**Инструментальные средства разработки программ**

В процессе разработки программных средств в той или иной мере используется компьютерная поддержка процессов разработки ПС.

  Это достигается путем представления хотя бы некоторых программных документов ПС (прежде всего, программ) на компьютерных носителях данных (например, дисках) и предоставлению в распоряжение разработчика ПС **специальных ПС**или включенных в состав компьютера **специальных устройств**, созданных для какой-либо обработки таких документов.

  В качестве такого специального ПС можно указать **компилятор** с какого-либо языка программирования.

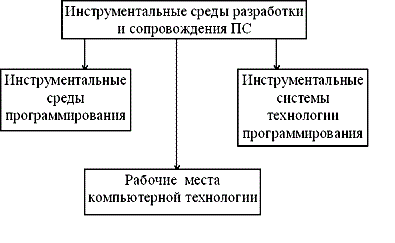
Специализированный набор инструментов, который используется при разработке и для изучения структуры построения кода.

Различают **три основных класса инструментальных сред разработки и сопровождения ПС**среды программирования, рабочие места компьютерной технологии, инструментальные системы технологии программирования.

Среда программирования предназначена в основном для поддержки процессов программирования (кодирования), тестирования и отладки ПС.

Рабочее место компьютерной технологии ориентировано на поддержку ранних этапов разработки ПС (спецификаций) и автоматической генерации программ по спецификациям.

Инструментальная система технологии программирования предназначена для поддержки всех процессов разработки и сопровождения в течение всего жизненного цикла ПС и ориентирована на коллективную разработку больших программных систем с длительным жизненным циклом.



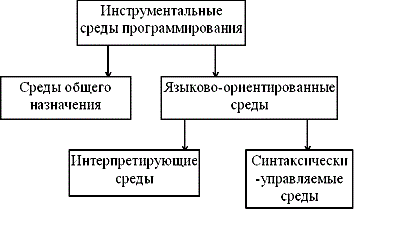
Основные классы инструментальных сред разработки и сопровождения ПС.

Инструментальные среды программирования содержат прежде всего текстовый редактор, позволяющий конструировать программы на заданном языке программирования, инструменты, позволяющие компилировать или интерпретировать программы на этом языке, а также тестировать и отлаживать полученные программы.

Кроме того, могут быть и другие инструменты, например, для статического или динамического анализа программ.

Взаимодействуют эти инструменты между собой через обычные файлы с помощью стандартных возможностей файловой системы.

Различают следующие **классы инструментальных сред программирования**: среды общего назначения, языково-ориентированные среды.



Классификация инструментальных сред программирования

**Понятие компьютерной технологии разработки программных средств и ее рабочие места.**

Имеются некоторые трудности в выработке строгого определения CASE-технологии (компьютерной технологии разработки ПС).

  CASE - это аббревиатура от английского Computer-Aided Software Engineering (Компьютерно-помогаемая Инженерия Программирования). Но без помощи (поддержки) компьютера ПС уже давно не разрабатываются (используется хотя бы компилятор).

  В действительности, в это понятие вкладывается более узкий (специальный) смысл, который постепенно размывается (как это всегда бывает, когда какое-либо понятие не имеет строгого определения).

Первоначально под CASE понималась инженерия ранних этапов разработки ПС (определение требований, разработка внешнего описания и архитектуры ПС) с использованием программной поддержки (программных инструментов).

Теперь под CASE может пониматься и инженерия всего жизненного цикла ПС (включая и его сопровождение), но только в том случае, когда программы частично или полностью генерируются по документам, полученным на указанных ранних этапах разработки. В этом случае CASE-технология стала принципиально отличаться от ручной (традиционной) технологии разработки ПС: изменилось не только содержание технологических процессов, но и сама их совокупность.

В настоящее время компьютерную технологию разработки ПС можно характеризовать

- Использованием программной поддержки для разработки графических требований и графических спецификаций ПС,

- автоматической генерации программ на каком-либо языке программирования или в машинном коде (частично или полностью),

- программной поддержки прототипирования.

Инструментальная система технологии программирования - это интегрированная совокупность программных и аппаратных инструментов, поддерживающая все процессы разработки и сопровождения больших ПС в течение всего его жизненного цикла в рамках определенной технологии.

Из этого определения вытекают следующие основные черты этого класса компьютерной поддержки: комплексность, ориентированность на коллективную разработку, технологическая определенность, интегрированность.

С учетом обсужденных свойств инструментальных систем технологии программирования можно выделить три их основные компоненты: база данных разработки (репозиторий), инструментарий, интерфейсы.

**Унифицированный язык моделирования UML**

* Большинство существующих методов объектно-ориентированного анализа и проектирования (ООАП) включают как язык моделирования, так и описание процесса моделирования. **Язык моделирования** – это нотация (в основном графическая), которая используется методом для описания проектов.
* **Нотация** представляет собой совокупность графических объектов, которые используются в моделях; она является синтаксисом языка моделирования. Например, нотация диаграммы классов определяет, каким образом представляются такие элементы и понятия, как класс, ассоциация и множественность.
* **Процесс** – это описание шагов, которые необходимо выполнить при разработке проекта.
* **Унифицированный язык моделирования UML** – это преемник того поколения методов ООАП, которые появились в конце 80-х и начале 90-х гг.

UML выделяют следующие типы диаграмм:

* **диаграммы вариантов использования** (usecase diagrams) – для моделирования бизнес-процессов организации (требований к системе);
* **диаграммы классов** (class diagrams) – для моделирования статической структуры классов системы и связей между ними. На таких диаграммах показывают классы, интерфейсы, объекты и кооперации, а также их отношения. При моделировании объектно-ориентированных систем этот тип диаграмм используют чаще всего. Диаграммы классов соответствуют статическому виду системы с точки зрения проектирования;
* **диаграммы поведения системы** (behavior diagrams);
* **диаграммы взаимодействия** (interaction diagrams) – для моделирования процесса обмена сообщениями между объектами.
* **диаграммы состояний** (statechart diagrams) – для моделирования поведения объектов системы при переходе из одного состояния в другое.