***1.Информационные системы и их классификации. Основные понятия.***

**Информационная система** (**ИС**) — система, предназначенная для хранения, поиска и обработки информации[[1]](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-ISO.2FIEC_2382-1.IPS-1) и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (ISO/IEC 2382:2015)[[2]](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-ISO.2FIEC_2382-1.IS-2).

Информационная система предназначена для своевременного обеспечения надлежащих людей надлежащей [информацией](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F)[[3]](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-William_S._Davis.2C_David_C._Yen.E2.80.941998.E2.80.94.E2.80.94Information_system_is_a_set_of_hardware.2C_software.2C_data.2C_human.2C_and_procedural_components_intended_to_provide_the_right_data_and_information_to_the_right_person_at_the_right_time.-3), то есть для удовлетворения конкретных информационных потребностей в рамках определенной [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C), при этом результатом функционирования информационных систем является *информационная продукция* — документы, [информационные массивы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2), [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и информационные услуги[[4]](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-gost-7.0-4).

**Классификация по**[**архитектуре**](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B)

По степени распределённости отличают:

* *настольные* (*desktop*), или *локальные* ИС, в которых все компоненты ([БД](https://ru.wikipedia.org/wiki/%D0%91%D0%94), [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94), [клиентские приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) находятся на одном компьютере;
* *распределённые* (*distributed*) ИС, в которых компоненты распределены по нескольким компьютерам.

Распределённые ИС, в свою очередь, разделяют на:

* *файл-серверные* ИС (ИС с архитектурой «[файл-сервер](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)»);
* *клиент-серверные* ИС (ИС с архитектурой «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)»).

В файл-серверных ИС база данных находится на [файловом сервере](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), а СУБД и клиентские приложения находятся на [рабочих станциях](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F).

В клиент-серверных ИС база данных и СУБД находятся на сервере, а на рабочих станциях находятся только клиентские приложения.

В свою очередь, клиент-серверные ИС разделяют на *двухзвенные* и *многозвенные*.

В двухзвенных ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *two-tier*) ИС всего два типа «звеньев»: [сервер базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), на котором находятся БД и СУБД ([back-end](https://en.wikipedia.org/wiki/Back-end_database" \o "en:Back-end database)), и рабочие станции, на которых находятся клиентские приложения ([front-end](https://en.wikipedia.org/wiki/Front-end_%D0%B8_back-end" \o "en:Front-end и back-end)). Клиентские приложения обращаются к СУБД напрямую.

В многозвенных ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *multi-tier*) ИС добавляются промежуточные «звенья»: [серверы приложений](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) (*application servers*). Пользовательские клиентские приложения не обращаются к СУБД напрямую, они взаимодействуют с промежуточными звеньями. Типичный пример применения [трёхзвенной архитектуры](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0) — современные [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5" \o "Веб-приложение), использующие базы данных. В таких приложениях помимо звена СУБД и клиентского звена, выполняющегося в веб-[браузере](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80" \o "Браузер), имеется как минимум одно промежуточное звено — [веб-сервер](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80" \o "Веб-сервер) с соответствующим серверным программным обеспечением.

**Классификация по степени автоматизации**

По степени [автоматизации](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) ИС делятся на:

* *автоматизированные*: информационные системы, в которых автоматизация может быть неполной (то есть требуется постоянное вмешательство персонала);
* *автоматические*: информационные системы, в которых автоматизация является полной, то есть вмешательство персонала не требуется или требуется только эпизодически.

«Ручные ИС» («без компьютера») существовать не могут, поскольку существующие определения предписывают *обязательное* наличие в составе ИС аппаратно-программных средств. Вследствие этого понятия «автоматизированная информационная система», «компьютерная информационная система» и просто «информационная система» являются синонимами.[[8]](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-.D0.9A.D0.BE.D0.B3.D0.B0.D0.BB.D0.BE.D0.B2.D1.81.D0.BA.D0.B8.D0.B9_.D0.9C._.D0.A0..E2.80.942003.E2.80.94.E2.80.94-8)

**Классификация по характеру обработки данных**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&veaction=edit&vesection=5) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&section=5)]

По характеру обработки данных ИС делятся на:

* *информационно-справочные*, или *информационно-поисковые ИС*, в которых нет сложных алгоритмов обработки данных, а целью системы является поиск и выдача информации в удобном виде;
* *ИС обработки данных*, или *решающие ИС*, в которых данные подвергаются обработке по сложным алгоритмам. К таким системам в первую очередь относят [автоматизированные системы управления](https://ru.wikipedia.org/wiki/%D0%90%D0%A1%D0%A3) и [системы поддержки принятия решений](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BF%D0%BE%D0%B4%D0%B4%D0%B5%D1%80%D0%B6%D0%BA%D0%B8_%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9).

**Классификация по сфере применения**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&veaction=edit&vesection=6) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&section=6)]

Поскольку ИС создаются для удовлетворения информационных потребностей в рамках конкретной предметной области, то каждой предметной области (сфере применения) соответствует свой тип ИС. Перечислять все эти типы не имеет смысла, так как количество предметных областей велико, но можно указать в качестве примера следующие типы ИС:

* [Экономическая информационная система](https://ru.wikipedia.org/w/index.php?title=%D0%AD%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&redlink=1) — информационная система, предназначенная для выполнения функций управления на предприятии.
* [Медицинская информационная система](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) — информационная система, предназначенная для использования в лечебном или лечебно-профилактическом учреждении.
* [Географическая информационная система](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) — информационная система, обеспечивающая сбор, хранение, обработку, доступ, отображение и распространение пространственно-координированных данных (пространственных данных).

**Классификация по охвату задач (масштабности)**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&veaction=edit&vesection=7) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&section=7)]

* *Персональная* ИС предназначена для решения некоторого круга задач одного человека.
* *Групповая* ИС ориентирована на коллективное использование информации членами рабочей группы или подразделения.
* [*Корпоративная* ИС](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BF%D0%BE%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) автоматизирует все бизнес-процессы целого предприятия (организации) или их значительную часть[[9]](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-is.glossary.2009-9):73, достигая их полной информационной согласованности, безызбыточности и прозрачности. Такие системы иногда называют *информационными системами предприятия* и *системами комплексной автоматизации предприятия*.

***2. Архитектура информационных систем.***

**Архитектура системы** — [принципиальная](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF) организация [системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), воплощенная в её [элементах](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82_(%D1%84%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D0%B8%D1%8F)), их взаимоотношениях друг с другом и со средой, а также принципы, направляющие её [проектирование](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и эволюцию[[1]](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#cite_note-s1-1):3.

Понятие архитектуры в значительной мере [субъективно](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D1%84%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D0%B8%D1%8F)) и имеет множество противоречивых толкований; в лучшем случае оно отображает общую точку зрения команды разработчиков на результаты проектирования системы.[[2]](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#cite_note-.D0.A4.D0.B0.D1.83.D0.BB.D0.B5.D1.80_.D0.9C..E2.80.942006.E2.80.94.E2.80.94-2):27 Существует большое количество определений архитектуры. Коллекция определений, относящихся, в основном, к [архитектуре программного обеспечения](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), собрана на сайте [Института программной инженерии](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B9_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B8%D0%B8)[**?!**](https://ru.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD:%D0%9D%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%BE_5#.D0.95.D1.81.D0.BB.D0.B8_.D1.81.D1.82.D1.80.D0.B0.D0.BD.D0.B8.D1.86.D0.B0_.D1.81.D1.83.D1.89.D0.B5.D1.81.D1.82.D0.B2.D1.83.D0.B5.D1.82) [Университета Карнеги — Меллона](https://ru.wikipedia.org/wiki/%D0%A3%D0%BD%D0%B8%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%82%D0%B5%D1%82_%D0%9A%D0%B0%D1%80%D0%BD%D0%B5%D0%B3%D0%B8_%E2%80%94_%D0%9C%D0%B5%D0%BB%D0%BB%D0%BE%D0%BD)[[3]](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#cite_note-3).

