**1-2 ОАП Лк1 Развитие языков программирования**

**Первые языки программирования**

В пятидесятые годы двадцатого века с появлением компьютеров на электронных лампах началось бурное развитие языков программирования. Компьютеры, стоившие в то время значительно дороже, чем разработка любой программы, требовали высокоэффективного кода. Такой код разрабатывался вручную на языке *Ассемблер*. В середине 50-х годов под руководством Джона Бэкуса для фирмы *IBM* был разработан алгоритмический *язык программирования* *FORTRAN*. Несмотря на то, что уже существовали разработки языков, выполняющие преобразование арифметических выражений в машинный код, создание языка *FORTRAN* (FORmula TRANslator), предоставляющего возможность записи алгоритма вычислений с использованием условных операторов и операторов ввода/вывода, стало точкой отсчета эры алгоритмических языков программирования.

К языку *FORTRAN* предъявлялись требования cоздания высокоэффективного кода. Поэтому многие конструкции языка первоначально разрабатывались с учетом архитектуры *IBM* 407. Успех разработки этого языка привел к тому, что производители других вычислительных систем стали создавать свои версии трансляторов. С целью некоторой возможной на тот момент унификации языка язык *FORTRAN* IV, разработанный в 1966 году, стал первым стандартом, именуемым *FORTRAN* 66.

Как *альтернатива* языку *FORTRAN*, первоначально ориентированному на архитектуру *IBM*, под руководством Питера Наура в конце 50-х годов был разработан язык *ALGOL* (*ALGOrithmic Language*). Основной целью, преследуемой разработчиками этого языка, была независимость от конкретной архитектуры вычислительной системы. Кроме того, создатели языка *ALGOL* стремились разработать язык, удобный для описания алгоритмов и применяющий систему обозначений, близкую к той, что принята в математике.

Языки *FORTRAN* и *ALGOL* были первыми языками, ориентированными на *программирование* вычислений.

Язык PL 1, первые версии которого появились в начале 60-х годов, был первоначально ориентирован на *IBM* 360 и расширял возможности языка *FORTRAN* некоторыми средствами языка *COBOL*, разработанного в эти же годы. Несмотря на определенную популярность языка *PL/I* у программистов, работавших на компьютерах *IBM* и машинах серии ЕС, в настоящее время он представляет чисто теоретический интерес.

В конце 60-х годов под руководством Найарда и Дала был разработан язык Simula-67, использующий концепцию пользовательских типов данных. Фактически это первый язык, применяющий понятие классов.

В середине 70-х годов Вирт предложил язык *Pascal*, который сразу стал широко использоваться. В это же время по инициативе Министерства обороны США началась работа по созданию языка высокого уровня, получившего название *Ada* – в честь Ады Лавлейс, программистки и дочери лорда Байрона. Создание языка началось с определения требований и выработки спецификаций. Над проектом работали четыре независимые группы, но все они использовали как основу язык *Pascal*. В начале 80-х годов был разработан первый промышленный *компилятор* языка *Ada*.

Универсальный *язык программирования* С был разработан в середине 70-х годов Денисом Ритчи и Кеном Томпсоном. Этот язык стал популярным языком системного программирования и в свое время использовался для написания ядра операционной системы *UNIX*. *Стандарт языка* С начал разрабатываться рабочей группой института стандартов *ANSI* в 1982 году. Международный *стандарт языка* С принят в 1990 году. Язык С лег в основу разработки языков программирования C++ и *Java*.

Наряду с алгоритмическими языками параллельно развивались и языки, предназначаемые для обработки деловой информации, а также языки искусственного интеллекта. К первым относится язык *COBOL* (COmmon Business Oriented *Language*), а ко вторым – языки *LISP* (*LISt Processing*) и Prolog. Язык *LISP*, разработанный в 60-х годах под руководством Дж. Маккарти, был первым функциональным языком обработки списков, который нашел широкое применение в теории игр.

С появлением персональных компьютеров языки стали составными частями интегрированных сред разработки. Появились языки, применяемые в различных офисных программах, например *VBA* (*Visual Basic* for *Application*).

В 90-х годах с распространением сети *Интернет* расширяется возможность распределенной обработки данных, что отражается и на развитии языков программирования. Появляются языки, ориентированные на создание серверных приложений, такие как *Java*, Perl и *PHP*, языки описания документов – *HTML* и *XML*. Традиционные языки программирования С++ и *Pascal* также претерпевают изменения: под языком программирования начинает пониматься не только функциональность самого языка, а также библиотеки классов, предоставляемые средой программирования. Акцент со спецификации самих языков программирования переносится на стандартизацию механизмов взаимодействия распределенных приложений. Появляются новые технологии – *COM* и *CORBA*, специфицирующие взаимодействие распределенных объектов.

**Области применения языков программирования**

В настоящее время языки программирования применяются в самых различных областях человеческой деятельности, таких как:

* научные вычисления (языки C++, FORTRAN, Java);
* системное программирование (языки C++, Java);
* обработка информации (языки C++, COBOL, Java);
* искусственный интеллект (LISP, Prolog);
* издательская деятельность (Postscript, TeX);
* удаленная обработка информации (Perl, PHP, Java, C++);
* описание документов (HTML, XML).

С течением времени одни языки развивались, приобретали новые черты и остались востребованы, другие утратили свою актуальность и сегодня представляют в лучшем случае чисто теоретический интерес. В значительной степени это связано с такими факторами, как:

* наличие среды программирования, поддерживающей разработку приложений на конкретном языке программирования;
* *удобство сопровождения* и тестирования программ;
* стоимость разработки с применением конкретного языка программирования;
* четкость и ортогональность конструкций языка;
* применение объектно-ориентированного подхода.

