

**PROBLEMS OF THE INITIAL PHASE IN THE PROGRAMMING  
LANGUAGES LEARNING**

Gladysheva M., Zaretsky M.

Magnitigorsk State Technical University, Russia

*The problems associated with the choice of languages used for primary programming language training of IT-specialists are considered in the article.*

**ПРОБЛЕМЫ НАЧАЛЬНОГО ЭТАПА ИЗУЧЕНИЯ ЯЗЫКОВ  
ПРОГРАММИРОВАНИЯ**

Гладышева М.М., Зарецкий М.В.

Магнитогорский государственный технический университет им.

Г.И. Носова, Магнитогорск, Россия

*Рассмотрены проблемы, связанные с выбором языков, применяемых для первоначального обучения программированию IT-специалистов.*

Современный IT-специалист должен обладать компетенциями, позволяющими успешно работать на любом этапе жизненного цикла программного продукта — анализе предметной области, проектировании программного продукта, его разработке (модернизации), внедрении, сопровождении. Кроме того, он должен непрерывно следить за состоянием рынка труда, самостоятельно определять перспективные сферы применения своих знаний и умений, совершенствовать их с учетом перспективы [1].

Ключевую роль в формировании всех перечисленных ранее компетенций играет освоение студентами основ программирования. Одной из важнейших проблем в построении курса программирования, на наш взгляд, является отделение принципиальных аспектов программирования от конкретных реализаций основных принципов в системах команд вычислительных машин, операционных системах, языках программирования.

В 50-60 (и в начале 70-х) годах прошедшего века считалось очевидным, что начинать систематическое изучение программирования следует с освоения технологии работы в машинных кодах. Чтобы абстрагироваться от непринципиальных

деталей авторы учебников чаще всего рассматривали систему команд гипотетической вычислительной машины и программирование в ее кодах [2, 3]. В публикациях, предназначенных для школьников, применялась та же схема изложения [4]. О массовом изучении программирования в школах не было и речи. Иногда в качестве объекта изучения использовались система команд и программирование в кодах реальной вычислительной машины [5]. После освоения программирования в машинных кодах студент переходил к изучению программирования на языках высокого уровня.

Такой подход был вполне логичен до начала массового внедрения ЭВМ третьего поколения [6, 7]. Сложность системы команд этих машин не позволяет считать применявшиеся ранее гипотетические модели адекватными реальности ввиду их архаичности. Не менее архаичной на фоне системы команд машин третьего поколения выглядит и изучаемая в [5] система команд ЭВМ типа М-20. Более того, для изучения систем команд и Ассемблера машин третьего поколения требовались определенные программистские навыки. Эти навыки можно было приобрести только в результате изучения языков программирования высокого уровня.

Таким образом, в результате усложнения архитектуры ЭВМ появилась схема изучения программирования, предполагающая освоение на начальном этапе одного из языков высокого уровня. Успех учебного процесса в значительной степени зависит от выбора такого языка программирования.

Сделаем важное замечание. Современный первокурсник-программист знакомится с языком программирования в школе. Согласно проведенному нами опросу более 90% процентов наших первокурсников изучали в школе язык Паскаль. Остальные изучали самые разные языки — VBA, C++, PHP, Python.

К сожалению, немногие школьники изучают язык Паскаль по таким содержательным пособиям, как [8]. Очень часто школьники овладевают некоторым подмножеством языка, не включающим такие интересные возможности языка, как записи, классы и объекты. Более того, часто в программе не выделяются даже функции и процедуры. Использование мощного средства

разработки Delphi обычно сводится к хаотическому разбрасыванию по пользовательской форме элементов управления с непродуманной функциональностью. Не лучше чаще всего обстоят дела и в тех случаях, когда школьники изучают не Паскаль, а другие языки программирования.

Таким образом, перед преподавателями часто стоит задача не только обучения, но и переучивания, что существенно затрудняет освоение современной технологии программирования.

Существует несколько подходов к обучению основам программирования будущих IT-специалистов [9]. У каждого из шести подходов есть свои положительные и отрицательные стороны. Заметим, что алгоритмический подход, на наш взгляд, может рассматриваться в качестве развития подхода, который применялся в [2, 3, 5]. Объектный подход на начальном этапе возможен только в сочетании с императивным или функциональным подходом — без их использования невозможно придать объектам функциональность.

Немаловажным является вопрос о допустимой степени абстрагирования при написании программы. При написании программы в кодах машины программист сам распределяет память, организует взаимодействие всех процессов. Уже при работе с Ассемблером могут применяться макросредства. При программировании на языке высокого уровня программист может не вникать в детали организации взаимодействия с внешними устройствами. Еще больше возможностей появляется при работе с современной интегрированной средой разработки. При организации учебного процесса необходимо определить, какими «готовыми» средствами студент может пользоваться, а какими — нет. Мы считаем, что в данном случае отсутствует универсальный ответ. Все зависит от того, какие компетенции мы хотим сформировать. В частности, на начальном этапе знакомства с объектной парадигмой программирования уместным является использование SCRATCH-технологий [10].

