

**С.Н. Ширёва**

**НЕКОТОРЫЕ АСПЕКТЫ ПРЕПОДАВАНИЯ ПРОГРАММИРОВАНИЯ ДЛЯ  
БАКАЛАВРОВ**

*Ширёва Светлана Николаевна*

*shireva@mail.ru*

*ФГАОУ «Российский государственный профессионально-педагогический  
университет», Россия, г. Екатеринбург*

**SOME ASPECTS OF TEACHING PROGRAMMING FOR BACHELORS**

*Shireva Svetlana Nikolaevna*

*The Russian State Vocational Pedagogical University, Russia, Yekaterinburg*

***Аннотация.** В статье рассматриваются вопросы преподавания программирования в высшей школе. Подготовка специалистов, которые смогут принимать участие в разработке программного обеспечения, включает широкий спектр разнообразных тем. Но количество часов в учебном плане, выделенных для дисциплин, посвященных программированию, ограничен. Поэтому возникает необходимость выделить главные направления подготовки специалистов.*

***Abstract.** The article considers the issues of teaching programming in high school. Training of specialists who will be able to participate in the development of the software includes a wide variety of topics. But the number of hours in the curriculum allocated to the subjects on programming, limited. Therefore, there is a need to highlight the main areas of training.*

***Ключевые слова:** императивные языки программирования; алгоритмы; концепция связывания.*

***Keywords:** imperative programming languages; algorithms; the concept of linking.*

Вопросы алгоритмизации и программирования являются обязательными при подготовке бакалавров. Эта тема присутствует при изучении дисциплины Информатика и входит в перечень разделов, по которым происходит тестирование в рамках дисциплины Информатика. Между тем, в том объеме часов, которые выделяются на изучение дисциплины Информатика, сложно хорошо познакомиться с этими разделами. В каждое учебное заведение по-своему решает эту проблему. Некоторые вузы программированию уделяют большое внимание на дисциплине Информатика, и, по сути, студенты все время посвящают изучению программирования. В других вузах изучение алгоритмизации и программирования происходит формально, для этого выделяется 4 часа, только чтобы познакомить студентов с терминологией этих разделов. Данная статья не рассматривает решение этой проблемы, но о ней следует упомянуть.

Кроме изучения программирования в рамках дисциплины Информатика, некоторым не компьютерным специальностям можно рекомендовать изучение программирования. Причем, чем дальше, тем больше прослеживается тенденция необходимости изучения офисного

программирования, (языка VBA). Все больше сотрудников банков и других компаний направляются на курсы по изучению языка VBA [1].

Если рассматривать преподавание программирования для компьютерных специальностей, то тут тоже существует много проблем.

Современное программирование включает в себя большое количество разделов, но далеко не все эти темы понадобятся в дальнейшем студентам. Поэтому надо сразу же определиться с набором наиболее важных разделов для изучения.

В идеале, порядок изучения программирования следует историческому развитию языков программирования. Сначала знакомство с основами программирования на языке низкого уровня ассемблер. На этом этапе изучается работа с регистрами, ячейками памяти. После этого у студентов складывается представление, что такое адреса, сегменты памяти, стек.

Следующий этап – это процедурные языки программирования. У студентов к этому моменту уже сложилось представление о механизмах работы с памятью. Им проще представить механизм передачи параметров. У студентов нет проблем при работе с указателями. На этом этапе основное внимание следует посвятить изучению различных алгоритмов.

После изучения алгоритмов и программирования на языках высокого уровня следует перейти к объектно-ориентированному программированию. Язык программирования, в котором наиболее развиты механизмы объектно-ориентированного программирования является язык C#. Но при изучении объектно-ориентированного программирования по-прежнему надо представлять механизм работы на уровне ячеек. В частности, без представления работы ссылочного механизма невозможно понять полиморфизм.

Но все это в идеале. Реально для изучения программирования не выделяется столько часов, чтобы рассмотреть все эти языки.