Парадигмы программирования

Синтаксис языка описывает систему правил написания различных языковых конструкций, а семантика языка программирования определяет смысл этих конструкций. Синтаксис языка программирования может быть описан с помощью НБФ-нотаций.

Семантика языка взаимосвязана с используемой вычислительной моделью. В настоящее время языки программирования в зависимости от применяемой вычислительной модели делятся на четыре основные группы:

Процедурные языки, которые представляют собой последовательность выполняемых операторов. Если рассматривать состояние ПК как состояние ячеек памяти, то процедурный язык – это последовательность операторов, изменяющих значение одной или нескольких ячеек. К процедурным языкам относятся FORTRAN, C, Ada, Pascal, Smalltalk и некоторые другие. Процедурные языки иногда также называются императивными языками. Код программы на процедурном языке может быть записан следующим образом:

оperator1; operator2; operator3;

Аппликативные языки, в основу которых положен функциональный подход. Язык рассматривается с точки зрения нахождения функции, необходимой для перевода памяти ПК из одного состояния в другое. Программа представляет собой набор функций, применяемых к начальным данным, позволяющий получить требуемый результат. К аппликативным языкам относится язык LISP. Код программы на аппликативном языке может быть записан следующим образом:

function1(function2( function3(beginning\_date)));

**Языки системы правил,** называемые также языками логического программирования, основываются на определении набора правил, при выполнении которых возможно выполнение определенных действий. Правила могут задаваться в виде утверждений и в виде таблиц решений. К языкам логического программирования относится язык Prolog.

Код программы на языке системы правил может быть записан следующим образом:

if condition1 then operator1; if condition2 then operator2; if condition3 then operator3;

**Объектно-ориентированные языки**, основанные на построении объектов как набора данных и операций над ними. Объектно-ориентированные языки объединяют и расширяют возможности, присущие процедурным и аппликативным языкам. К объектно-ориентированным языкам относятся C++, Object Pascal, Java.

В настоящий момент наибольшее распространение получили языки, основанные на объектно-ориентированной модели. Они, реализуя процедурную модель построения языка, поддерживают аппликативность конструкций, позволяя представлять блок-схему выполнения структурированной программы как некоторый набор аппликативных функций.

**Стандартизация языков программирования**

Концепция языка программирования неотрывно связана с его реализацией. Для того чтобы компиляция одной и той же программы различными компиляторами всегда давала одинаковый результат, разрабатываются стандарты языков программирования. Существует ряд организаций, целенаправленно занимающихся вопросами стандартизации. Это Американский национальный институт стандартов ANSI (American National Standards Institute), Институт инженеров по электротехнике и электронике IEEE (Institute of Electrical and Electronic Engineers), Организация международных стандартов ISO (International Organization for Standardization).

Как правило, при создании языка выпускается частный стандарт, определяемый разработчиками языка. Если язык получает широкое распространение, то со временем появляются различные версии компиляторов, которые не точно следуют частному стандарту. В большинстве случаев идет расширение зафиксированных первоначально возможностей языка. Для приведения наиболее популярных реализаций языка в соответствие друг с другом разрабатывается согласительный стандарт. Очень важным фактором стандартизации языка программирования является своевременность появления стандарта – до широкого распространения языка и создания множества несовместимых реализаций. В процессе развития языка могут появляться новые стандарты, отражающие современные нововведения. Так, язык FORTRAN первоначально был стандартизирован в 1966 году. В результате был издан стандарт FORTRAN 66. Далее этот стандарт несколько раз пересматривался (в 1977 году был выпущен FORTRAN 77, затем появился и FORTRAN 90).

Язык Java, ставший в последнее время весьма распространенным, постепенно был значительно расширен и модифицирован: новая спецификация получила название Java 2.

В процессе развития языка некоторые его конструкции и функции устаревают. Однако с целью обратной совместимости новые версии должны поддерживать и все устаревающие возможности. Это ведет к "разбуханию" компиляторов. В последнее время в реализациях введено понятие не рекомендуемой и устаревшей возможности. В первом случае следующий стандарт еще будет поддерживать не рекомендуемую возможность, но может перевести ее в категорию устаревшей. Во втором случае стандарт может исключить поддержку возможности, объявленной ранее как устаревшая. Введение не рекомендуемых и устаревших возможностей предоставляет разработчикам временной интервал, в течение которого они могут модифицировать код в соответствии с новыми требованиями стандарта.

**Среда проектирования**

С развитием языков программирования совершенствовались и средства разработки программ – от режима командной строки до интегрированной среды проектирования. Такая среда предоставляет удобный графический интерфейс разработки и большой спектр сервисов, включающих управление версиями хранимых данных, утилиты просмотра и управления информацией, библиотеки классов, мастера создания шаблонов приложений и т.п.

Компилятор языка программирования выступает как составная часть среды проектирования. Сама программа наряду с конструкциями, предусмотренными стандартом, как правило, использует библиотечные функции и классы, предоставляемые средой проектирования. Так, интегрированная среда разработки VisualStudio.NET содержит библиотеку классов MFC (Microsoft Foundation Classes), значительно упрощающую процесс разработки приложений, использующих оконный интерфейс.

Интегрированная среда проектирования VisualStudio.NET позволяет создавать и компилировать приложения на языках C++, C#, Visual Basic и VisualJ. Для разработки приложений на языке С++ предназначается также среда CBuilder.

Для проектирования приложений на языке Object Pascal используется интегрированная среда проектирования Delphi.

Наиболее удобной средой разработки программ на языке Java является интегрированная среда проектирования JBuilder.