Дискуссионным является вопрос о количестве языков программирования, изучаемых на начальном этапе обучения [11]. Мы считаем, что таких языков должно быть не менее двух. Это поможет студенту понять разнообразие языков программирования,

научит сопоставлять их, а в дальнейшем рационально выбирать язык программирования для решения конкретной задачи.

Конкретный выбор языков программирования, применяемых в на начальном этапе обучения, обусловлен многими факторами. К важнейшим из них можно отнести:

- пригодность для первоначального обучения;
- наличие многочисленных областей применения языка;
- наличие свободно распространяемой «студенческой» версии, оснащенной средой разработки;
- наличие доступной учебной литературы.

Безусловно, оценка пригодности языка программирования для первоначального обучения не может не быть субъективной, но, тем не менее, очевидно, что не все языки в равной степени пригодны для обучения начинающих.

Мы считаем, что для первоначального обучения пригодны Паскаль (версии Pascal ABC, Turbo Delphi), C++ (версия Turbo C++), Prolog, Python.

Одной из важнейших компетенций программиста является, на наш взгляд, компетенция самообучения. В течение профессиональной жизни программисту придется не раз и не два осваивать принципиально новые для себя парадигмы. Поэтому мы считаем несколько утопичными следующие требования к языку программирования, применяемому в учебном процессе: «Для навчання потрібна мова, яка була б не лише сучасна, зручна, зрозуміла та ефективна, а й не потребувала від людини подальшого переучування в процесі професійного розвитку» [12]. Вряд ли возможно создание языка программирования, концепции которого не подлежат пересмотру.

Подводя итоги, сформулируем два основных тезиса:

на начальном этапе профессиональной подготовки будущий IT-специалист должен освоить программирование на нескольких языках высокого уровня;

процесс освоения этих языков должен готовить его к изучению организации ЭВМ и Ассемблера, а также формировать компетенции самостоятельной учебы и переучивания.

Литература

1. Гладышева М.М. Образование IT-специалистов, конкурентоспособных на глобализованном рынке труда /

- М.М. Гладышева, М.В. Зарецкий // Сборник трудов Пятой Международной конференции «Новые информационные технологии в образовании для всех: непрерывное образование». – К.: Международный научно-учебный центр информационных технологий и систем НАН и МОН Украины, 2010. – С. 369 – 373.
2. Гнеденко Б.В. Элементы программирования / Б.В. Гнеденко, В.С. Королюк, Е.Л. Ющенко – М.: Физматгиз, 1961. – 348 с.
  3. Жоголев Е.А. Курс программирования / Е.А. Жоголев, Н.П. Трифонов – М.: Наука, 1967. – 408 с.
  4. Лященко М.Я. Розв'язування задач на електронних цифрових машинах / М.Я. Лященко // Збірник науково-популярних статей «У світі математики» - К.: Радянська школа, 1968. – С. 50-74.
  5. Лавров С.С. Введение в программирование / С.С. Лавров – М.: Наука, 1977. – 368 с.
  6. Королев Л.Н. Структуры ЭВМ и их математическое обеспечение / Л.Н. Королев – М.: Наука, 1974. – 254 с.
  7. Стэбли Д. Логическое программирование в системе /360 / Д. Стэбли – М.: Мир, 1979. – 752 с.
  8. Павловская Т.А. Паскаль. Программирование на языке высокого уровня. 2 – е изд. / Т.А Павловская. – СПб.: Питер, 2010.- 464 с.
  9. Сейдаметова З.С. Моделі навчання основ програмування на молодших курсах комп'ютерних спеціальностей університетів / З.С. Сейдаметова, Л.М. Меджитова // Науковий часопис НПУ ім. М.П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання. Зб. Наукових проць. / Редрада. – К.: НПУ ім. М.П. Драгоманова, 2009, № 7 (14). – С. 103 – 107.
  10. SCRATCH [Електронний ресурс]. Режим доступа: <http://www.scratch.mit.edu>
  11. Сейдаметова З.С. Мови програмування в навчанні майбутніх програмістів / З.С. Сейдаметова, Л.О. Манжос // Науковий часопис НПУ ім. М.П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання. Зб. Наукових проць./ Редрада. – К.: НПУ ім. М.П. Драгоманова, 2010, № 8 (15). – С. 35 – 41
  12. Шевчук П.Г. Використання платформи Microsoft.NET для навчання програмуванню в середніх загальноосвітніх навчальних закладах/ П.Г. Шевчук // Збірник праць Четвертої Міжнародної

конференції «Нові інформаційні технології в освіті для всіх: інноваційні методи та моделі». – К.: Міжнародний науково-навчальний центр інформаційних технологій та систем НАН и МОН України, 2009. – С. 278 – 287.