В настоящее время существует сильная тенденция рассматривать архитектурное и неархитектурное проектирование как различные виды деятельности; делаются попытки определить их как отдельные практики, однако эти виды проектирования в значительной мере «переплетены». Архитектурные решения в сравнении с обычными проектными решениями рассматриваются как более абстрактные, концептуальные и глобальные; они нацелены на успех всей миссии и на наиболее высокоуровневые структуры системы[[4]](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#cite_note-s5-4):272.

**Логическая архитектура**

Логическая архитектура поддерживает функционирование системы на протяжении всего её жизненного цикла на логическом уровне. Она состоит из набора связанных технических концепций и принципов. Логическая архитектура представляется с помощью методов, соответствующих тематическим группам описаний, и как минимум, включает в себя функциональную архитектуру, поведенческую архитектуру и временную архитектуру.

*Функциональная архитектура*. Функциональная архитектура представляет собой набор функций и их подфункций, определяющих преобразования, осуществляемые системой при выполнении своего назначения.

*Поведенческая архитектура*. Поведенческая архитектура — соглашение о функциях и их подфункциях, а также интерфейсах (входы и выходы), которые определяют последовательность выполнения, условия для управления или потока данных, уровень производительности, необходимый для удовлетворения системных требований. Поведенческая архитектура может быть описана как совокупность взаимосвязанных сценариев, функций и/или эксплуатационных режимов.

*Временная архитектура*. Временная архитектура является классификацией функций системы, которая получена в соответствии с уровнем частоты её исполнения. Временная архитектура включает в себя определение синхронных и асинхронных аспектов функций. Мониторинг решений, который происходит внутри системы, следует той же временной классификации [[4]](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#cite_note-s5-4):287.

**Физическая архитектура**

Цель проектирования физической архитектуры заключается в создании физического, конкретного решения, которое согласовано с логической архитектурой и удовлетворяет установленным системным требованиям.

После того, как логическая архитектура определена, должны быть идентифицированы конкретные физические элементы, которые поддерживают функциональные, поведенческие, и временные свойства, а также ожидаемые свойства системы, полученные из нефункциональных требований к системе.

Физическая архитектура является систематизацией физических элементов (элементов системы и физических интерфейсов), которые реализуют спроектированные решения для продукта, услуги или предприятия. Она предназначена для удовлетворения требований к системе и элементам логической архитектуры и реализуется через технологические элементы системы. Системные требования распределяются как на логическую, так и физическую архитектуру. Глобальная архитектура системы оценивается с помощью системного анализа и, после выполнения всех требований, становится основой для реализации системы[[4]](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#cite_note-s5-4):296.

***3. Классы автоматизированных информационных систем.***

Деление [АС](https://ru.wikipedia.org/w/index.php?title=%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&redlink=1) на соответствующие классы по условиям их функционирования с точки зрения [защиты информации](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D1%89%D0%B8%D1%82%D0%B0_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8) необходимо в целях разработки и применения обоснованных мер по достижению требуемого уровня защиты информации. В соответствии с Руководящим Документом распространяется на все действующие и проектируемые АС учреждений, организаций и предприятий, обрабатывающие конфиденциальную [информацию](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F).

Выбор методов и средств защиты определяется важностью обрабатываемой информации, различием АС по своему составу, структуре, способам обработки информации, количественному и качественному составу [пользователей](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C) и обслуживающего персонала.

Классификация[[править](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D1%85_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC&veaction=edit&vesection=1) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D1%85_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC&action=edit&section=1)]

РД предусматривает следующие этапы классификации АС:

* разработка и анализ исходных данных;
* выявление основных признаков АС, необходимых для классификации;
* сравнение выявленных признаков АС с классифицируемыми;
* присвоение АС соответствующего класса защиты информации от НСД.

Для проведения классификации конкретной АС необходимо провести анализ следующих данных:

* перечень защищаемых [информационных ресурсов](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%80%D0%B5%D1%81%D1%83%D1%80%D1%81%D1%8B&action=edit&redlink=1) АС и их уровень конфиденциальности;
* перечень лиц, имеющих доступ к штатным средствам АС, с указанием их уровня полномочий;
* матрицу доступа или полномочий [субъектов](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%B1%D1%8A%D0%B5%D0%BA%D1%82) доступа по отношению к защищаемым информационным ресурсам АС;
* режим обработки данных в АС.

Определяющими признаками, по которым производится группировка АС в различные классы, являются:

* наличие в АС информации различного уровня конфиденциальности;уровень полномочий субъектов доступа АС на доступ к конфиденциальной информации;
* режим обработки данных в АС — коллективный или индивидуальный.

***4. Документальное, фактографическое и концептографическое информационное обслуживание***.

В процессе развития автоматизированных информационно-поиско-вых систем сформировались три вида информационного обслуживания: документальное, фактографическое и концептографическое. Каждому из этих видов соответствует своя информационная система.

Сущность документального обслуживания заключается в том, что информационные потребности членов общества удовлетворяются путём предоставления им первичных документов, необходимые сведения из которых потребители извлекают сами. Таким образом, потребность в информации при документальном обслуживании удовлетворяется опосредованно, через первичный документ.

В отличие от документального обслуживания фактографическое предполагает удовлетворение информационных потребностей непосредственно, т.е. путём представления потребителям самих сведений (отдельных данных, фактов, концепций). всё то, что может быть объектом извлечения из текста, описания на определённом информационном языке, хранения и поиска в той или иной информационной системе.

при концептографическом обслуживании документы и сведения подвергается интерпретации, оценке, обобщению со стороны информационного работника. В результате такой интерпретации формулируется так называемая ситуативная информация, содержащая в себе оценку рассматриваемых сведений, тенденций и перспективы развития отдельных научных и технических направлений, рекомендаций и пр., в случае концептографического обслуживания потребителю представляются не только сведения о документе или сами сведения из документа, но и некоторая дополнительная информация, привнесённая информационным работником в процессе их интерпретации.

***5.Основные процессы жизненного цикла ИС.***

Понятие жизненного цикла (ЖЦ) является одним из ключевых понятий методологии проектирования информационных систем. ***Жизненный цикл информационной системы***– это непрерывный процесс, начинающийся с момента принятия решения о создании информационной системы и заканчивающийся в момент полного изъятия ее из эксплуатации [[4](http://edu.dvgups.ru/METDOC/GDTRAN/YAT/ITIS/PROEK_INF_SIS/METOD/UMK_DO/frame/UMK_DO/M11/Literatura.htm#lit4)].

