в своей книге дал определение <sup>1-2</sup>

**«Кибернетика - это наука об информации, управлении и связях в живой и неживой природе, ДНК структурах интеллекта и др.**

**Компьютер – инструмент этой науки».**

**Процессы обработки информации и управления ЭВМ должны основываться на элементах живого мира, выполнять арифметические операции над данными, отображать мыслительную деятельность человека средствами естественного (символического) языка и помогать исследованию ДНК и разума человека, развивая биологические и генетические области знаний.**

**Эта наука считалась лженаукой в СССР. И только в 1958 г. после выхода перевода книги Винера **кибернетика** и информатика были признаны нашими учеными (А.И.Берг, В.М.Глушков, Л.В. Канторович, А.И. Колмогоров и А.А. Ляпунов, Н.Винера привели в перспективе к переходу вычислительных машин к биологическим, медицинским, генетическим и др. системам.**

---

1. Н. Винер «Кибернетика, или управление и связь в животном и машине». Советское радио.-М.: 1961, 325с.

2. Д.А. Поспелов, Я.И. Фет. Очерки истории информатики в России. ISBN 5-7692-0101-0.-ОИГГМ СО РАН, 1998,662с.

## **Информатика**<sup>1-5</sup>

**Н. Винер**- «информатика - природный элемент», «информация, а не теория и энергия».

**Ф. Бауэр**<sup>2</sup> (1973) «**Информатика** – это компьютерная наука, которая включает технические связи для определения физиологии, психологии и нейробиологии и др.».

**Информатика** включает теорию кодирования, теорию информации, экстраполяции. Ядро информатики включает:

- цифровые и символьные вычисления, логическое исчисление, теории автоматов, алгоритмов, информационные и интеллектуальные систем.

Теория, понятия и техника первых ЭВМ способствовали образованию новых научных дисциплин – информатика и компьютерные науки (Computer Science). Они привели к образованию информационных и интеллектуальных систем и технологий, «умных машин» для решения математических задач по их алгебраической постановке в специальном символьном языке «Аналитик» на ЭВМ Мир 1-3 (1960 годы)<sup>3-5</sup>.

---

1. Норберт Винер. Кибернетика или управление и связь в животном и машине. New York-London 1948;

Советское радио. М.: 1961, втор. Изд. 1968. - 325с.

2. Ф. Бауэр, Г. Гооз. Информатика, М.: 1976. - 484 с.

3. Глушков В.М. Кибернетика, ВТ, Информатика. - Избранные труды в 3-х томах. – К.: Наук. думка, 1980, 262.- с, 267 с., 281 с.

4. Системы компьютерной алгебры семейства Аналитик. Теория. Реализация. Применение.- К.: 2010, ИММиС, Кол. Авторы, 762с.

5. Е.М. Лаврищева. Кибернетика, информатика и программная инженерия: аспекты развития.- К.: Пробл. Программирования, 2010. - с. 1-14.

# 1. Компьютерная, электронная эпоха XX века

**И. Брук и Р. Рамеев** создали первый вариант ЭВМ (1948).

В Англии сделана машина ЭДСАК и в 1951 г. **ЭВМ «УНИВАК»**, в Киеве в 1946-1951 гг. под руководством С.А. Лебедева была изготовлена первая в Европе **МЭСМ**, а в 1953 году ЭВМ «БЭСМ» в институте точной механики и ВТ.

В 1949г. выходит Постановление правительства СССР об использовании отечественных ЭВМ для решения технических и математических задач в оборонной, ракетной, авиационной промышленности, а также для решения задачи обработки информации и управления предприятиями.

В 1954 году под руководством М. Лесечко (Энергомаш СССР) сделана **ЭВМ «Стрела»**. В 1969-х созданы ЭВМ Днепр, УМШН, Мир 1, Минск-1, Наири и др.

## 1.1. Технологизация компьютеров

В 1960-е годы появились первые интегральные микросхемы и в 1964 году была изготовлена ЭВМ IBM/360, в которой начали использоваться интегральные микросхемы. В СССР первые интегральные микросхемы, как копии американских микросхем использовались в оборонной технике и в производстве универсальных ЭВМ (рис.1, 2). На пластинке (рис.3) даны транзисторы, которые вырабатывали электрический ток. На рис.4 приведены элементы молекулярного **нанокомпьютера**, размер которого соответствует нескольким нанометрам и предназначен для работы с мельчайшими атомами и молекулами близкими по толщине нити.

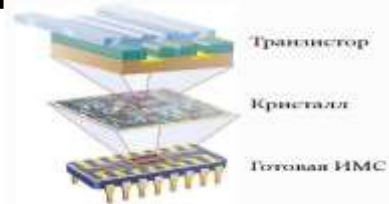


Рис.1



Рис.2



Рис.3

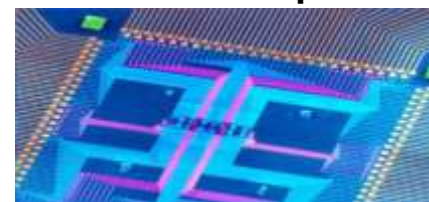


Рис.4

Появились чипы с поперечником 1 миллиметр для обеспечения взаимодействия компьютера с живыми клетками, на котором размещены 16384 транзисторов и сотни конденсаторов. Транзисторы получают сигналы, которые передаются клеткам с помощью конденсаторов. Был создан механизм **совмещения клетки с чипом** в Американской лаборатории Беркли и в Калифорнийском университете при проведении **генетических исследований клеток** молекул ДНК с целью создания **искусственных ДНК-систем**. Фактически они привлекли чипы в нейронную технику с живыми клетками.

## 1.2. Компьютеризация программ

Задачи для ЭВМ кодировались с помощью простых языков типа Автокода, схем Ляпунова, которые кодировались к машинному языку (M1, M2 и др.). К 1960 г. на отечественных ЭВМ (М-20, БЭСМ) стали использоваться языки программирования (ЯП) **Algol-60**, **Fortran**, **PL/1** и др.) для описания программ. Для их трансформации в код ЭВМ создавались **трансляторы с ЯП** (ТА1 – ТА4), отладчики и библиотеки программ и ОС. ЭВМ получили большую скорость и внешние устройства ввода и вывода информации.

На ЭВМ решались разные математические задачи, проводилось создание дискретных схем машин, задачи синтеза автоматов, обработка данных больших объемов и др.

По решению **ГКНТ СССР** были созданы Республиканские фонды алгоритмов и программ (РФАП), так называемые склады программ, и разработан метод сборки, объединения, интеграции программ, описанных в разных ЯП в большие системы для их выполнения в операционных средах ЭВМ. Одновременно считались математические, интеллектуальные задачи и доказательство теорем. **В 1975 г.** СССР перешел на IBM-360 = ЕС ЭВМ, которые изготавливались на заводах СССР и поставлялись по заявкам в разные города. Они имели большой объем памяти, систему разделения времени, высокое быстродействие и ОС.

## 2. Технологизация систем

### 2.1. Операционные системы для ЭВМ, М20, БЭСМ-6 и др.

1) **ОС- 68** или **Диспетчер 68**(Л.Н.Королев, В.П. Иванников, А.Н. Томилин и др. ) для БЭСМ выполняла:

- Мультипрограммное решение задач;
- Управление каналами связи, устройствами и ресурсами;
- Параллельное выполнение задач и работы устройств;
- распределение и буферизация ввода – вывода, диалоговый режим работы.

Диспетчер 68 (НД-70) управлял полетом космических аппаратов, баллистических и телеметрических программных комплексов более 20 лет. Дальнейшим развитием этой ОС является ОС для многомашинных комплексов АС-6, объединения и взаимодействия разных ЭВМ.

2) **ОС ИПМ для БЭСМ-6** функционировала с 1970 г. (И.Б. Задыхайло, С.С. Камынин и Э.З. Любимский). В этой ОС реализовано управление ресурсами со средствами синхронизации процессов и управления аварийными ситуациями и включена в нее система программирования и отладки программ на ЯП.

3) **Монитор - Дубна** (Н.Н. Говорун и В.П. Шириков) обеспечили управление заданиями, созданием библиотечных программ, в том числе Европейских программ CERN и трансляцию программ на языках программирования Фортран, Алгол, Паскаль и др. Был аппарат управления системой прерывания, защиты памяти, обеспечения режимов реального времени и разделения времени.



## 2.2. Системы программирования и Библиотеки программ

Программирующая программа для языка операторных схем Ляпунова А.А. сделана в 1953 г. в МГУ (Э.З. Любимский, А.П. Ершов) для М29. Позднее создавались трансляторы с языка Алгол-60 (ТА) для работы с программами на ЯП:

**ТА1**– С.С. Лавров (ЛГУ, 1962);

**ТА2** – М.Р. Шура-Бура и Э.З. Любимский (ИПМ, 1963);

**ТА3** - (Альфа-система) русская версия языка Алгол-60 (СО АН СССР, 1964);

**ТА4** – Е.Л. Ющенко, Е.М. Лаврищева для УВК «Днепр-2» (ИК АН УССР).

Транслятор с АЛМО (Любимский З.Л., 1965),

Система Algol-68 (Терехов А.Н., 1968). Транслятор с Кобола для ЭВМ «Минск1.2» и УВМ Д»непр-2» (1970).

