Перспективыне технологии ООП.

Парадигма ООП впервые была представлена в языке Simula67 в 60ых. С тех пор прошло много времени, а эта концепция сильно укрепилась и является, наверное, самым широко распространенным методом создания программных средств. Популярные языки постоянно развиваются, приобретают новые возможности. Большинство этих языков являются мультипарадигменными, их развитие чаще всего проявляется в расширении этой мультипарадигменности С ростом популярности приложений с графическими пользовательскими интерфейсами в 90ых, возрасла необходимость в поддержке событийно-ориентированной методологии программирования.

Языки приобрели новые инструменты, дающие возможность создавать события, и реагировать на них: делегаты, события.

Будучи чисто академической функциональная парадигма, сейчас же получила достаточно широкое распространение и в коммерческих продуктах. Как следствие появились такие языки как Scala, работающие в JVM и F# в .net часто используются для разработки высоконагруженных серверных решений. Другие же языки, например python, c#, начиная с версии 3.0 и Java 8 добавили несколько концепций, относящихся скорее к функцианальным языкам: лямдба и анонимные методы, map/reduce. Анонимные классы в Java, например, могу быть использованы как замыкания/clojures. Язык D так вообще поддерживает «чистые функции», аналогичные чисто функциональным языкам Haskel или OCaml.

Также в последнее время становится все больше объектно ориентированных языков с динамической типизацией, таких как Ruby, Python, Groovy, Dart. В C# с версии 3 также появились dynamic типы, позволяющие конструировать объекты во время выполнения программы.

Так же не стоит забывать про развитие стандартных библиотек популярных языков, например, с помощью нескольких библиотек с++ можно легко использовать приемы функционального программирования . Бьерн Страуструп, создатель языка С++, утверждает, что это самый перспективный путь для развития его языка.

Наиболее значимыми технологиями, разработанными в последнее время являются Java и .net.

Программы на java компилируются в байткод, который запускается на любой системе, под которую написана виртуальная Java-машина. Программа может быть написана на нескольких языках, которые компилируются в байткод, например на Scala,JRuby, Jython и т.ж .Net использует IL (Intermidiet Language) в качестве посредника между высокоуровневым языком в системе типов clr и машинными инструкциями. Все языки, базирующиеся на системе типов .net могут взаимодействовать друг с другом без каких либо сложностей. Класс написанный на IronRuby, затем, может быть использован в С# или vb.net. Существует аналог .net под названием Mono, который стремительными темпами развивается при поддержки Linux-сообщества. Программы с mono runtime запускаются на огромном количестве платформ, и предоставляют доступ к множеству библиотек, написанных для .net.