



ПРОГРАММИРОВАНИЕ

Авторы: В. В. Кулямин

ПРОГРАММИРОВАНИЕ, процесс создания компьютерных [программ](#) или [программного обеспечения](#), а также дисциплина, изучающая методы и приёмы создания и развития компьютерных программ (более точное назв. дисциплины – инженерия программного обеспечения или программная инженерия).

П. включает: анализ предметной области – выделение требований к программе и точную постановку решаемых задач; проектирование программы – выделение компонентов или модулей на основе отд. решаемых задач, определение способов взаимодействий между ними, определение алгоритмов работы и структур данных, используемых каждым модулем; кодирование – написание отд. модулей на определённых [языках программирования](#); выявление дефектов и ошибок при помощи тестирования (проверка программы на наборе заранее выбранных сценариев, позволяющем оценить её корректность) и др. методов [верификации](#), а также [отладку программы](#); развёртывание – размещение программы в её рабочем окружении, настройка и подготовка её к работе, обучение пользователей работе с программой; сопровождение – поддержка работоспособности, настройка под изменяющееся окружение, выявление дефектов и новых задач, внесение исправлений и изменений. Сопровождение играет особую роль, поскольку мн. программы используются в течение десятилетий и должны изменяться в соответствии с изменением решаемых задач или их масштабов, появлением новых устройств или необходимости во взаимодействии с др. программами. В рамках сопровождения много раз проводится перепроектирование, кодирование и тестирование, а также добавление новых компонентов, поэтому большая часть затрат на разработку программы падает на него. Дополнит. сложности для сопровождения и развития программ возникают из-за размеров и сложности совр. программных систем, достигающих десятков млн. строк кода, на разл. языках программирования. Такие масштабы программных систем стали

возможны за счёт использования подпрограмм и компонентов, позволяющих создавать иерархич. системы из большого количества небольших программ.

Перечисленные виды деятельности при П. обычно выполняются не в жёстко заданной последовательности, а по мере необходимости; напр., исправление ошибки, обнаруженной при тестировании, может потребовать дополнит. анализа требований, уточнения задач и выбора др. алгоритма работы содержащего ошибку модуля.

Процессы разработки программного обеспечения, предписывающие определённые правила создания программ, с целью упрощения планирования работ могут налагать разл. ограничения на возможные последовательности выполнения таких действий.

В зависимости от поставленных целей, размерности задачи, методов решения различают [параллельное программирование](#), [распределённое программирование](#) и др.

Слово «П.» используется также в некоторых устоявшихся словосочетаниях, напр. [динамическое программирование](#), [линейное программирование](#), [математическое программирование](#), где оно обычно является синонимом слова «планирование». Языки программирования поддерживают разл. стили П. ([парадигмы программирования](#)). В искусство П. входит выбор языка программирования, наиболее полно подходящего для решения поставленной задачи.

Большинство методов и технич. приёмов П. не универсальны, применимы лишь для специфич. видов программ (прикладные, системные, встроенные). Однако можно выделить как базовые следующие три принципа П.: модульность – существенно разл. задачи должны решаться разными программными компонентами, взаимодействующими друг с другом через чётко определённые интерфейсы, и не зависеть от внутр. алгоритмов и структур данных друг друга (см. в ст. [Модульное программирование](#)); использование абстракций – решение любой задачи необходимо оформлять в терминах набора сущностей, достаточных для описания всех существенных элементов задачи и не содержащих лишней, несущественной информации; многократное использование кода – каждый отд. элемент знания о задаче или её решения должен быть описан однократно, следует избегать дублирования информации и описаний одних и тех же знаний и/или решений в нескольких разл. местах в коде программы, поскольку при изменении требований

исправлять такую программу гораздо сложнее.

Первым программистом, написавшим в 1843 программу (вычисление чисел Бернулли) для вычислит. устройства (аналитической машины Ч. [Бэббиджа](#)), считается гр. А. [Лавлейс](#). П. на первых компьютерах осуществлялось путём установки переключателей в нужные положения; программ как таковых ещё не существовало. Первая ЭВМ с хранимой в памяти программой (реализация т. н. принципов фон Неймана) – EDSAC (1949, см. в ст. [Вычислительная машина](#)). Развитие П. (в нач. 1950-х гг.) связано с переходом от написания программ на языках машинных инструкций к более удобному для восприятия человека языком [ассемблер](#), а затем – к языкам высокого уровня, независимым от архитектуры компьютера, первыми из которых были фортран (fortran, 1954–57) и лисп (LISP, 1958, от LISt Processing language – язык обработки списков). Некоторые идеи, реализованные в таких языках, сформулированы А. А. [Ляпуновым](#) в его операторном методе программирования (1953).

Как дисциплина П. изучает принципы построения и функционирования программ, используя методы и технич. приёмы, а также способы организации как крупных программных систем (выбор архитектуры, выделение компонентов и организация эффективного взаимодействия между ними), так и небольших элементов программ (выбор алгоритмов работы, организация кода отд. компонента, использование особенностей языков программирования). В англоязычной лит-ре вместо термина «П.» используется «software engineering» – «программная инженерия» (введён в 1968 Ф. Л. Бауэром, США). В СССР в 1970-х гг. А. П. [Ершовым](#) термин переводился как «технология программирования». Важными вехами в развитии дисциплины стали выделение понятий программного модуля и его интерфейса канад. инж. Д. Парнасом в 1972, чёткое определение в нач. 1990-х гг. понятия архитектуры программного обеспечения и постепенно расширяющееся применение анализа программ.

Литература

Лит.: Software engineering / Ed. P. Naur, B. Randell. Brussels, 1969; Журавлев Ю. И. А. А. Ляпунов и становление кибернетики в нашей стране // Ляпунов А. А. Проблемы теоретической и прикладной кибернетики. М., 1980; Allen F. E. The history of language

processor technology in IBM // IBM Journal of Research and Development. 1981. Vol. 25. № 5; Zuse K. The computer, my life. В., 1993;Соммервилл И. Инженерия программного обеспечения. 6-е изд. М., 2002; Fuegi J., Francis J. Lovelace & Babbage and the creation of the 1843 «notes» // IEEE Annals of the History of Computing. 2003. Vol. 25. № 4; Липаев В. В. Программная инженерия. Методологические основы. М., 2006;Kruchten P., Obbink H., Stafford J. The past, present and future for software architecture // IEEE Software. 2006. Vol. 25. № 2.