Основным стандартом, определяющим структуру жизненного цикла, является ГОСТ Р ИСО/МЭК 12207-02 [[5](http://edu.dvgups.ru/METDOC/GDTRAN/YAT/ITIS/PROEK_INF_SIS/METOD/UMK_DO/frame/UMK_DO/M11/Literatura.htm#lit5)]. Согласно стандарту структура жизненного цикла основывается на трех группах процессов:

· [***основные процессы***](http://edu.dvgups.ru/METDOC/GDTRAN/YAT/ITIS/PROEK_INF_SIS/METOD/UMK_DO/frame/UMK_DO/M1/L2.htm#2_2) (заказ, поставка, разработка, эксплуатация, сопровождение);

· ***вспомогательные процессы*** (обеспечивают выполнение основных процессов):

o *документирование* – работы по разработке, выпуску, редактированию, распространению и сопровождению документов, в которых нуждаются все заинтересованные лица;

o *управление конфигурацией* (конфигурационное управление) включает работы: определение и установление состояния программных объектов в системе; управление изменениями и выпуском объектов; обеспечение полноты, совместимости и правильности объектов; управление хранением, обращением и поставкой объектов;

o *обеспечение качества* – работы по обеспечению соответствия создаваемой системы и реализуемых процессов жизненного цикла установленным требованиям и утвержденным планам;

o *верификация* – работы соответствующего субъекта (заказчика, поставщика или независимой стороны) по проверке соответствия создаваемых промежуточных результатов установленным требованиям по мере реализации проекта. Различают верификацию договора, процесса, требований, проекта, системы, сборки системы и документации;

o *аттестация* – работы соответствующего субъекта по проверке полного соответствия требований и конечного продукта функциональному назначению системы;

o *совместный анализ* – работы по оценке состояния или результатов какой-либо работы (системы);

o *аудит* – работы независимых (по отношению к проекту) экспертов по определению соответствия деятельности субъекта принятым требованиям, планам и условиям договора;

o *разрешение проблем* – работы по анализу и устранению проблем, обнаруженных при реализации проекта;

· ***организационные*:**

o *управление проектами* – работы по планированию и управлению процессами, включая контроль, проверку и оценку выполненных работ с формированием отчетности;

o *создание инфраструктуры проекта* – работы по установлению и обеспечению инфраструктуры, необходимой для любого другого процесса. Инфраструктура может содержать технические и программные средства, инструментальные средства, методики, стандарты и условия для разработки, эксплуатации или сопровождения системы;

o *усовершенствование* – работы по оценке, контролю и улучшению процессов жизненного цикла;

o *обучение* – работы по планированию и проведению обучения персонала, включая разработку учебных материалов. При этом под персоналом понимаются не только конечные пользователи, которые будут эксплуатировать систему, но и разработчики системы. Например, разработчики должны быть обучены технологиям и средствам программирования, принятым в организации, и даже обучены правильно внедрять и обучать конечных пользователей работе с системой. Как бы это ни парадоксально звучало, но обучать правильной методике и приемам обучения тоже необходимо.

***6.Модели жизненного цикла информационных систем.(16)***

***7.Типовая организация современной СУБД.***

**Типовая организация современной СУБД**

СУБД состоит из нескольких программных компонентов, каждый из которых нацелен на выполнение специфической операции.

Логически в современной реляционной СУБД можно выделить внутреннюю часть — ядро СУБД, компилятор языка БД (обычно SQL), подсистему поддержки времени выполнения, набор утилит.

Ядро СУБД отвечает за управление данными во внешней памяти, управление буферами оперативной памяти, управление транзакциями и журнализацию.

Ядро СУБД обладает собственным интерфейсом, не доступным пользователям напрямую и используемым в программах, производимых компилятором SQL и утилитах БД.

Ядро СУБД является основной резидентной частью СУБД. При использовании архитектуры «клиент-сервер» ядро является основной составляющей серверной части системы.

Обобщенная компонентная струк¬тура СУБД:

***8.Экспертные системы и главные признаки экспертных систем.***

**Экспе́ртная систе́ма (ЭС,**[**англ.**](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)***expert system*)** — компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации. Современные экспертные системы начали разрабатываться исследователями[искусственного интеллекта](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82) в 1970-х годах, а в 1980-х получили коммерческое подкрепление. Предшественники экспертных систем были предложены в [1832 году](https://ru.wikipedia.org/wiki/1832_%D0%B3%D0%BE%D0%B4) [С. Н. Корсаковым](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%81%D0%B0%D0%BA%D0%BE%D0%B2,_%D0%A1%D0%B5%D0%BC%D1%91%D0%BD_%D0%9D%D0%B8%D0%BA%D0%BE%D0%BB%D0%B0%D0%B5%D0%B2%D0%B8%D1%87), создавшим механические устройства, так называемые «интеллектуальные машины», позволявшие находить решения по заданным условиям, например определять наиболее подходящие лекарства по наблюдаемым у пациента симптомам заболевания[[1]](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-1).

В [информатике](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) экспертные системы рассматриваются совместно с [базами знаний](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D0%B8%D0%B9) как модели поведения [экспертов](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D1%86%D0%B5%D0%BD%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) в определенной области знаний с использованием процедур логического вывода и [принятия решений](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9), а базы знаний — как совокупность фактов и правил логического вывода в выбранной предметной области деятельности.

Похожие действия выполняет такой программный инструмент как «[Мастер](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%82%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82))» ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Wizard*). Мастера применяются как в [системных программах](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) так и в [прикладных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для упрощения интерактивного общения с пользователем (например, при [установке ПО](https://ru.wikipedia.org/wiki/%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)). Главное отличие мастеров от экспертных систем — отсутствие [базы знаний](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D0%B8%D0%B9) — все действия жестко запрограммированы. Это просто набор форм для заполнения пользователем.

Другие подобные программы — [поисковые](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) или [справочные](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%80%D0%B0%D0%B2%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) (энциклопедические) системы. По запросу пользователя они предоставляют наиболее подходящие ([релевантные](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D0%BD%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C)) разделы *базы статей* (представления об объектах областей знаний, их виртуальную модель).

В настоящее время «классическая» концепция экспертных систем, сложившаяся в 70-80 годах прошлого века, переживает серьезный кризис, по всей видимости связанный с её глубокой ориентацией на общепринятый в те годы [текстовый человеко-машинный интерфейс](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9_%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8), который в настоящее время в пользовательских приложениях почти полностью вытеснен графическим ([GUI](https://ru.wikipedia.org/wiki/GUI)). Кроме того, «классический» подход к построению экспертных систем плохо согласуется с [реляционной моделью данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), что делает невозможным эффективное использование современных промышленных [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) для организации баз знаний таких систем. Все приводимые в литературных и интернет-источниках примеры «известных» или «распространенных» экспертных систем на самом деле относятся к 80-м годам прошлого столетия и в настоящее время давно не существуют, либо безнадежно устарели и поддерживаются лишь немногочисленными энтузиастами.[[*источник не указан 790 дней*](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8)] С другой стороны, нередко в качестве маркетингового хода экспертными системами объявляются современные программные продукты, в «классическом» понимании таковыми не являющиеся (например, [компьютерные справочно-правовые системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%80%D0%B0%D0%B2%D0%BE%D1%87%D0%BD%D0%BE-%D0%BF%D1%80%D0%B0%D0%B2%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)). Предпринимаемые энтузиастами попытки объединить «классические» подходы к разработке экспертных систем с современными подходами к построению пользовательского интерфейса (проекты [CLIPS Java Native Interface](https://ru.wikipedia.org/w/index.php?title=CLIPS_Java_Native_Interface&action=edit&redlink=1), [CLIPS.NET](https://ru.wikipedia.org/w/index.php?title=CLIPS.NET&action=edit&redlink=1) и др.) не находят поддержки среди крупных компаний-производителей программного обеспечения и по этой причине остаются пока в экспериментальной стадии.

***9.Основные функции экспертных систем.***

ЭС может функционировать в 2-х режимах.

1. **Режим ввода знаний** — в этом режиме эксперт с помощью инженера по знаниям посредством редактора базы знаний вводит известные ему сведения о предметной области в базу знаний ЭС.
2. **Режим консультации** — пользователь ведет диалог с ЭС, сообщая ей сведения о текущей задаче и получая рекомендации ЭС. Например, на основе сведений о физическом состоянии больного ЭС ставит диагноз в виде перечня заболеваний, наиболее вероятных при данных симптомах.