**2.3. Метод сборки модулей (1975-1980)** через интерфейс с помощью 64 примитивных функций библиотеки **системы АПРОП**, преобразование разнородных типов данных в ЯП. Сформировано **сборочное программирование** в рамках ПВК комплексов - ПРОМЕТЕЙ, ЯУЗА, РУЗА и др. <sup>1-5</sup>. Система сборки АПРОП передана в ЕрНУЦ (Ереван, 1982). Основные разработчики этих комплексов были награждены премией Кабинета Министров СССР (1985). Система поставлена в 52 организации СССР и стала составной частью системы программирования IBM 360 (ЕС ЭВМ).

---

1. Лаврищева Е.М., Грищенко В.Н. Связь разноязыковых модулей в ОС ЕС.-М.:1982.-137с.

2. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование.-1991.-287с.

3. Липаев В.В., Штрик А.А., Позин Б.А. Технология сборочного программирования .-1992.-307с.

4. Орлов О.И. , Жоголев Е.А. Модульное программирование, М.: 1984.

5. Лаврищева Е.М. «Сборочное программирование. Основы индустрии программных продуктов».-2009.-471с.

## 2.4. Синтез программы, как способ манипулирования знаниями <sup>1-2</sup>

Синтезирующее программирование (Тыугу Э.Х.) — это процесс получения программы из условия задачи и метода ее решения. **Синтез программ** основан на логической и аналитической спецификациях и состоит в доказательстве теоремы существования и преобразования этих спецификаций в программу решения задачи. При логическом подходе спецификация трактуется как формулировка теоремы, утверждающей существование решения задачи. При аналитическом подходе спецификация трактуется как уравнение для символьного преобразования программы с проверкой правильности.

**Синтезирующее программирование** — это некоторый способ манипулирования знаниями, воплощенными в условии задачи предметной области и универсальным знанием, отражающим общематематические закономерности и сущность доказательного рассуждения по некоторой математической задаче.

---

1. Кахро М. И., Калья А. П., Тыугу Э. Х. Инструментальная система программирования ЕС ЭВМ (ПРИЗ). М.: Финансы и статистика, 1981.-157с.

2. Тыугу Э. Х. Концептуальное проектирование. М.: 1984.- 287 с.

**2.5. Композиция программ** - это операции объединения функций и данных вида: «данные–функция–имя» и «функция-композиция-дескрипция» на множестве именованных данных, дескрипций и денотатов (значений).

Редько В.Н. Композиции программ и композиционное программирование // Программирование. – 1978. – № 5. с. 17–26).

**2.6. Функциональное программирование РЕФАЛ** применяется для решения задач, связанных с распознаванием образов, общением на естественном языке, реализацией экспертных систем, доказательством теорем, СИМВОЛЬНЫМИ ВЫЧИСЛЕНИЯМИ и др.

Смирнов В.К. [Аппаратная реализация языка Рефал в ИПМ им.М.В.Келдыша](#) //Препринт ИПМ им. М.В.Келдыша. — 2003. — № 99. — С. 1-21.

**2.7. Расслоенное (аспектное ASPECTJ (<http://aspect.org>))программирование**

Технология рассредоточенных действий, технология вертикального СЛОЕНИЯ (Фуксман. А.Л. Технологические аспекты создания программных систем. М.: Статистика,1979.- 183с.) Сущность АОП – вставлять в готовый продукт операции защиты данными, взаимодействия с другими системами и др.

**2.8. Алгебраическое и агентное программирование (АП)**

Парадигма АП основывается на теории переписывания термов с помощью системы равенств алгоритма вычислений. В АП интегрируется процедурное, функциональное и логическое программирование. Используется граф термов для представления данных и знаний с помощью многоосновной алгебры данных (группы, кольца, поля). На основе АП сформировано *инсерционное (insort)* программирование. АП было реализовано на ЭВМ Мир 1-3<sup>1-2</sup>, а агентная теория развивается для принятия управленческих решений Э.А.Трахтенгерцем<sup>3</sup>

1. Капитонова Ю.В., Летичевский А.А. Методы и средства алгебраического программирования // Кибернетика. – 1993.– № 3. – С. 7–12.

2. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Языки, Алгебра, Программирования. Наука. 1974.-374с.

3. Транхтенгерц Э.А. Взаимодействие агентов в многоагентных средах.- Автоматика и телемеханика.- М.: 1998.- № 8.- с.3-52.

## 2.9. Графовое программирование

Теория графов в программировании начал развиваться в школе А.П. Ершова (Касьянов В.И, Иткин В.Э., Евстигнеев А.А. и др.). Были выпущены работы<sup>1-2</sup>. Эта теория активно развивается в ИСП РАН для схемной организации памяти в ОС Linux и используется в авиации, медицине и др.

1. Евстигнеев А.А. Применение теории графов в программирование. Москва.-Наука.- Редакция физ.-мат. наук, Под редакцией А.П. Ершова. 1985 г. 351с.
- 2.. И.Б. Бурдонов, А.С. Косачев, В.В. Кулямин Теория соответствия для систем с блокировками и разрушениями. Москва, Физматлит, 2008.-411с.

## 2.10. Технология программирования и конференции в СССР

Всесоюзные конференций ВКТП I-III (1982, 1985, 1992) от имени ГКНТ СССР и ИК АН УССР. Эти конференции собирали тысячи специалистов страны. На них выступали с научными докладами по технологии программирования: Ершов А.П., Липаев В.В., Лавров С.С. Шура-Бура М.Р. и др. Материалы конференций находятся в музеи ВТ (<http://www.computer-museum.ru>).

1. Глушков В.М. Фундаментальные исследования и технология программирования. Программирование. 1980г. №2.-с.3-13.
2. Вельбицкий И.В. Технология программирования. К.: Техника, 1984. 277 с.
3. ВКП-II. Технология программирования.-1986- М.; ГКНТ, ГКВТИ. Труды ВПК- II.-381с.

## 2.11. Зарубежные и отечественные технологии

1) АСКОН (1989) включает следующие элементы:

КОМПАС-3D для трехмерного моделирования на основе CAD / CAM / CAE;  
ЛОЦМАН:PLM – система управления данными и ЖЦ;  
ВЕРТИКАЛЬ – автоматизированное проектирование ТП.

2) ЗАО < Топ Системы > (1992) - адаптированный вариант T-FLEX документооборот;  
САМ – линейка программных продуктов (Project Studio CS) в AutoCAD для станков с ЧПУ; Autodesk Inventor для чертежей продуктов в ЕСКД и др.

### 3. Эпоха информатизации<sup>1-7</sup>

А.П. Ершов (1972г.) «Информация – это совокупность знаний о фактических данных, зависимостях между ними для познания и воспроизведения человеческого мышления в системе искусственного интеллекта (1972г.).

**Информатика** - это наука, изучающая структуру и общие свойства информации, закономерности процессов обмена, обработки, хранения, поиска и распространения научной, финансовой, экономической и другой информации. Информатика дает механизмы реформирования документов на естественных языках и создания информационно-поисковых, информационно-логических, интеллектуальных систем методами системного анализа и искусственного интеллекта.

**Информатика** рассматривалась как комплексная научная дисциплина, в состав которой входят теория проектирования и функционирования сложных компьютерных, информационных и интеллектуальных систем (баз знаний и данных), технологий программирования и ИТ(ИТ).

Впервые у нас термин информатика начал использоваться в 1960 г. в связи с теорией искусственного интеллекта, позволяющей моделировать умственную деятельность человека<sup>1-4</sup> в работах отечественных ученых - Велихова, Дородницина, Бауэра, Пospelова, Глушкова и др.

---

1. П. Велихов. Информатика – актуальное направление советской науки, 1960,

2. А.А. Дородницин. Информатика. Предмет и задачи, 1961.

3. В.С. Михалевич и др. Информатика – новая область науки и практики, 1961.

4. Ф.Бауэр, Г.Гооз. Информатика. М.: Перевод с нем. 1976.-484с..

5. В.М.Глушков. Основы безбумажной информатики, м.: 1982. 383с.

6. Г.И. Пospelов «Искусственный интеллект – новая информационная технология», 1993, и др.

7. Д.А.Пospelов, Я.И. Фет. Очерки истории информатики в России. СО АН СССР.-1998, 662с.

## 3.1. Информатика и Computer Science <sup>1</sup>

**Информационные технологии** (Information Technology) – это технологии управления решениями, бизнесом, коммерцией, экономикой, социальной сферой и др.).

**Информационные системы** (Information Systems) – это системы обработки информации (обмен информацией, документами, данными и др.).