Еще одна проблема при преподавании программирования: программирование не возможно без алгоритмов. Первая задача, с которой мы сталкиваемся при решении задачи – грамотно разработать алгоритм. В какой-то момент, когда рассмотрены все операторы языка, на первое место выступает умение разрабатывать алгоритмы. Разработке алгоритмов тоже можно научить. Можно рассматривать достаточно сложные, олимпиадные алгоритмы, но насколько это оправдано? Сколько времени можно выделить на обучение разработке алгоритмов? В этой ситуации надо помнить, что не все студенты компьютерной специальности собираются стать разработчиками, многие из них будут системными администраторами, web-дизайнерами и т.д. Для таких студентов сложно разобраться с алгоритмами перебора, сложными рекурсивными алгоритмами, поэтому, изучение таких алгоритмов следует вынести на факультатив, а из сложных алгоритмов оставить только алгоритмы сортировки (хотя бы сортировку методом пузырька).

Возникает вопрос, на кого же тогда ориентироваться при преподавании программирования: на тех, кто собирается в дальнейшем заниматься разработкой программ, или на тех, кто будет работать в другой области. Это проблему можно решить только через индивидуальные задания. Кому-то надо продемонстрировать минимальный уровень, чтобы получить зачет, а другие студенты, выполняя более сложные индивидуальные задания, будут изучать дополнительный материал. Такие студенты будут разрабатывать сложные алгоритмы,

а студенты, которые не собираются в дальнейшем стать разработчиками, будут решать задачи со стандартными алгоритмами.

Какие же вопросы надо рассматривать, чтобы подготовить специалистов разработчиков программного обеспечения. Ответы на эти вопросы можно получить из соответствующей литературы, посвященной собеседованию при поступлении на работу в IT-компанию. Это работа со связными списками, деревьями, массивами и строками. Это вопросы сортировки и эффективности алгоритмов и, конечно, объектно-ориентированное программирование[2]. Большинство из этих тем входит в курс программирования для студентов бакалавров. Некоторые из этих тем студентом придется осваивать самостоятельно или в рамках факультатива.

Последняя проблема, которая будет рассмотрена в этой статье – это язык программирования, который надо изучать. Если студент собирается работать программистом, неизвестно, на каком языке ему придется программировать. Многие крупные фирмы имеют свои корпоративные информационные системы, со своим собственным языком программирования. Таким образом, при изучении программирования следует рассматривать общие вопросы программирования, иллюстрируя их примерами на конкретных языках, которые студенты уже изучали (например, язык VBA), или изучают на практике в данный момент. Среди этих вопросов механизмы статического и динамического связывания, вопросы представления данных в памяти компьютера, указатели [3]. Работу этих механизмов можно хорошо визуализировать, создав приложение, моделирующее работу этих механизмов.

На практических занятиях по программированию надо использовать свободно распространяемый язык. В данный момент к таким языкам относятся Visual Basic, C++ и C# среды Visual Studio. При изучении процедурного программирования лучше остановить выбор на языке C++, потому что, как это сказано выше, необходимо рассмотреть работу с указателями, а в языке C# указатели не используются. При изучении объектно-ориентированного программирования лучше выбрать современный язык C# с его богатым выбором средств объектно-ориентированного программирования.

### ***Список литературы***

1. *Ширёва, С.Н.* Роль языка Visual Basic for Application при изучении программирования / С.Н. Ширёва // Новые информационные технологии в образовании : материалы Междунар. науч.-практ.конф., 11-14 марта 2014 г., г. Екатеринбург. / Рос. гос. проф.-пед. ун-т. — Екатеринбург, 2014. — С. 160–162.
2. *Монган, Дж.* Работа мечты для программиста. Тестовые задачи и вопросы при собеседовании в ведущих IT-компаниях [Текст] / Дж. Монган, Н. Киндлер, Э. Гижере – СПб.: Питер, 2014. – 386 с.
3. *Собеста У. Роберт* Основные концепции языков программирования [Текст] / Роберт У. Собеста – 5-е изд.; Пер. с англ. – М.: Издательский дом “Вильямс”, 2001. – 672 с.