***10.Реляционная модель данных.***

**Реляционная модель данных** (РМД) — [логическая](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0) [модель данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), прикладная [теория](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F) построения [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), которая является приложением к задачам обработки данных таких разделов [математики](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), как [теория множеств](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2) и [логика первого порядка](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%BA%D0%B0).

На реляционной модели данных строятся [реляционные базы данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

Реляционная модель данных включает следующие компоненты:

* [Структурный](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0) аспект (составляющая) — данные в базе данных представляют собой набор [отношений](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C)).
* Аспект (составляющая) [целостности](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) — отношения (таблицы) отвечают определенным условиям [целостности](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). РМД поддерживает декларативные [ограничения целостности](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D1%8F_%D1%86%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B8&action=edit&redlink=1) уровня [домена](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD_(%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)) (типа данных), уровня отношения и уровня базы данных.
* Аспект (составляющая) обработки (манипулирования) — РМД поддерживает операторы манипулирования отношениями ([реляционная алгебра](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0), [реляционное исчисление](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5)).

Кроме того, в состав реляционной модели данных включают теорию [нормализации](https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0).

Термин «реляционный» означает, что теория основана на математическом понятии [отношение](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2)) (*relation*). В качестве неформального синонима [термину](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%80%D0%BC%D0%B8%D0%BD) «отношение» часто встречается слово [таблица](https://ru.wikipedia.org/wiki/Table_(database)). Необходимо помнить, что «таблица» есть понятие нестрогое и неформальное и часто означает не «отношение» как [абстрактное понятие](https://ru.wikipedia.org/w/index.php?title=%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D0%BE%D0%BD%D1%8F%D1%82%D0%B8%D0%B5&action=edit&redlink=1), а [визуальное представление](https://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) отношения на бумаге или экране. Некорректное и нестрогое использование термина «таблица» вместо термина «отношение» нередко приводит к недопониманию. Наиболее частая ошибка состоит в рассуждениях о том, что РМД имеет дело с «плоскими», или «двумерными» таблицами, тогда как таковыми могут быть только визуальные представления таблиц. Отношения же являются абстракциями и не могут быть ни «плоскими», ни «неплоскими».

Для лучшего понимания РМД следует отметить три важных обстоятельства:

* модель является логической, то есть отношения являются логическими (абстрактными), а не физическими (хранимыми) структурами;
* для реляционных баз данных верен [информационный](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) [принцип](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF): всё информационное наполнение базы данных представлено одним и только одним способом, а именно — явным заданием значений атрибутов в [кортежах](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%82%D0%B5%D0%B6_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) отношений; в частности, нет никаких указателей (адресов), связывающих одно значение с другим;
* наличие реляционной алгебры позволяет реализовать [декларативное программирование](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и декларативное описание ограничений целостности, в дополнение к навигационному (процедурному) программированию и процедурной проверке условий.

***11.(10)***

***12.Понятие и сущность проектирования ИС.***

*Информационная система*(ИС) *—*этовзаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели

ИС представляет собой совокупность организационных, технических, программных и информационных средств, объединенных в единую систему с целью сбора, хранения, обработки и выдачи необходимой информации, предназначенной для выполнения функций управления.

К ИС предъявляются следующие требования:

1) Полнота и достоверность информации для реализации функций управления.

2) Своевременность предоставления информации.

3) Обеспечение необходимой степени достоверности информации в зависимости от уровня управления.

4) Экономичность обработки информации – это значит, что затраты на обработку данных не должны превышать получаемый эффект.

5) Адаптивность к изменениям информационным требованиям пользователей.

Внедрение ИС проводится с целью повышения эффективности производственно-хозяйственной деятельности фирмы за счет принципиально новых методов управления, основанных на моделировании деятельности специалистов фирмы при принятии решений (методы искусственного интеллекта, экспертные системы и т.п.) использование современных средств телекоммуникации (e-mail, теле конференции) и вычислительных систем, а также сокращение времени выполнения типовых операций по обработке различного рода документов.

Информационная система — совокупность информационных, экономико-математических методов и моделей, технических, программных, технологических средств и специалистов, предназначенная для сбора, хранения, обработки и выдачи информации и принятия управленческих решений.

***13.Фундаментальные понятия реляционной базы данных: объект, связь, атрибут.***

Домен — это набор всех допустимых значений, которые может содержать атрибут. Понятие «домен» часто путают с понятием «тип данных». Необходимо различать эти два понятия. Тип данных — это физическая концепция, а домен — логическая. Например, «целое число» — это тип данных, а «возраст» — это домен.

Связи — на концептуальном уровне представляют собой простые ассоциации между сущностями. Например, утверждение «Покупатели приобретают продукты» указывает, что между сущностями «Покупатели» и «Продукты» существует связь, и такие сущности называются участниками этой связи.

Сущность — некоторый обособленный объект или событие, информацию о котором необходимо сохранять в базе данных, имеющий определенный набор свойств — атрибутов. Сущности могут быть как физические (реально существующие объекты: например, СТУДЕНТ, атрибуты — номер зачетной книжки, фамилия, его факультет, специальность, номер группы и т. д.), так и абстрактные (например, ЭКЗАМЕН, атрибуты — дисциплина, дата, преподаватель, аудитория и пр.). Для сущностей различают ее тип и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр — конкретными значениями свойств.

***14. Содержание процесса проектирования. Цель проектирования ИС.***

**Принципы системного проектирования**[[править](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&veaction=edit&vesection=10) | [править вики-текст](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&section=10)]

Системное проектирование должно базироваться на системном подходе. На сегодняшний день нельзя утверждать, что известен его полный состав и содержание применительно к проектной деятельности, однако можно сформулировать наиболее важные из них:

* Практическая полезность:
  + деятельность должна быть *целенаправленной*, устремленной на удовлетворение действительных потребностей реального потребителя или определенной социальной, возрастной или иной групп людей;
  + деятельность должна быть *целесообразной*. Важно вскрыть причины, препятствующие использованию существующих объектов для удовлетворения новых потребностей, выявить вызывающие их ключевые противоречия и сконцентрировать усилия на решении главных задач;
  + деятельность должна быть *обоснованной и эффективной*. Разумным будет использование не любого решения задачи, а поиск [*оптимального варианта*](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B5);
* Единство составных частей:
  + целесообразно любой объект, сложный ли он или простой, рассматривать как *систему*, внутри которой можно выделить логически связанные более простые части — [*подсистемы*](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D1%8F), единство частных свойств которых и образует качественно новые свойства объекта-системы;
  + разрабатываемые объекты предназначены для людей, ими создаются и эксплуатируются. Поэтому человек также обязан рассматриваться в качестве одной из взаимодействующих систем. При этом должно приниматься во внимание не только физическое взаимодействие, но и духовно-эстетическое воздействие;
  + внешняя, или как её ещё называют — *жизненная среда*, также должна рассматриваться в качестве системы, взаимосвязанной с проектируемым объектом;
* Изменяемость во времени:
  + учёт этапов [жизненного цикла](https://ru.wikipedia.org/wiki/%D0%96%D0%B8%D0%B7%D0%BD%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%86%D0%B8%D0%BA%D0%BB_%D0%B8%D0%B7%D0%B4%D0%B5%D0%BB%D0%B8%D1%8F) объекта;
  + учёт истории и перспектив развития и применения разрабатываемого объекта, а также областей науки и техники, на достижениях которых базируются соответствующие разработки.

***15.Средства проектирования ИС***.

Под **средствами проектирования информационных систем** (СП ИС) будем понимать комплекс инструментальных средств, обеспечивающих в рамках выбранной методологии проектирования поддержку полного жизненного цикла (ЖЦ) ИС.