**Программная инженерия** (Software Engineering) – методы, средства и инструменты разработки, эксплуатации, сопровождения и прекращения использования программ. Она занимает центральное место в пространстве информатики.

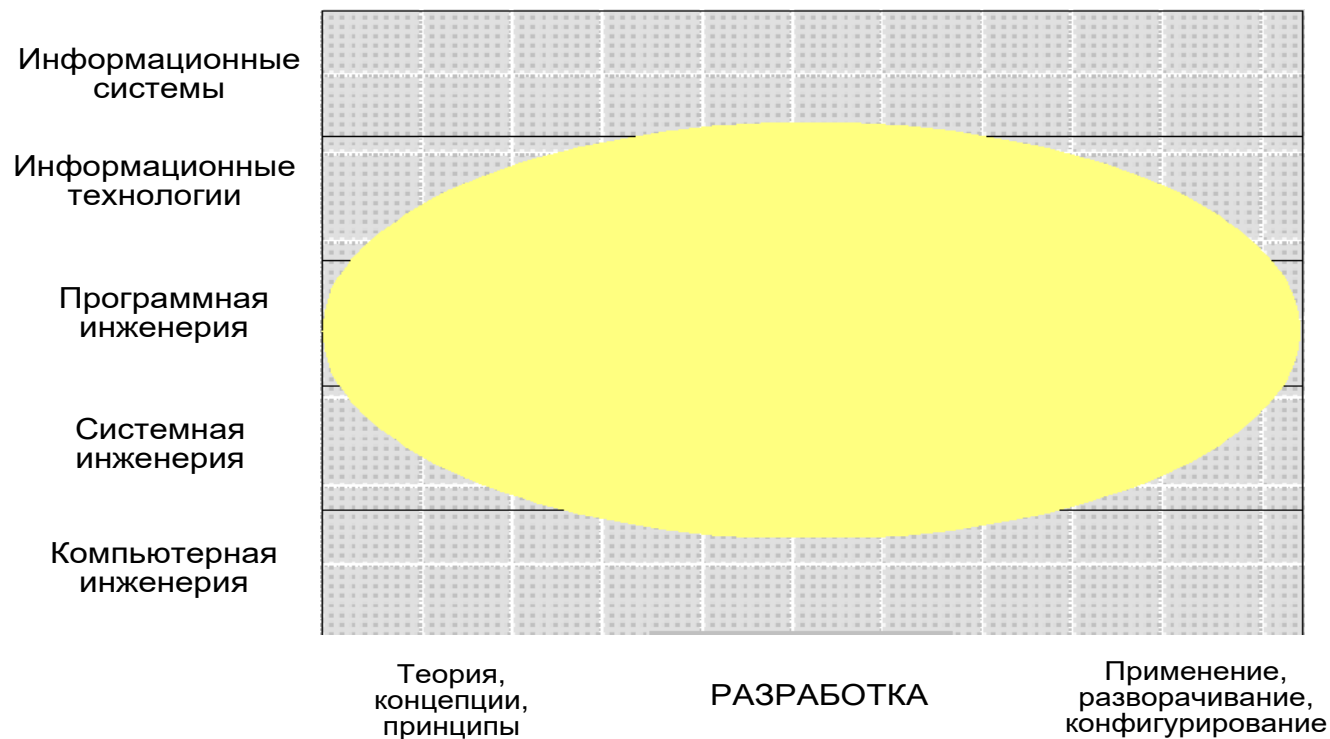
**Системная инженерия** (Engineering System) - это теория, методы и принципы построения ПО компьютеров (ОС, трансляторов, интерпретаторов, анализаторов, планировщиков и др.), обслуживания информационных и программных систем, управления данными и др.

**Компьютерная инженерия** (Computer Engineering) – это теория и принципы построения компьютеров, суперкомпьютеров, многопроцессорных и макроконвейерных машин, кластеров и др. Основу СЕ составляют теории Тьюринга, фон Неймана, автоматов, алгоритмов, математики, логики, анализа и др. и т.п.

---

1. Russell Shackelford, James H. Cross , Gordon Davies, John Impagliazzo and others. Серия «Учебные программы по информатике», Комиссия Учебных программ по информатике 2005.-70p. ACM, AIS, IEEE-CS, <http://www.acm.org/education/curricula.html>, <http://computer.org/curriculum>.

# Структура информатики в Computer Science



**Рис.6. Пространство информатики. Программная инженерия в центре информатики для создания систем и технологий**

### **3. 2. Структура информатики – Очерки истории СО АН(1998)<sup>1-2</sup>:**

- теория алгоритмов;
- логические модели;
- базы данных;
- искусственный интеллект (представление, использование знаний, ЭС, вывод, моделирование и др.);
- бионика (мат. модели в биологии, модели поведения, генетические системы и алгоритмы);
- распознавание образов и обработка зрительных сцен, 3-мерные сцены.....;
- теория роботов (представление знаний о мире, управление, планирование поведения и др.);
- теория компьютеров и вычислительных сетей(многоагентные системы, переработка информации и др.);
- теория математического обеспечения (ЯП, технологии создания программных, информационных , инструментальных систем, ОС и др.
- компьютерная лингвистика (модели языка, анализ, синтез тестов, машинный перевод, моделирование мыслительных процессов, работа с естественными языками и др.);
- нейроматематика, нейросистемы, нейросети (теория формальных нейронных систем, сетей, компьютеров и др.);
- модели реального мира, машинное обучение и др.

---

1. Д.А.Поспелов, Я.И. Фет. Очерки истории информатики в России. СО АН СССР.-1998, 662с.

2. Г.И. Поспелов «Искусственный интеллект – новая информационная технология», 1993



### 3. 3. Интеллектуальные системы <sup>1-5</sup>

Интеллектуальная система решает творческие задачи, знания о которых хранятся в ее памяти. Система содержит — **базу знаний, механизмы вывода решений и интеллектуальный интерфейс.**

Интеллектуальные системы изучаются средствами и методами искусственного интеллекта.

Основные задачи искусственного интеллекта:

- символьное моделирование мыслительных процессов,
- работа с естественными языками,
- представление и использование знаний,
- машинное обучение,
- биологическое моделирование интеллекта,
- робототехника.

Эти системы развивались с начала появления ЭВМ.

Ряд ученых проводят изучение этих проблем в течение многих лет<sup>1-3</sup>.

В период 1985-1990 годов вопросами интеллектуализации знаний и построения умных машин занимались специалисты Японии<sup>4</sup>. Но результатом этого проекта стали первые японские роботы.

---

1. Представление и использование знаний // Н. Ueno, V.Isudzuka. Москва-Мир.-1987. 220с.

2. Поспелов Г.С. Искусственный интеллект.- основа новой информационной цифровой технологии. Наука.- 1988.- 279 с.

3. Рыбина Г.В., Интеллектуальные системы от А до Я. Серия из трех книг. Книга 1.-2014, 224с.; книга 2 .-2015.-160с.; книга 3.-2015, 180с. М.: Науктехиздат.-2014.

4. Рыбина Г.В. Основы построения интеллектуальных систем. — М.: [Финансы и статистика](#); ИНФРА-М, 2010. — 432 с. — [ISBN 978-5-279-03412-3](#).

5. Г.С. Поспелов «Искусственный интеллект – новая информационная технология», 1993

### 3.4. Инженерия знаний

Знания представляются в Базе знаний (БЗ), содержащей правила вывода и **информацию** о человеческом опыте и знаниях<sup>1</sup> и стандарт ISO/IEC 2382-1:1993 и др.

На основе фактической информации БЗ и правил вывода делается автоматическое умозаключение об имеющихся или вновь вводимых фактах и осмысленной обработки информации<sup>2</sup>.

Современные БЗ работают совместно с системами поиска и извлечения информации, заданной онтологией в виде набора **понятий и отношений**.

Онтологии в Семантик Веб описываются в специальных языках **OWL, FODA, ODM, DSL**. Проведена разработка онтологии домена **ЖЦ ISO/IEC 12207-2007** в OWL, средствами Protégé 2.3 с выходным результатом в языке XML<sup>3</sup>.

Подход к автоматизации ЖЦ докладывался на Международной конференции **«Science and Information-2015»** в Лондоне и предложено IEEE сделать патент на идею интеллектуализации ЖЦ.

Онтологический подход набирает обороты Семантик Веб Интернет ([www.semantic\\_web.org](http://www.semantic_web.org)), как инструмент построения БЗ разных доменов.

---

1. ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.

2. Гаврилова и др. Базы знаний интеллектуальных систем // Учебник для вузов. — СПб.: Питер, 2000.

3. Lavrischeva E.M. [Ontology of Approach to the FORMAL Specification of The Standard Life Cycle](#), Science and Information Conference -2015”, July 28-30, 2015 London UK, [www.conference.thesai.org](http://www.conference.thesai.org), p.965-972.