Они включают в себя, как правило, *стратегическое планирование, анализ, проектирование, реализацию, внедрение и эксплуатацию*.

Каждый этап характеризуется определёнными задачами и методами их решения, исходными данными, полученными на предыдущем этапе, и результатами. При анализе СП их следует рассматривать не локально, а в комплексе, что позволяет реально охарактеризовать их достоинства, недостатки и место в общем технологическом цикле создания ИС.

Необходимая аппаратная платформа для ИС может формироваться из компонентов различных фирм-производителей. Однако выбрать и скомплексировать разные инструментальные средства, каждое из которых может являться одним из мировых лидеров в своём классе, достаточно трудно, а порой и невозможно.

16.Модель жизненного цикла и технология проектирования.

Каскадная стратегия (однократный проход, водопадная или классическая модель) подразумевает линейную последовательность выполнения стадий создания информационной системы (рис.3.1). Другими словами, переход с одной стадии на следующую происходит только после того, как будет полностью завершена работа на текущей.

Данная модель применяется при разработке информационных систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования.

***Достоинства модели:***

- на каждой стадии формируется законченный набор документации, программного и аппаратного обеспечения, отвечающий критериям полноты и согласованности;

- выполняемые в четкой последовательности стадии позволяют уверенно планировать сроки выполнения работ и соответствующие ресурсы (денежные, материальные и людские).

***Недостатки модели:***

- реальный процесс разработки информационной системы редко полностью укладывается в такую жесткую схему. Особенно это относится к разработке нетиповых и новаторских систем;

- жизненный цикл основан на точной формулировке исходных требований к информационной системе. Реально в начале проекта требования заказчика определены лишь частично;

- основной недостаток – результаты разработки доступны заказчику только в конце проекта. В случае неточного изложения требований или их изменения в течение длительного периода создания ИС заказчик получает систему, не удовлетворяющую его потребностям.

Инкрементная стратегия (англ. increment – увеличение, приращение) подразумевает разработку информационной системы с линейной последовательностью стадий, но в несколько инкрементов (версий), т. е. с запланированным улучшением продукта.

В начале работы над проектом определяются все основные требования к системе, после чего выполняется ее разработка в виде последовательности версий. При этом каждая версия является законченным и работоспособным продуктом. Первая версия реализует часть запланированных возможностей, следующая версия реализует дополнительные возможности и т. д., пока не будет получена полная система.

Данная модель жизненного цикла характерна при разработке сложных и комплексных систем, для которых имеется четкое видение (как со стороны заказчика, так и со стороны разработчика) того, что собой должен представлять конечный результат (информационная система). Разработка версиями ведется в силу разного рода причин:

- отсутствия у заказчика возможности сразу профинансировать весь дорогостоящий проект;

- отсутствия у разработчика необходимых ресурсов для реализации сложного проекта в сжатые сроки;

- требований поэтапного внедрения и освоения продукта конечными пользователями. Внедрение всей системы сразу может вызвать у ее пользователей неприятие и только «затормозить» процесс перехода на новые технологии. Образно говоря, они могут просто «не переварить большой кусок, поэтому его надо измельчить и давать по частям».

***Достоинства*** и ***недостатки*** этой стратегии такие же, как и у классической. Но в отличие от классической стратегии заказчик может раньше увидеть результаты. Уже по результатам разработки и внедрения первой версии он может незначительно изменить требования к разработке, отказаться от нее или предложить разработку более совершенного продукта с заключением нового договора.

 Спиральная стратегия (эволюционная или итерационная модель, автор Барри Боэм, 1988 г.) подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий.

Данная модель жизненного цикла характерна при разработке новаторских (нетиповых) систем. В начале работы над проектом у заказчика и разработчика нет четкого видения итогового продукта (требования не могут быть четко определены) или стопроцентной уверенности в успешной реализации проекта (риски очень велики). В связи с этим принимается решение разработки системы по частям с возможностью изменения требований или отказа от ее дальнейшего развития. Как видно из рис.3.3, развитие проекта может быть завершено не только после стадии внедрения, но и после стадии анализа риска.

***Достоинства модели:***

- позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований;

- допускает изменение требований при разработке информационной системы, что характерно для большинства разработок, в том числе и типовых;

- обеспечивает большую гибкость в управлении проектом;

- позволяет получить более надежную и устойчивую систему. По мере развития системы ошибки и слабые места обнаруживаются и исправляются на каждой итерации;

- позволяет совершенствовать процесс разработки – анализ, проводимый в каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации;

- уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта.

***Недостатки модели:***

- увеличивается неопределенность у разработчика в перспективах развития проекта. Этот недостаток вытекает из предыдущего достоинства модели;

- затруднены операции временного и ресурсного планирования всего проекта в целом. Для решения этой проблемы необходимо ввести временные ограничения на каждую из стадий жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа выполнена. План составляется на основе статистических данных, полученных в предыдущих проектах и личного опыта разработчиков.

***17.Этапы и стадии проектирования информационных систем.***

Объектно-ориентированная технология проектирования ИС предоставляет мощную, гибкую, универсальную концептуальную основу для конструирования информационно-управляющих систем в различных областях хозяйственной деятельности и управления, сочетающую использование моделей современной логистики, объектного подхода к компонентам предметной области, современных инструментальных средств визуального программирования и СУБД с SQL-интерфейсом. Объектно-ориентированная технология проектирования ИС включает в себя следующие компоненты:

• технологию конструирования концептуальной объектно- ориентированной модели предметной области;

• инструментальные средства спецификации проектных решений;

• библиотеки типовых компонент модели предметной области;

• типовые проектные решения для ряда функциональных областей

Функционально-модульный, или структурный, определяется принципом алгоритмической декомпозиции. В соответствии с этим принципом осуществляется разделение функций ИС на модули по функциональной принадлежности, и каждый модуль реализует один из этапов общего процесса. Функционально-модульный подход к проектированию ИС, получивший название «модель водопада», предусматривает строго последовательный порядок действий. Главный недостаток такого подхода заключается в движении информации в одном направлении. Если при проектировании или эксплуатации возникает проблема, то она решается только на данной стадии проекта, не затрагивая предыдущих стадий. Недостаточная обратная связь приводит к ограниченным исправлениям, что, в свою очередь, приводит к деформированным реализациям. Ориентация на функционально-модульный подход увеличивает вероятность потери контроля над решением возникающих проблем.

***18.Архитектура CASE-средств.***

Репозиторий данных

Является специализированной базой данных для отображения состояния проектируемой ЭИС в любой момент времени. В нём хранится информация об объектах проектироуемой системы и все подсистемы обмениваются данными с ним.

Графический редактор диаграмм

Графический редактор диаграмм предназначен для отображения в графическом виде в заданной нотации проектируемой ЭИС. Он позволяет:

* создавать элементы диаграмм и взаимосвязи между ними
* задавать описания элементов диаграмм
* задавать описания связей между элементами диаграмм
* редактировать элементы диаграмм, их взаимосвязи и описания

Верификатор диаграмм

Верификатор диаграмм служит для контроля правильности построения диаграмм в заданной методологии проектирования ЭИС. Он выполняет:

* мониторинг правильности построения диаграмм
* диагностику и выдачу сообщений об ошибках
* выделение на диаграмме ошибочных элементов

Документатор проекта

Документатор проекта позволяет получать информацию о состоянии проекта в виде различных отчётов. Отчёты могут строиться по нескольким признакам, например по времени, автору, элементам диаграмм, диаграмме или проекту в целом.

Администратор проекта

Администратор проекта представляет собой инструменты, необходимые для выполнения следующих административных функций:

* инициализация проекта
* задания начальных параметров проекта
* назначения и изменения прав доступа к элементам проекта
* мониторинга выполнения работ

Сервис

Сервис представляет собой набор системных утилит по обслуживанию репозитория. Данные утилиты выполняют функции архивации данных, восстановления данных и создания нового репозитория.

***19. Основные понятия технологии проектирования информационных систем.(17)***

***20. Типы данных в базе данных, основные характеристики.***

* Тип данных определяется значениями, которые предполагается хранить в поле, и операциями, которые будут выполняться с этими значениями. В Access допускается использование двенадцати типов данных. **екстовый** (Text) — используется для хранения текста или комбинаций алфавитно-цифровых знаков, не применяемых в расчетах (например, код товара). Максимальная длина поля 255 знаков.
* **Поле МЕМО** (Memo) — используется для хранения обычного текста или комбинаций алфавитно-цифровых знаков длиной более 255 знаков. Поля с этим типом данных в базах данных формата Access 2007 поддерживают также форматирование текста. Это единственный в Access тип данных, обеспечивающий встроенную поддержку отображения и хранения форматированного текста. Максимальный размер поля 1 Гбайт знаков или 2 Гбайт памяти (2 байта на знак) при программном заполнении полей, и 65 535 знаков при вводе данных вручную в поле и в любой элемент управления, связанный с этим полем.
* **Числовой** (Number) — служит для хранения числовых значений (целых или дробных), предназначенных для вычислений, исключением являются денежные значения, для которых используется тип данных **Денежный**(Currency). Размер поля 1, 2, 4 и 8 байтов, или 16 байтов (если используется для кода репликации) зависит от типа чисел, вводимых в поле.
* **Дата/время** (Date/Time) — используется для хранения значений даты и времени в виде 8-байтовых чисел двойной точности с плавающей запятой. Целая часть значения, расположенная слева от десятичной запятой, представляет собой дату. Дробная часть, расположенная справа от десятичной запятой, — это время. Хранение значений даты и времени в числовом формате позволяет выполнять различные вычисления с этими данными.
* **Денежный** (Currency) — используется для хранения денежных значений в виде 8-байтовых чисел с точностью до четырех знаков после запятой. Этот тип данных применяется для хранения финансовых данных и в тех случаях, когда значения не должны округляться.
* **Счетчик** (AutoNumber) — используется для уникальных числовых 4-байтовых значений, которые автоматически вводит Access при добавлении записи. Вводимые числа могут последовательно увеличиваться на указанное приращение или выбираться случайно. Обычно используются в первичных ключах.
* **Логический** (Yes/No) — применяется для хранения логических значений, которые могут содержать одно из двух значений: Да/Нет, Истина/Ложь или Вкл/Выкл. (8 битов = 1 байт). Используется 1 для значений Да и 0 для значений Нет. Размер равен 1 биту.
* **Поле объекта OLE** (OLE Object) — используется для хранения изображений, документов, диаграмм и других объектов из приложений MS Office и других программ Windows в виде растровых изображений, которые затем отображаются в элементах управления форм или отчетов, связанных с этим полем таблицы.  
  Чтобы в Access просматривать эти изображения, необходимо, чтобы на компьютере, использующем базу данных, был зарегистрирован OLE-сервер (про-грамма, поддерживающая этот тип файлов). Если для данного типа файлов OLE-сервер не зарегистрирован, отображается значок поврежденного изображения.
* **Гиперссылка** (Hyperlink) — применяется для хранения ссылок на Web-узлы (URL-адреса), на узлы или файлы интрасети или локальной сети (UNC-адреса — стандартного формата записи пути), а также на узлы или файлы локального компьютера. Кроме того, можно использовать ссылку на объекты Access, хранящиеся в базе данных. Может хранить до 1 Гбайт данных.
* **Вложение** (Attachment) — используется для вложения в поле записи файлов изображений, электронных таблиц, документов, диаграмм и других файлов поддерживаемых типов точно так же, как в сообщения электронной почты. Вложенные файлы можно просматривать и редактировать в соответствии с заданными для поля параметрами. Эти поля не имеют ограничений, связанных с отсутствием зарегистрированных OLE-серверов. Более рационально используют место для хранения, чем поля с типом данных **Поле объекта OLE** (OLE Object), поскольку не создают растровые изображения исходного файла. Максимальная длина поля для сжатых вложений — 2 Гбайт, для несжатых — примерно 700 Кбайт в зависимости от степени возможного сжатия вложения.
* **Вычисляемый** (Calculated) — предназначен для создания вычисляемых полей: числовых, текстовых, денежных, дата/время, логических. Значение вычисляемого поля определяется выражением, записанным в поле и использующим другие поля текущей записи, некоторые встроенные функции и константы, связанные арифметическими, логическими или строковыми операторами.
* **Мастер подстановок** (Lookup Wizard) или **Подстановка и отношения** (Lookup & Relationship) — вызывает мастера подстановок, с помощью которого можно создать поле, позволяющее выбрать значения из списка, построенного на основе значений поля другой таблицы, запроса или фиксированного набора значений. Такое поле отображается как поле со списком. Если список построен на основе поля таблицы или запроса, тип данных и размер создаваемого поля определяется типом данных и размером привязанного столбца; если на основе набора значений — размером текстового поля, содержащего значение. Кроме того, мастер подстановок позволяет определить связь таблиц и включить проверку связной целостности данных.

***21. Операторы определения и манипулирования схемой БД.***

Операторы манипулирования данными UPDATE и DELETE построены на тех же принципах, что и оператор выборки данных SELECT. Набор кортежей указанного отношения, подлежащих модификации или удалению, определяется входящим в соответствующий оператор логическим выражением, которое может включать сложные предикаты, в том числе и с вложенными подзапросами.

В операторе вставки кортежа(ей) в указанное отношение заносимый кортеж может задаваться как в литеральной форме, так и с помощью внутреннего подоператора выборки.

***22. Системы поддержки принятия решений.***

Система поддержки принятия решений предназначена для поддержки многокритериальных решений в сложной информационной среде. При этом под многокритериальностью понимается тот факт, что результаты принимаемых решений оцениваются не по одному, а по совокупности многих показателей (критериев) рассматриваемых одновременно. Информационная сложность определяется необходимостью учета большого объема данных, обработка которых без помощи современной вычислительной техники практически невыполнима. В этих условиях число возможных решений, как правило, весьма велико, и выбор наилучшего из них "на глаз", без всестороннего анализа может приводить к грубым ошибкам.

Система поддержки решений СППР решает две основные задачи:

* выбор наилучшего решения из множества возможных (оптимизация),
* упорядочение возможных решений по предпочтительности (ранжирование).

***23. Язык структурированных запросов SQL (Structured Query Language).***

SQL (англ. Structured Query Language — язык структурированных запросов) — универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

Вопреки существующим заблуждениям, SQL в его чистом (базовом) виде является информационно-логическим языком, а не языком программирования. Вместе с тем стандарт языка спецификацией SQL/PSM предусматривает возможность его процедурных расширений, с учётом которых язык уже вполне может рассматриваться в качестве языка программирования.

***24. Многоуровневая архитектура.***

**Трёху́ровневая архитекту́ра** (*трёхзве́нная архитекту́ра*, [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *three-tier*) — [архитектурная модель](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) программного комплекса, предполагающая наличие в нём трёх компонентов: [клиента](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), [сервера приложений](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) (к которому подключено клиентское приложение) и [сервера баз данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) (с которым работает сервер приложений).