4. Федоренко Д.Г., Астраханцев Н.А. Автоматическое извлечение новых концептов предметно-специфичных терминов. Труды ИСП РАН, том 25, 2013, стр. 167-178.

### 3. 5. Нано технологии в эпоху информатизации

Н. Винер сравнивал ЭВМ с **живой клеткой**. В ней молекула ДНК реализуются в долговременной памяти хромосома, состоящей из молекулы и РНК (информационных, транспортных и рибосомальных).

**Молекула ДНК – это** совокупность нуклидов, входящих в 46 хромосом клетки человека. Главные события клетки ДНК - деление или синтез.

Идею синтеза атомов клетки в макроатомы предложил Р. Фейнман (1959) с помощью программного сборщика. Он считал, что будут созданы миллионы миниатюрных заводиков, на которых "крошечные станки будут непрерывно штамповать маленькие детали" для маленьких приборов, собирая в них макромеханизмы, вещества и устройства. Эта идея привела к современной идее миниатюризации и получения новых макроэлементов для применения.

Маленькая частица названа **нано или «карлик»** с одним свойством - слипание (синтез) друг с другом, что приводит к образованию новых наночастиц или новых элементов.

Синтез молекул белков ДНК приводит к образованию комплексной наноструктуры с новыми специфическими свойствами<sup>1-3</sup>. Это свойство используется в медицине, биологии, авиации керамике, металлургии и др.

---

1. E.M. Lavrischeva, I.B. Petrov, Ways of Development of Computer Technologies to Perspective Nano, Future Technologies Conference (FTC-2017), 29-30 November 2017| Vancouver, Canada.-p.539-547.

2. Лаврищева Е.М. Программная инженерия Парадигмы, технологии, CASE-средства. 2 ред. Москва- Юрайт, 2017.- 284 с. (bibloo – Online.ru).

3. Лаврищева Е.М. Программная инженерия и технология программирования сложных систем. Учебник. 2-издание. Москва – Юрайт, 2018.- 431с. (bibloo – Online.ru).

**Нанотехнология** ориентированная на получение веществ и устройств с заранее заданной атомарной архитектурой (Э.Дрекслер).

**Атом – это  $10^{10} = 1$**  нанометра (нм), а бактерия это  $10^{-9}$  нм. Частицы от 1 до 100 нанометров называют наночастицами.

Один из важнейших вопросов нанотехнологии — это группирование молекул, маленьких элементов самоорганизовываться, чтобы получить новый материал или устройство. Это относится к белкам ДНК, которые могут образовывать комплексные структуры путем синтеза молекул.

В перспективе программные элементы малые по размеру будут синтезироваться к виду «маленьких частиц» с заданной структурой и функциями, образуя малые nano программы.

В 2010-х появились nano элементы - **кванторы, графены, кубиты** (МФТИ, МИСиС и др.). **Кубит** состоит из нескольких джозефсоновских контактных элементов, разделенных тонким слоем диэлектрика. Квантовые биты **кубита** способны выполнять вычисления, которые не доступны современным компьютерам.

При создании маленьких приборов компьютерного и нейрокомпьютерного типа используются маленькие элементы **чипы, транзисторы** и др. Из них собираются новые технические приборы и устройства, как на сборочном конвейере, и применяются в e-science (биологии, генетике, физике, медицине и др.). Они способствуют улучшению здоровья общества и жизни на земле.

### 3. 6. Методы синтеза ДНК<sup>1-4</sup>

В МФТИ разработан подход к синтезу молекул ДНК на основе теории распознавания Маркова в диссертации А.В.Островского<sup>1,2</sup>, представленной на сайте <http://7dragons.ru/bio>.

Метод синтеза молекул ДНК реализуется следующими операциями:

- определение фрагментов генов — экзонов и интронов с заданной последовательностью нуклеотидов гена;
- выявления принадлежности аминокислот белка к пространственным структурам — спиралям, листам и нерегулярным образованиям и др.
- проведение метода синтеза этих фрагментов ДНК для образования новых структур.

Эти задачи решаются методами, использующими скрытые Марковские модели и их обобщения. Математическая формулировка этих задач, алгоритмы их решения и доказательство их математической корректности при синтезе белков изложены в диссертации Островского.

---

1. Островский А.В. Методы распознавания ДНК на основе моделей Маркова со скрытыми переменными. – Автореф. диссертации.- ИК НАНУ.-23с.

2. Лаврищева Е.М. Программная инженерия и технология программирования сложных систем. Учебник. 2-издание. Москва – Юрайт, 2018.- 431с. (bibloo – Online.ru).

3. Цыганков В.Д. Вселенский разум и квантовый нейрокомпьютер.- Синтег. Москва- 2002.-171с.

4. Искусственный интеллект в технических системах. Государственный институт физико-технических проблем.- Сборник научных трудов под редакцией Лупичева Л.И.-1998.-153 с.

### 3. 7. Пути развития Интернет<sup>1-3</sup>

1. Web 4.0 Интернет будет обеспечивать взаимодействие участников сети с помощью **интеллектуальных агентов**.
2. Корпоративные решения для **облачных технологий** (PaaS, SaaS), будут сращиваться с Интернет-пространством и использоваться для управления адаптивными Web-приложениями, в которых облачные сервисы будут взаимодействовать через **веб-страницы агентов**.
3. **Internet of Things, Smart IoT** будут поддерживать конкурентные APPS через: распределенные microservices; Hypercat мобильной связи; GSM-R для цифровых дорог. Промышленный Интернет (Industrial Internet) ориентируется автоматизацию новых концептов - **«умная энергетика»**, **«умный транспорт»**, **«умные приборы»**, **«умная промышленность»**, **«умные дома»**, **«умные города и медицина»** и т.п.
4. Интернет 4.2 предоставляет технологическое **конфигурирование** и взаимодействие с устройствами обработки больших данных для превращения **Big Data в Smart Data**.
5. Семантика взаимодействия информационных объектов, которые задают **цифровые проекции** реальных или абстрактных объектов, будут взаимодействовать через **Web-сервисы** в языке OWL стандарта ISO 15926 Интернета 3.0.

---

1. E.M. Lavrishcheva, I.B. Petrov, Ways of Development of Computer Technologies to Perspective Nano, Future Technologies Conference (FTC-2017), 29-30 November 2017| Vancouver, Canada.-p.539-547.

2. Боровков А.И. Компьютерный инжиниринг Учебное пособие // Боровков А.И., Клямин О.А., Мельникова М.П. и др. /Computing Engineering.- San-Peterburg.-2012.- Россия, Полит. ул. 29.- 93 с.

3. Пройдаков Э.А. и др. Платформа «АН2», 2016, 139с.

## 4. Эпоха информатизации и рыночной экономики (1992-2018)

### Проект информатизации России 1992

В 1992 г. создан Научно-технический Совет «**Информатизация России**» для создания новых фундаментальных основ информатизации. В совет входили 30 ведущих специалистов от РАН (Иванников В.П., Липаев В.В. и др.) и от других НИИ России. Совет разработал основы информатизации России:

1. Экономические и социальные задачи информатизации.
2. Техничко-экономическое развитие рыночной экономики информатизации.
3. Научно-техническое обеспечение информатизации России (Фундаментальные разработки ИТ и систем); технология и программная инженерия разработки систем; Обеспечение защиты информации; Стандартизация средств создания систем; поддержка информатизации в образовании; развитие инфраструктуры информатизации России.
4. Развитие инфраструктуры информатизации России.
5. Системный проект информатизации России.

В 3.1 к научным направлениям информатизации отнесены (Иванников В.П.):

- алгебраические и логические теории формализации систем;
- анализ параллельных и распределенных систем;
- формальные спецификации и верификации сложных систем;
- парадигмы программирования;
- формализмы представления знаний;
- модели человеко-машинных интерфейсов и визуализации.

В 3.2 определены основные задачи **инженерии программного обеспечения** (В.В. Липаев):

- методы разработки по стандартной ЖЦ (ISO/IEC 12207-2007),
- методы обеспечения безопасности, защиты данных;
- методы верификации моделей и тестирования систем из ГОР и сбор данных об ошибках, дефектах и отказах;
- оценивание надежности, качества ГОР и систем из них для формирования сертификата созданного продукта;

**Инфраструктура и программа информатизации России (1992)** включала:

- систему коммуникации вычислительных средств и сетей для взаимодействия информационных объектов и технологий;
- программные средства функционирования комплексов аппаратуры, информационных и программных средств;
- систему образования кадров для эффективного развития информатизации в стране.

Основным инструментом информатизации России определена **рыночная экономика**.



## 4.1. Технологизация информатизации (1992-2018)

После 1992 года стали использоваться в основном зарубежные технологии:

- UML (Unified Modeling Language),
- OWL (Ontology Web Language),
- WSDL (Web Service Definition Language),
- XML, XSL, RDF, Semantic Web и др.

**Новые ЯП:**

C++, C, C#, Basic, JAVA, Ada, Prolog, Python и др.

С 2004 года сформировалась **индустриальная технология** производства программ на **ProductLine/ProductFamily**, в Grid-системах и появились **фабрики программ AppFab**.

Их основу составляет **конвейерная сборка** готовых программных ресурсов (ГОР) из модулей, объектов, компонентов, сервисов, накопившихся в Интернет по разным прикладным и математическим областям в библиотеках и репозиториях Интернет. Разработан метод ОКМ для моделирования новых систем. В нем формируются графовая модель ПрО и две модели:

- модель прикладной системы *Мпс*, включающая совокупность ГОР, отвечающая функциям реализуемой системы  $Мпс = (F1, F2, \dots, Fn)$ ,  $n=1, \dots, N$ :
- модель варибельности MF (**Model Feature**), включающая базовые внешние функции, которые могут уточняться и развиваться.

При этом ГОР могут быть готовы и находиться в среде Семантик Веб Интернета ([www.semanticweb.org](http://www.semanticweb.org)). Для их интеграции в Веб-систему используется метод композитной сборки сервисных элементов по моделям SOA (Service-oriented Architecture), SCA (Service-Component Architecture).

## 4.2. Методология и технология программирования (Проект 1992)

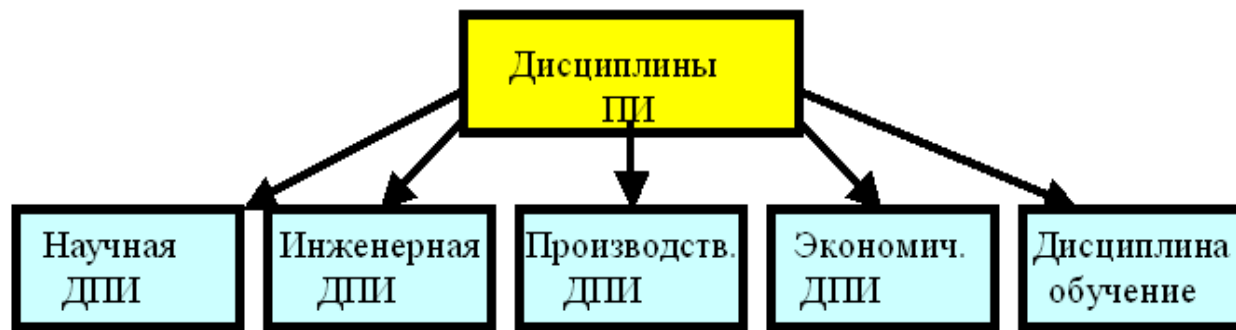
**Программная инженерия.** Разработка больших программных проектов требует повышенной надежности программ и качества.

Применяемые методологии и технологии CASE обеспечат **эффективную** поддержку всех этапов ЖЦ больших ПС, начиная от эскизного проектирования, математического моделирования до реализации и сопровождения программ.

Результаты такого подхода можно видеть по качеству программных продуктов, поступающих с Запада в нашу страну.

**Параллельные вычисления.** Создание оптимизированных компиляторов с языков высокого уровня, новых алгоритмических языков, распределенных операционных систем для векторных супер-ЭВМ и ЭВМ с распределенной параллельной архитектурой.

**Машинная графика.** Создание гипер- и мультисреды, искусственная реальность и графических суперкомпьютеров и рабочих станций на основе параллельных архитектур. Визуализация – это инструмент исследований в физике, биологии, химии, медицине, архитектуре, искусстве.



**Научная дисциплина SE** включает классические науки (теория алгоритмов, теория множеств, теория доказательств, математическая логика, дискретная математика); теория программирования, теория абстрактных данных, теория управления и др. Эта дисциплина задает базовые понятия и объекты, а также формализмы описания элементов систем и описания данных [13, 14] и др.

**Инженерная дисциплина SE** включает способы применения технологических правил и процедур, процессов ЖЦ, методик измерения и оценки качества разработки ПП. Данная дисциплина задает набор инженерных приемов, способов, средств и стандартов, ориентированных на изготовление целевых ПП. Базовые понятия инженерии SE включают: ядро знаний SWEBOK; базовый процесс SE; инфраструктуру среды обеспечения базового процесса; общесистемные и инструментальные средства поддержки ПП.

**Дисциплина управления SE** основана на теории менеджмента проектов и стандарт IEEE Std.1490 PMBOK (Project Management Body of Knowledge); методе CRM (Critical Path Method) для графического представления работ, операций и времени их выполнения; метод сетевого планирования PERT (Program Evaluation and Review Technique) и др. В стандарте PMBOK определены процессы ЖЦ проекта и главные области знаний, и процессы планирования, мониторинга, управления и завершения.

**Экономическая дисциплина SE.** Эта дисциплина обеспечивает расчет разных сторон деятельности разработчиков при выполнении проекта и определения затрат, стоимости, сроков и экономических показателей согласно требований к ПП. При этом используются методы: прогнозирования размера ПП (FPA – Function Points Analyses, Feature Points, Mark-II Function Points, 3D Function Points и др.); оценки трудозатрат на разработку ПП с помощью семейства моделей COCOMO [1] и систем (Angel, Slim, Seer-SEM и др.), а также качества ПП.

**Производственная дисциплина SE** определяет выпуск ПП и получение прибыли. В области SE продукты массового производства, создаваемые известными фирмами Microsoft, IBM, Intel и др., а также результаты аутсорсинга (обновление устаревшего, унаследованного ПО) приносят владельцам ПО большие *прибыли*. Производство ПП базируется на технологических процессах изготовления определенных видов продуктов с применением теории проектирования и использования инструментальных сред

1. Программная инженерия—научная и инженерная дисциплина.—/Лаврищева Е.М.—Кибернетика и системный анализ.—2008.—№3.—С. 19—28.
2. [Classification of software engineering disciplines](#). [E. M. Lavrischeva](#) 2008, [Volume 44, Number 6](#), Pages 791-796, Software—Hardware Systems
3. Классификация дисциплин программной инженерии.—/Лаврищева К.М.—Кибернетика и системный анализ.—2008.—№6.—С.3—9.
4. [Software engineering as a scientific and engineering discipline](#) [E. M. Lavrishcheva](#), 2008, [Volume 44, Number 3](#), Pages 324-332

## 4.2. Технологии ИСП РАН после 2000

В рамках ГосНИИАС, НИИСИРАН, ИСП РАН, НПО РусБиИТех и зарубежных – Samsung, Huawei, Intel, Rogue Wave, Linux Foundation реализуются научные проекты в области системного программирования – анализ кода, верификация, обработка больших данных, технология параллельных вычислений, развития операционных систем и др. В ИСП РАН разработаны технологические инструменты:

- эмулятор QEMU для анализа программ,
- набор детекторов для обнаружения дефектов в программах,
- верификатор цифровой аппаратуры,
- средства защиты информации и драйверов ядра ОС Linux,
- онтология ОС Tizen<sup>1</sup> с участием университета Мексики и Уругвая и онтология ЖЦ ПС<sup>2</sup>,
- анализатор больших объемов данных для Cloud Computing, MapReduce, Hadoop,
- механизмы управления транспортом с помощью Smart IoT и метод восстановления разрушенных дорог (Сан-Петербург, ПУ) и др.
- ОКМ моделирование ОС, ИС и ПС<sup>4</sup>,
- теория и метод создания фабрик программ (спрос в Китае, Индии, Тайвани и др.<sup>5-6</sup>).

---

1. Сборник технологий. ИСП РАН Москва, 2017.-34с. (info-isp@ispras.ru).

2. Lavrischeva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle, «Science and Information Conference-2015", July 28-30, London, UK, [www.conference.thesai.org](http://www.conference.thesai.org). - p. 965-972.

3. Лаврищева Е.М. Реализация метода онтологического моделирования домена ЖЦ стандарта ISO/IEC 12207 онтологии стандарта ЖЦ 12207. Док 20180711135024, 2018.

4. Лаврищева Е.М. Теория объектно-компонентного моделирования программных систем.- препринт ИСП РАН, 2016, 48 с. ISBN 978-5-9-474-025-9, <http://www.ispras.ru/lavrishcheva/>.

5. Lavrischeva K.M. Theoty and Practice of Software Factories. - Cybernetic and Systems Analyses, 2011. Vol. 47.-N6. - p. 961-972.

6. Lavrischeva K. A. Aronov, A. Dzubenko. Programs Factory – A conception of Knowledge Representation of Scientific Artifacts From Standpoint of Software Engineering // Comp. and Inf. Sci., Canadian Center of Sci. and Edu. – 2013. – P. 21–28.