[Клиент](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) (*слой клиента*) — это интерфейсный (обычно [графический](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F)) компонент комплекса, предоставляемый конечному пользователю. Этот уровень не должен иметь прямых связей с базой данных (по требованиям безопасности и масштабируемости), быть нагруженным основной [бизнес-логикой](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0" \o "Бизнес-логика) (по требованиям [масштабируемости](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D0%BE%D1%81%D1%82%D1%8C" \o "Масштабируемость)) и хранить состояние приложения (по требованиям надёжности). На этот уровень обычно выносится только простейшая бизнес-логика: интерфейс [авторизации](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), алгоритмы шифрования, проверка вводимых значений на допустимость и соответствие формату, несложные операции с данными (сортировка, группировка, подсчёт значений), уже загруженными на терминал.

[Сервер приложений](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) (*средний слой*, *связующий слой*) располагается на втором уровне, на нём сосредоточена бо́льшая часть бизнес-логики. Вне его остаются только фрагменты, экспортируемые на клиента (терминалы), а также элементы логики, погруженные в базу данных (хранимые процедуры и триггеры). Реализация данного компонента обеспечивается [связующим программным обеспечением](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D1%8F%D0%B7%D1%83%D1%8E%D1%89%D0%B5%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Серверы приложений проектируются таким образом, чтобы добавление к ним дополнительных экземпляров обеспечивало [горизонтальное масштабирование](https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D1%80%D0%B8%D0%B7%D0%BE%D0%BD%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BC%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)производительности программного комплекса и не требовало внесения изменений в программный код приложения.

Сервер баз данных (*слой данных*) обеспечивает хранение данных и выносится на отдельный уровень, реализуется, как правило, средствами [систем управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), подключение к этому компоненту обеспечивается только с уровня сервера приложений.

В простейших конфигурациях все компоненты или часть из них могут быть совмещены на одном вычислительном узле. В продуктивных конфигурациях как правило используется выделенный вычислительный узел для сервера баз данных или [кластер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80_(%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BE%D0%B2)) серверов баз данных, для серверов приложений — выделенная группа вычислительных узлов, к которым непосредственно подключаются клиенты (терминалы).

***25. Автоматизированные информационно-вычислительные системы.***

Автоматизированные информационно-вычислительные системы (АИВС) предназначены для решения сложных в математическом отношении задач, требующих больших объемов разнообразной информации с проведением различных расчетов. Они могут использоваться как подсистемы автоматизированных систем управления и систем поддержки принятия решений в тех случаях, когда выработка управленческих решений должна опираться на сложные вычисления. В зависимости от специфики области деятельности, в которых применяются эти системы, различают несколько их видов.

***26. Архитектура CASE-средств (18)***

***27.Информационно-расчетные системы.(12)***

***28.Архитектура CASE-средств(18)***

***29.Основные функции СУБД.***

Основные функции СУБД

* управление данными во [внешней памяти](https://ru.wikipedia.org/wiki/%D0%92%D0%BD%D0%B5%D1%88%D0%BD%D1%8F%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) (на дисках);
* управление данными в [оперативной памяти](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) с использованием [дискового кэша](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BA%D1%8D%D1%88);
* [журнализация изменений](https://ru.wikipedia.org/wiki/%D0%96%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9), [резервное копирование](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B7%D0%B5%D1%80%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BA%D0%BE%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [восстановление базы данных](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D1%81%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) после сбоев;
* поддержка языков БД ([язык определения данных](https://ru.wikipedia.org/wiki/DDL), [язык манипулирования данными](https://ru.wikipedia.org/wiki/DML)).

***30.Классицикация информационных систем по сфере применения.***

Поскольку ИС создаются для удовлетворения информационных потребностей в рамках конкретной предметной области, то каждой предметной области (сфере применения) соответствует свой тип ИС. Перечислять все эти типы не имеет смысла, так как количество предметных областей велико, но можно указать в качестве примера следующие типы ИС:

* [Экономическая информационная система](https://ru.wikipedia.org/w/index.php?title=%D0%AD%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0&action=edit&redlink=1) — информационная система, предназначенная для выполнения функций управления на предприятии.
* [Медицинская информационная система](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) — информационная система, предназначенная для использования в лечебном или лечебно-профилактическом учреждении.
* [Географическая информационная система](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) — информационная система, обеспечивающая сбор, хранение, обработку, доступ, отображение и распространение пространственно-координированных данных (пространственных данных).

***31.Информационные системы и их классификация. Основные понятия.(1)***

***32.Системы поддержки принятия решений.(22)***

***33.Классификация рынка информационных систем.***

Развитие рынка компьютерных систем, способных обеспечить   
эффективное управление организацией, связано с системами двух   
классов:

- системы, ориентированные на автоматизацию отдельных функций управления;   
- интегрированные системы управления.

На рынке представлены как российские, так и зарубежные   
системы различного назначения. Если говорить о реальном сек-  
торе экономики, то для него на рынке имеются системы двух   
классов:

- финансово-управленческие системы;   
- производственные системы.   
**Финансово-управленческие системы:   
-**локальные;   
- малые интегрированные.

**Назначение:**ведение учета по одному или нескольким направ-  
лениям (бухгалтерия, сбыт, склады, учет кадров и т.д.), управле-  
ние финансовыми потоками.

**Свойства систем:**

- Универсальность.

-   
Небольшой цикл внедрения.

- Имеются «коробочные» варианты.   
- Гибкость в адаптации к нуждам конкретного предприятия.   
- Способность работать на персональных компьютерах в обычных сетях передачи данных *Novell Netware*или *Wtndows NT.   
-*Использование простых средств разработки *(Clipper, FoxPro, dBase, Paradox).   
-*Снижение уровня эффективности при работе на сложных конфигурациях сети и при увеличении объемов обрабатывае-  
мых данных.

**Производственные системы:   
-**средние;   
- крупные интегрированные.

**Назначение:**управление и планирование производственного   
процесса. Учетные функции глубоко проработаны и выполняют   
вспомогательную роль.

***34.Типовая организация современной СУБД.(7)***

***35.Этапы создания информационных систем*.**

Стадии создания» выделяют следующие основные стадии создания и этапы разработки автоматизированной системы (АС):

1. Формирование требований к АС.
2. Разработка концепции АС.
3. [Техническое задание](http://www.prj-exp.ru/patterns/pattern_tech_task.php).
4. [Эскизный проект](http://www.prj-exp.ru/patterns/pattern_draft_project.php).
5. [Технический проект](http://www.prj-exp.ru/patterns/pattern_tech_project.php).
6. Рабочая документация.
7. Ввод в действие.
8. Сопровождение АС.

Как правило, этапы стадий формирования требований к автоматизированной системе объединяют с этапами разработки технического задания, а этапы разработки концепции - с этапами эскизного проектирования. Также к приведенным стадиям создания АС добавляют стадию подготовки к началу проекта.

***36.Экспертные системы и их классификация.(8)***

***37.Виды качественных исследований: интервьюирование, анкетирование и др.???***

***38.Классификация технологий проектирования информационных систем.***

***Технологии проектирования информационных систем***

Технология проектирования ИС – совокупность методов и средств проектирования ИС, а также организации и управления, внедрения и модернизации проекта. ИС.

Организация проектирования ИС предполагает использование определенной совокупности методов проектирования.

***Методы проектирования принято классифицировать по различным признакам:***

1) По степени автоматизации разработки проектных решений:

1.1) Ручное (традиционное) проектирование

1.2) Методы автоматизированного проектирования

2) По степени типизации проектных решений

2.1) Методы оригинального (индивидуального) проектирования

2.2) Методы типового проектирования

3) По степени адаптивности проектных решений

3.1) Методы реконструкции – адаптация проектных решений выполняется путем изменения соответствующих компонентов готовой системы.