### 4.3. Теория и технология ОКМ<sup>1-3</sup>

ОКМ - метод проектирования ОМ системы из объектов, функции которых реализуются программными компонентами и интерфейсами и записывается в виде графа. При моделировании формируются модели ОМ, Мпс, MF и программы с помощью:

- ГОР и КПИ (reuses, assets, servises, artifacts);
- математических операций (объединения  $\cup$ , пересечения  $\cap$ , вычитания  $\ominus$ , симметричного вычитания  $\setminus$  и др.);
- объектной и компонентной алгебры;
- операции конфигурирования КПИ и преобразования ТД к разным форматам платформ сред (VS.Net, IBM, Corba, Eclipse и др.) с помощью примитивных функций GDT ISO/IEC 11404–2007 к FDT ЯП (и обратно);
- тестирования ГОР и систем из них на множестве наборов данных, сбора данных об ошибках, отказах, дефектах и т.д.
- оценивание качественных показателей (включая надежность) готовой системы,

---

1. Лаврищева Е.М. Теория объектно-компонентного моделирования программных систем.- препринт ИСП РАН, 2016, 48 с. ISBN 978-5-9-474-025-9, <http://www.ispras.ru/lavrishcheva/>.

## Парадигма объектно-ориентированного программирования

ООП — парадигма основана на представлении предметной области в виде системы взаимосвязанных абстрактных объектов и их реализаций. **Принципы ООП:**

- классы (классификация) объектов, суперклассы;
- *наследование* — механизм установления отношений «потомок — предок» (порождать один класс от другого с сохранением всех свойств и методов класса-предка);
- *инкапсуляция* — свойство сокрытия реализации класса;
- *абстракция* — описание взаимодействия в терминах сообщений/событий предметной области;
- *полиморфизм* — возможность подмены в сценарии взаимодействия объектов одного класса другим объектом со сходной структурой.

---

Буч Г. Объектно-ориентированный анализ. М. : Бином, 1998;

Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя : пер. с англ. М. : ДМК, 2000.

ОКМ позволяет моделировать Предметную область, домен в виде множества объектов со свойствами и характеристиками, которые необходимы и достаточны для их определения и идентификации, а также описания их поведения в рамках системы понятий и абстракций. Объекты в ОКМ определяются на **четырёх уровнях** моделирования с привлечением логико-математических формализмов и уточнения функций, свойств и характеристик объектов.

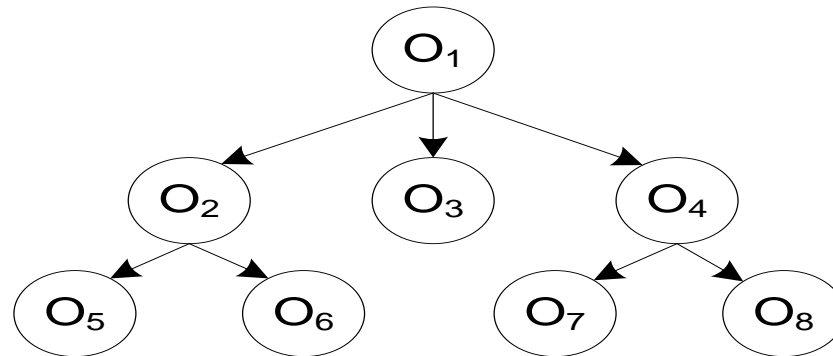
К формальным уровням представления ОМ из объектов относятся:

1. **Обобщающий уровень**
2. **Структурный уровень**
3. **Характеристический уровень**
4. **Поведенческий уровень**

На **обобщающем уровне** формируется множество **объектов**

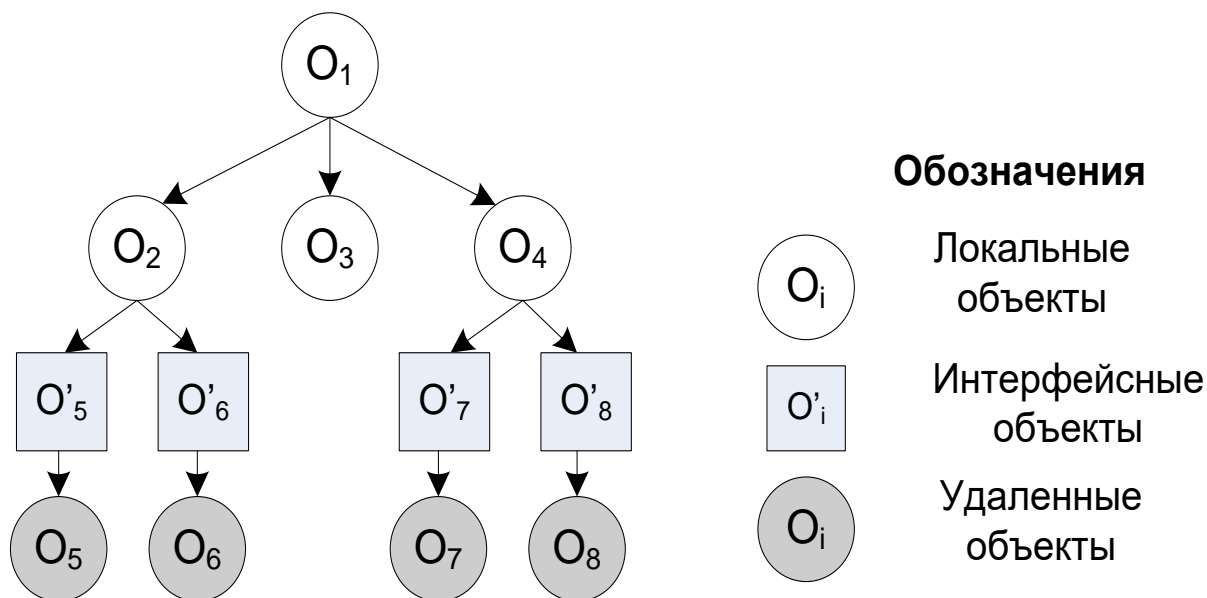
$O = (O_0, O_1, \dots, O_n)$ , на **структурном** – система  $\Sigma = (O', \Omega)$ , где

$\Omega$  - множество операций ( $\cup, \cap, /, \diamond, \oplus, -$  и др.) и граф  $G$





На **поведенческом уровне** задается последовательность состояний объектов в зависимости от времени и окончательный граф  $G$ .



В данном графе  $G$  заданы:

- $O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8$  - функциональные объекты/КПИ;
- $O'_5, O'_6, O'_7, O'_8$  - интерфейсные объекты, которые

размещаются в репозитории интерфейсов, а дуги соответствуют связям между видами объектов.

Элементы графа  $O_1 - O_8$  описываются в ЯП (Fortran, Basic, C++ и др.), а интерфейсные объекты  $O'_5 - O'_8$  - в **языке IDL** (Interface Definition Language).

По графу  $G$  можно построить программы  $P_0 - P_5$  с использованием мат. операций  $\cup$  объединения и оператора **link**:

$$P_0 = (P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5).$$

$$P_1 = O_2 \cup O_5, \text{ link } P_1 = \text{In } O'_5 (O_2 \cup O_5):$$

$$P_2 = O_2 \cup O_6, \text{ link } P_2 = \text{In } O'_6 (O_2 \cup O_8):$$

$$P_3 :$$

$$P_4 = O_4 \cup O_7, \text{ link } P_4 = \text{In } O'_7 (O_4 \cup O_7):$$

$$P_5 = O_4 \cup O_8, \text{ link } P_4 = \text{In } O'_8 (O_4 \cup O_8):$$

Модель вариабельности  $M_{var}$  системы:

$$M_{var} = (SV; AV),$$

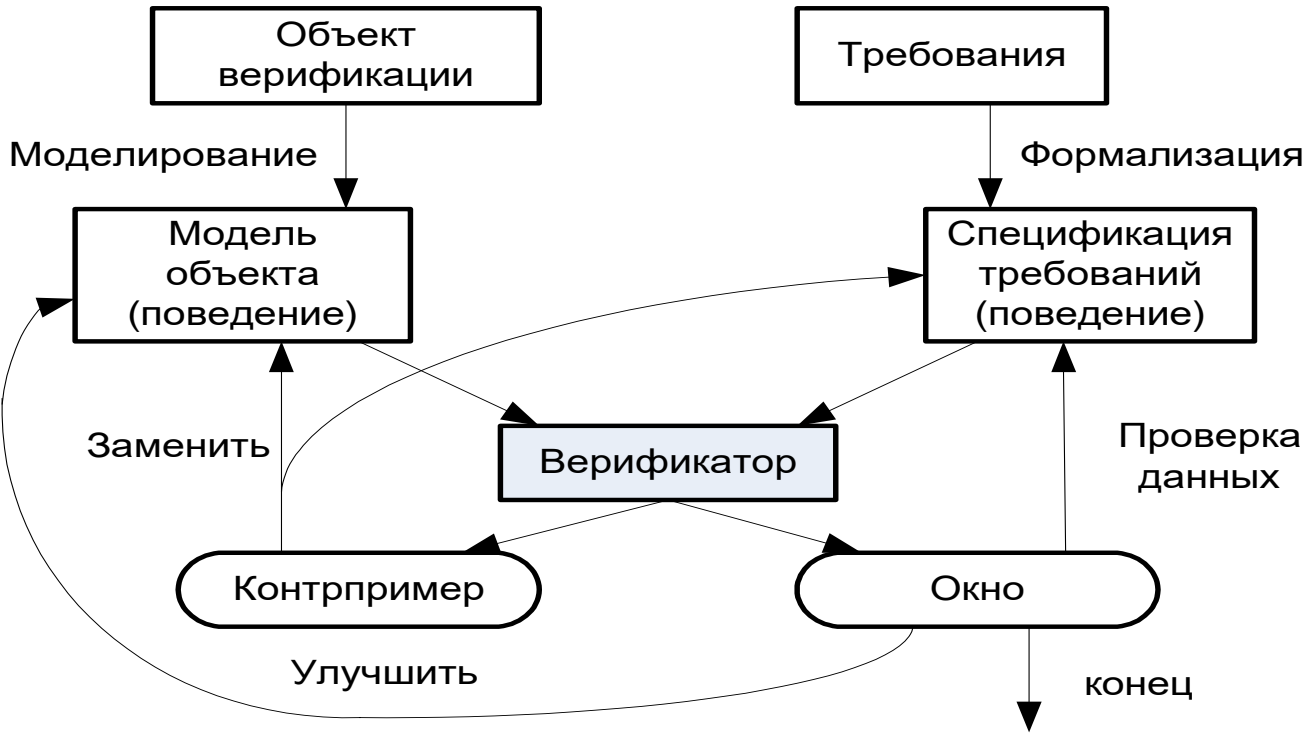
где  $SV$  – подмодель вариабельной архитектуры ПС;

$AV$  – подмодель вариабельности артефактов – ресурсов.

**Модель  $M_{var}$**  задает:

- уровень изменяемости продуктов ПС с учетом требований;
- изменение артефактов приложений и ПС как ГОР.

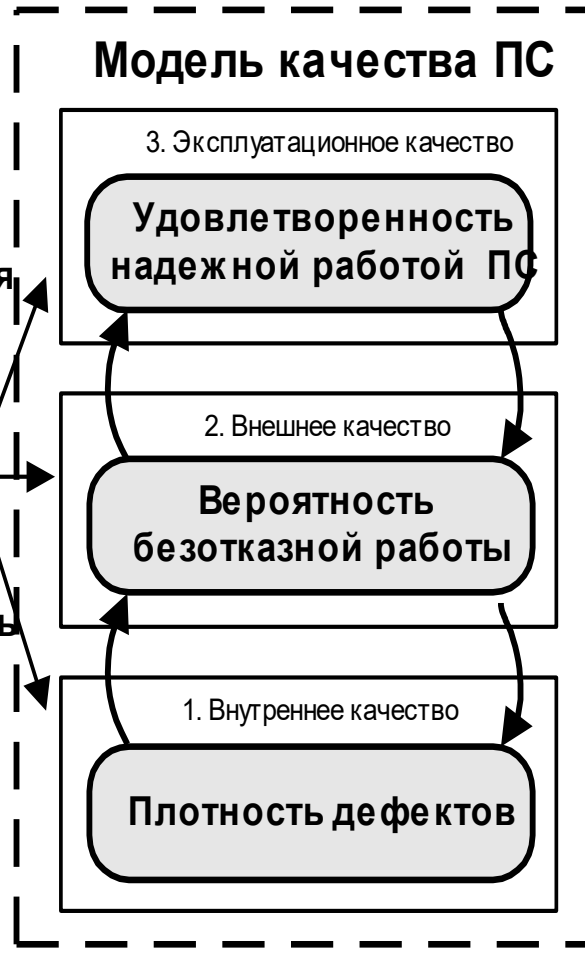
Верификатор модели MF устанавливает соответствие спецификации модели объекта, требованиям и свойствам. Если соответствие удовлетворяется, то верификатор сообщает правильность модели, в противном случае дается пояснение о возникновении несоответствия.



## Качество ПС

- Функциональность
- Удобство применения
- **Надежность**
  - **Завершенность**
  - Отказоустойчивость
  - Восстанавливаемость
- Эффективность
- Сопровождаемость
- Переносность

Обеспечиваемые  
характеристики  
качества по ISO/IEC  
9126-1:2001



## Эксплуатационное качество ПС

- Удовлетворенность
- Безопасность функционирования
  - Результативность
  - Продуктивность
- Потребительские характеристики качества по ISO/IEC 9126-4:2001

**В плане развития технологии моделирования, разработки и анализа программ на передний план выходят проблемы обеспечения безопасности, защиты данных, обеспечения надежности и отказоустойчивости программных и аппаратных комплексов критической инфраструктуры<sup>1-2</sup>.**

**Одним из важных аспектов надежности и информационной безопасности программных систем является моделирование и верификация так называемых "вариабельных" и качественных программных систем (и семейств) программных продуктов.**

**1. Доклад на конференции OS DAY -2018 Надежность - [Лаврищева Е.М., Пакулин Н.В.](#)**

**2. [Лаврищева Е.М., Пакулин Н.В., Рыжов А.Г., Зеленов С.В.](#) Анализ методов оценки надежности оборудования и систем. Практика применения методов. Труды ИСП РАН, том5 DOI: 10.15514/ISPRAS-2018-30(3), 2018.**

**Современной проблеме моделирования систем был посвящен проект РФФИ 16-01-00352 (2016-2018).**

**"Теория и методы разработки вариабельных программных систем». В нем отработаны все аспекты моделирования изменяемых систем, верификации, тестирования и оценки надежности с участием сотрудников ИСП РАН и студентов МФТИ.**

**Основные результаты этого проекта опубликованы ниже [1-13].**

## Публикации по проекту РФФИ (2016-2018)

1. Е.М. Лаврищева, Л. Е. Карпов, А. Н. Томилин. Семантические ресурсы для разработки онтологии научной и инженерной предметных областей, Доклад на конференции "Научный сервис в сети Интернет" 19-24 сентября 2016, Абрау.
2. Лаврищева Е.М. Теоретические основы моделирования программных систем из объектов и компонентов. Доклад на Международной научно-практической конференции «Теория активных систем (ТАС-2016), 16-17 ноября 2016.- ИПУ РАН им. В.А.Трапезникова.
3. Кулямин В.В., Лаврищева Е.М., Мутилин В.С., Петренко А.К. Верификация и анализ переменных операционных систем. *Труды ИСП РАН*, 2016, том X вып. 28, вып. 6 с. 48-59.
4. Ekaterina M.Lavrishcheva. Assembling Paradigms of Programming in Software Engineering.- 2016, 9, 2016.- p.296-317, <http://www.scrip.org/journal/jsea>, <http://dx.do.org/10.4236/jsea.96021>
5. Лаврищева Е.М. Петренко А.К. Моделирование систем и их семейств. *Труды ИСП РАН*, 2016, том 28, вып. 6, с. 49-65.
6. Lavrishcheva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle, Conference "Science and Information Conference-2015", July 28-30, London, UK, [www.conference.thesai.org](http://www.conference.thesai.org).- p.965-972.
7. Лаврищева Е.М. Рыжов А.Г. Применение теория общих типов данных стандарта ISO/IEC 12207 GDT применительно к Big Data.- The conference "Actual problems in science and ways their development", 27 december 2016, <http://euroasia-science.ru/> p.99-110.
8. Лаврищева Е.М. Теория объектно-компонентного моделирования программных систем. - препринт ИСП РАН, 2016, 48 с. ISBN 978-5-9-474-0259, <http://www.ispras.ru/lavrishcheva/>.
9. Е.М. Лаврищева, В.С. Мутилин, А.Г. Рыжов. Аспекты моделирования переменных программных и операционных систем.- Сб. XIX Всероссийский научной конференции «Научный сервис в сети Интернет», Новороссийск, 18-23 сентября 2017.-327-341.
10. Lavrishcheva E.M., Mutilin V.S., Ryzhov A.G. Designing variability models for software, operating systems and their families. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 5, 2017, pp. 93-110, DOI: 10.15514/ISPRAS-2017(5).
11. Е. М. Лаврищева, Л. Е. Карпов, А. Н. Томилин. Подходы к представлению научных знаний в Интернет науке. Сб. XIX Всероссийский научной конференции «Научный сервис в сети Интернет», Новороссийск, 18-23 сентября 2017.-с.310-326.
12. Лаврищева Е.М. Развитие теории программ и систем в СССР. История и современные теории.-Сборник SORUCOM-17, 2017. Развитие ВТ в России и в странах бывшего СССР. История и перспективы.- 3-5 октября 2017.-с.162-176.
- 13 E.M. Lavrishcheva. Scientific Basis of System Programming.- Journal of Software Engineering and Applications (JSEA), Vol. 11 No. 8 of August issue, 2018.-N 11.-p.408-434, ISSN online 1945-3124, ISSN Print 1945-3116.