3.2) Методы параметризации – изменение проектных решений в соответствии с новыми параметрами объекта проектирования

3.3) Методы реструктуризации – изменение проектных решений в связи с изменением модели ПО.

***39.Принципы создания ИС на основе CASE-технологий.***

Существует несколько принципов CASE-технологий. Рассмотрим основные **принципы:**

1. Принцип всесторонней компьютерной поддержки проектирования. CASE-технология – это разновидность САПРв области создания ИС.
2. Принцип модельного подхода – это может быть методология функционально ориентированного подхода или методология объектно-ориентированного подхода.

3. Иерархическое представление модели предметной области. Суще­ствуют плоские модели, предусматривающие представление всей модели в виде единого листа, Но когда встречаются сложные си­стемы, то возникают определенные трудности. Преодолеть эти трудности позволяют иерархические модели, в которых предусмотрена иерархическая последовательность детализации (декомпозиции) описания системы. Эти модели соответствуют принципу проекти­рования «сверху вниз», от общего к частному.

4. Наглядность представления модели, т.е. наличие визуальных средств проектирования. Это связано с тем, что процесс построения модели ИС так и не удается формализовать до конца и в этом процессе должен принимать участие человек. Гра­фические средства обозначения и правила, предназначенные для описания структуры системы, этапов обработки информации пред­ставляют собой нотации CASE-технологии. Нотации включают гра­фы, диаграммы, таблицы, формальные и естественные языки. Их использование является существенной особенностью CASE-технологии. Поэтому CASE-технология предусматривает четырехуровне­вую парадигму проектирования, в которой важное место отводится нотациям:

***40.Средства проектирования ИС.(15)***

***41.Разработка CASE-модели.***

Основной целью CASE-технологии является разграничение процесса проектирования программных продуктов от процесса кодирования и последующих этапов разработки, максимально автоматизировать процесс разработки. Для выполнения поставленной цели CASE-технологии используют два принципиально разных подхода к проектированию: структурный и объектно-ориентированный.

Структурный подход предполагает декомпозицию (разделение) поставленной задачи на функции, которые необходимо автоматизировать. В свою очередь, функции также разбиваются на подфункции, задачи, процедуры. В результате получается упорядоченная иерархия функций и передаваемой информацией между функциями.

Структурный подход подразумевает использование определенных общепринятых методологий при моделировании различных информационных систем:

1. [SADT](https://ru.wikipedia.org/wiki/SADT) (Structured Analysis and Design Technique);
2. [DFD](https://ru.wikipedia.org/wiki/DFD) (Data Flow Diagrams);
3. [ERD](https://ru.wikipedia.org/wiki/ERD) (Entity-Relationship Diagrams).

Существует три основных типа моделей, используемых при структурном подходе: функциональные, информационные и структурные.

Основным инструментом объектно-ориентированного подхода является язык [UML](https://ru.wikipedia.org/wiki/UML) — унифицированный язык моделирования, который предназначен для визуализации и документирования объектно-ориентированных систем с ориентацией их на разработку программного обеспечения. Данный язык включает в себя систему различных диаграмм, на основании которых может быть построено представление о проектируемой системе.

***42. Основные понятия технологии проектирования информационных систем.(17)***

***43. Архитектура CASE-средств. Локальные CASE-средства. Малые интегрированные CASE-средства.***

Классификация по типам в основном совпадает с компонентным составом CASE-средств и включает следующие основные типы:

* средства анализа (Upper CASE), предназначенные для построения и анализа моделей предметной области (Design/IDEF (Meta Software), BPwin (Logic Works));
* средства анализа и проектирования (Middle CASE), поддерживающие наиболее распространенные методологии проектирования и использующиеся для создания проектных спецификаций (Vantage Team Builder (Cayenne), Designer/2000 (ORACLE), Silverrun (CSA), PRO-IV (McDonnell Douglas), CASE.Аналитик (МакроПроджект)). Выходом таких средств являются спецификации компонентов и интерфейсов системы, архитектуры системы, алгоритмов и структур данных;
* средства проектирования баз данных, обеспечивающие моделирование данных и генерацию схем баз данных (как правило, на языке SQL) для наиболее распространенных СУБД. К ним относятся ERwin (Logic Works), S-Designor (SDP) и DataBase Designer (ORACLE). Средства проектирования баз данных имеются также в составе CASE-средств Vantage Team Builder, Designer/2000, Silverrun и PRO-IV;
* средства разработки приложений. К ним относятся средства 4GL (Uniface (Compuware), JAM (JYACC), PowerBuilder (Sybase), Developer/2000 (ORACLE), New Era (Informix), SQL Windows (Gupta), Delphi (Borland) и др.) и генераторы кодов, входящие в состав Vantage Team Builder, PRO-IV и частично - в Silverrun;
* средства реинжиниринга, обеспечивающие анализ программных кодов и схем баз данных и формирование на их основе различных моделей и проектных спецификаций. Средства анализа схем БД и формирования ERD входят в состав Vantage Team Builder, PRO-IV, Silverrun, Designer/2000, ERwin и S-Designor. В области анализа программных кодов наибольшее распространение получают объектно-ориентированные CASE-средства, обеспечивающие реинжиниринг программ на языке С++ (Rational Rose (Rational Software), Object Team (Cayenne)).

Интегрированное CASE-средство (или комплекс средств, поддерживающих полный ЖЦ ПО) содержит следующие компоненты;

* репозиторий, являющийся основой CASE-средства. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость;
* графические средства анализа и проектирования, обеспечивающие создание и редактирование иерархически связанных диаграмм (DFD, ERD и др.), образующих модели ИС;
* средства разработки приложений, включая языки 4GL и генераторы кодов;
* средства конфигурационного управления;
* средства документирования;
* средства тестирования;
* средства управления проектом;
* средства реинжиниринга.

***44. Основные процессы жизненного цикла информационных систем.(5)***

***45. Фундаментальные понятия реляционной базы данных: объект, связь, атрибут.(13)***

***46. Архитектура информационных систем.(2)***

***47. Информационные системы и их классификации. Организация системы. Целостность системы.***

Методы и способы реализации требований, изложенных в определениях термина, подробно описываются в рамках единой схемы обеспечения [информационной безопасности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C) объекта ([защиты информации](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D1%89%D0%B8%D1%82%D0%B0_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8)).

Основными методами обеспечения целостности информации (данных) при хранении в автоматизированных системах являются:

* **обеспечение отказоустойчивости** ([резервирование](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B7%D0%B5%D1%80%D0%B2%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), дублирование, [зеркалирование](https://ru.wikipedia.org/wiki/%D0%97%D0%B5%D1%80%D0%BA%D0%B0%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5" \o "Зеркалирование) оборудования и данных, например через использование [RAID](https://ru.wikipedia.org/wiki/RAID)-массивов);
* **обеспечение безопасного восстановления** ([резервное копирование](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B7%D0%B5%D1%80%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BA%D0%BE%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [электронное архивирование](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B0%D1%80%D1%85%D0%B8%D0%B2%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) информации).

Одним из действенных методов реализации требований целостности информации при её передаче по линиям связи является [**криптографическая защита информации**](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) ([шифрование](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [хеширование](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [электронная цифровая подпись](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D1%8C)).

При комплексном подходе к защите бизнеса, направление обеспечения целостности и доступности информации (ресурсов бизнес-процессов) перерастает в план мероприятий, направляемых на **обеспечение непрерывности бизнеса**

**Организация системы** – внутренняя упорядоченность, согласованность взаимодействия элементов системы, проявляющаяся, в частности, в ограничении разнообразия состояния элементов в рамках системы.

***48. Классификация ИС по сфере применения. Классификация ИС по масштабу.(30)***