## 4.4. Современные парадигмы программирования

К современным *парадигмам программирования* относятся<sup>1-4</sup>:

- объектная,
- компонентная,
- сервисная,
- аспектная,
- функциональная,
- агентная и др.,
- **Agile, EX, SCRUM** и др.

Сущность каждой парадигмы – формальное описание программных элементов для повторно используемых компонентов (КПИ) или Reuses типа – модуль, объект, компонент, сервис и др. Накопление их в библиотеки открытого типа в Интернет для их предоставления Всем желающим на коммерческой рыночной основе.

- 
1. Городняя Л.В. Парадигмы программирования: Анализ и сравнение.- СОРАН, 2017.-239 с.
  2. Лаврищева Е.М. Программная инженерия. Теория программирования. –2016.- МФТИ. - Методичка -51 с.
  3. Лаврищева Е.М. Программная инженерия. Технология программирования.-2016 – МФТИ.- Методичка - 52 с.
  4. Лаврищева Е.М. Программная инженерия. Технология программирования.-2016 – МФТИ. - Методичка.52 с.

**Парадигмы событийно-управляемого** программирования основываются на концепции организации *«внутреннего» потока управления* путем выполнения инструкций программы, не связанных с наступлением событий вне ее.

**Событийное программирование** (*event-oriented, event-based* или *event-driven programming*), которое заключается в подмене системных функций-триггеров «по умолчанию» нужными функциями, соответствующими каждому возможному событию. Программа состоит из ряда маленьких прикладных программ, называемых *обработчиками событий* и *диспетчера* обработки очередей событий и имитирующих среду, управляемую прерываниями. Тестирование операционных (сценарных) профилей программ и прогноз размера ПП по FPA.



## Согласованное программирование и параллельные вычисления

Для повышения производительности компьютеров проводится распараллеливание потока команд или потока данных.

Распараллеливание данных подразумевает применение одной операции сразу к нескольким элементам массива данных, а параллелизм задач — разбиение вычислительного процесса на несколько подзадач (процессов), каждый из которых выполняется на своем процессоре.

**Согласованное программированию** (*concurrent programming*) – это взаимодействие некоторых одновременно работающих вычислителей (нескольких процессоров или многоядерных процессоров). Подходы - *обмен сообщениями*, использование *разделяемой памяти* и *стандарт POSIX*.

**Параллельное программированию** (*parallel programming*) или параллельные вычисления — *parallel computing*) включает параллельные потоки, порождение и обработка асинхронных событий.

## Программирование по прототипу

В этом стиле программирования понятие класса отсутствует, а повторное использование (наследование) производится путем **клонирования** существующего экземпляра объекта, как прототипа.

**Стиль прототипного программирования** – обеспечение поведения небольшого количества «образцов», как «базовых» объектов при создании других объектов.

Создание нового объекта выполняется путем *клонирования* существующего объекта либо создания объекта «с нуля» и дальнейшего получения его копии — клона, который наследует все характеристики своего прототипа и становится самостоятельным, может изменяться без побочных эффектов.

Каноническим примером прототипного языка является язык [Self](#).

Прототипный стиль программирования положен в основу таких языков программирования, как [JavaScript](#), [Squeak](#), [NewtonScript](#), [MOO](#), [REBOL](#), [Kevo](#) и др.

# Agile-технологии

**Agile Software Development.** Включает методологии eXtreme Programming - XP (экстремальное программирование), SCRUM, Dynamic Systems Development Method (DSDM), Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming.

Методологии Agile ориентированы на тесное взаимодействие команды разработчиков с пользователями на основе итеративной модели ЖЦ с приращениями и быстрой реакции на изменяющиеся требования и ProductLine подход к разработке ПО. В **XP** любой член группы может изменить не только свой код, но и код другого программиста и каждый модуль снабжается автономным тестом (*unit test*), обеспечивая возможность регрессионного тестирования модулей. Тесты пишут сами программисты и они имеют право написать теста для любого модуля.

**SCRUM** — гибкая методика управления проектом с процессом разработки - анализ требований, проектирование, программирование, тестирование (<http://agile.csc.ncsu.edu>).

**DSDM** - метод разработки информационных систем консорциума DSDM.

**Agile** — манифест разработки ПО. [www.agilemanifesto.org/iso](http://www.agilemanifesto.org/iso).

## **Заключение**

**Рассмотрены пути становления информатики и технологий программирования для автоматизации создания программного обеспечения и прикладных систем, начиная с появления первых ЭВМ и дисциплин обработки информации первоначально и в современный период информатизации.**

**Приводятся компьютерные и информационные системы, а также технологии для автоматизации и интеллектуализации разработки программных и прикладных систем различных видов.**

**Описана программа информатизации России 1992 года. Отмечаются новые пути развития интеллектуальных систем в направлении nano и роботизации приборов и машин для разных областей, особенно в медицине.**

**Отмечены инновационные разработки в области информатизации, интеллектуализации, онтологизации и обеспечения надежности и безопасности компьютерных систем.**

# **Благодарю за внимание**

**LONDON PRESS 4 November 2016**

**Dear Lavrischeva E.M.**

**Membership ID#513442 considering your paper “Object-Component Development of Application and Systems Theory and Practica**

**I am writing with reference to your research paper titled, “Object-Component Development of Application and Systems Theory and Practice” and have acknowledged it worthy of commendation. I found your research work to be exceedingly impressive and impactful. This research can indeed prove to be significant for fellow researchers and scientists working in the same domain.**

**Taking note of your research interests which matches with our journal domain, I would like to welcome you to associate with us. To follow this, our Editorial Board has agreed to recognise you under "Quarterly Franklin Membership" (Membership ID#513442|UK) of London Journals Press. Also, we encourage you to have your upcoming research articles/papers published in, London Journal of Research in Computer Science and Technology (LJCST). As a part of this honorary membership, APC/Membership fee is fully waived off for you.**

**London Journals Press (U.K.) is an internationally acclaimed publication organisation and an accreditation authority in research standards. We follow COPE and Research Councils UK standards and are in association with researchers in all the leading discipline like Computer Science, Engineering, Management, Medical Research, Science, Social Science, etc.**

**We follow a comprehensive and extensive peer review process which is a time-consuming procedure. Following the convention, your paper will be reviewed on a priority basis since you belong to the Council of Franklin Members. As our forthcoming issue of LJCST is due on 25-Nov-16, we request you to submit your research paper/article at the earliest. To relieve authors from any guidelines/formatting/typesetting tasks we accept the research papers in any format. Our dedicated professionals format the paper following the protocols that meet our international standards.**

**Post submission, the publishing process can be live tracked on the website. For more support, you can also use the Live Chat facility available at the bottom right corner on our website. You can read more about London Journals Press, our core ethics and values on the website, [journalspress.com](http://journalspress.com).**

**I look forward hearing from you soon and a successful academic relationship in the future.**

**Yours Faithfully**

**Dr. Waley A.**

**London Journals Press**

## **Disciplines of SE**

In connection with the 40 year anniversary SE (2008) the author proposed a classification of scientific disciplines in SE the articles [13, 14, and 10]. Proposed discipline used in the program Curricula-13. Let us consider briefly their characteristics.

***Scientific discipline*** SE includes classic Sciences (theory of algorithms, set theory, proof theory, mathematical logic, discrete mathematics); theory of programming the theory of abstract data, management science, etc. This discipline defines the basic concepts and the objects and the formalism of the description of the system components and data description [13, 14] etc.

***Engineering discipline*** SE includes methods of using technology rules and procedures, processes, life cycle, methods of measuring and assessing the quality of development PP. This discipline defines the set of engineering methods, techniques, tools and standards focused on the production of the target PP. Basic concepts of engineering SE include: core knowledge SWEBOK; the basic process SE; infrastructure environment.

***Discipline management*** SE is based on the theory of management of projects and the IEEE Std.1490 PMBOK (Project Management Body of Knowledge); method CRM (Critical Path Method) for the graphic representation of works, operations and their execution time; method of network planning PERT (Program Evaluation and Review Technique), etc. In the PMBOK defined processes lifecycle of the project and the main areas of knowledge and processes of planning, monitoring, management and completion.

***The economic discipline of SE.*** This discipline provides for the calculation of the different parties activities of developers in the implementation of the project and identify the costs, cost, time and economic indicators according to the requirements of PP. Used methods: predicting the size of PP (FPA – Function Points Analyses, Feature Points, Mark II Function Points, 3D Function Points, etc.); the evaluation effort for the development of PP by using models COCOMO [1] and systems (Angel, Slim, Seer-SEM, etc.), as well as the quality of PP.

***Production discipline*** SE determines the production of PP and makes a profit. In the area of SE mass produced products created by the famous firms Microsoft, IBM, Intel, and others, as well as the results of outsourcing (upgrading a legacy inherited BY) bring ON large profits. The production of PP is based on the technological processes of manufacture of certain product types using the theory of the design and usage of tool environments.