UDK: 811.93:004.43:371.3

СРЕДСТВА И ФОРМЫ ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ В ВЫСШЕЙ ШКОЛЕ БУДУЩИХ УЧИТЕЛЕЙ ИНФОРМАТИКИ

Джураева Дилафруз Рауповна

Докторант Навайиского государственного педагогического института

Республика Узбекистан

***Аннотация**: в данной работе рассматриваются вопросы совершенствования методической системы обучения программированию будущих учителей информатики.*

***Ключевые слова:** алгоритм, программа, алгоритмизация, программирование, методическая* система обучения программированию, языки программирования.

MEANS AND FORMS OF ORGANIZATION OF TEACHING PROGRAMMING IN HIGHER SCHOOL OF FUTURE TEACHERS OF INFORMATICS

Djuraeva Dilafruz Raupovna

Doctoral Student of the Navoi State Pedagogical Institute

Republic of Uzbekistan

***Abstract**: In This paper we discuss the issues of improving methodical system of training of future informatics teachers programming.*

***Key words:** algorithm, the program, algorithmization, programming, methodical system for teaching programming, programming languages.*

ВВЕДЕНИЕ.

Программирование можно рассматривать как искусство, науку, ремесло. Программирование - это искусство получения ответов от компьютера. Для этого в узком смысле нужно составить специальный код для технического устройства, а в широком - разработать программы на языках программирования, т. е. не просто составить код, а выполнить интеллектуальную работу по составлению высоко разумных программ для решения различных задач во всех сферах человеческой деятельности.

МЕТОДОЛОГИЯ.

Сами по себе языки программирования относятся к категории искусственных языков. Из-за своих внутренних, неизменяемых системных слов, они являются ограниченными [1,2].

К основным требованиям, применяемых для языков программирования, можно отнести:

Простота использования уже существующих символов, которые знакомы и понятны программистам.

Единства использования, действительно сейчас практический каждый язык программирования высокого уровня имеет свои схожести, не во всем конечно, но это так.

Гибкость языка программирования, заключается в том, что он может применяться для разных платформ и использоваться на разных ОС. Это позволяет подходить к гибкой разработке.

Модульность языка программирования. Обязательно необходимо разбивать большие и сложные программы на более маленькие, тем самым определяя разные модули и зависимости внутри программы.

Одним из основных разделов современных курсов информатики, преподаваемых в республиканских вузах, являются разделы, связанные с обучением программированию. А также, формирование у обучаемых

алгоритмического стиля мышления, подготовке к оперированию с важнейшими инструментальными системами и средствами [1,2,3,4].

В опубликованных работах ведущих ученых неоднократно подчеркивалась необходимость совершенствования методических систем обучения программированию будущих учителей информатики в связи с потребностью подготовки специалистов, владеющих процедурными, объектно-ориентированными, логическими и функциональными подходами к разработке алгоритмов и программ [5,6,10].

Таким образом, становятся актуальными вопросы изучения существующих подходов к организации обучения программированию в вузе и развития курса информатики с целью подготовки специалистов, владеющих всеми парадигмами программирования [7,8,9].

РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЯ.

Программирование занимает одну из важнейших частей информатики, так как в нем концентрируются инженерные вопросы реализации алгоритма при заданных пространственно-временных ограничениях, средствами конкретного языка программирования с учетом всего жизненного цикла программного продукта. Введение нескольких языков, а, тем более, парадигм программирования позволяет адаптировать полученные знания к быстро меняющейся обстановке в сфере новых информационных технологий, что, в свою очередь, позволяет на новом качественном уровне использовать информационные технологии в учебном процессе, предоставляет возможность реализовать требуемую модель подготовки студентов.

Традиционная методика обучения программированию заключается прежде всего в том, что обучаемые знакомятся сначала с теоретическими основами программирования, затем им предлагается написать программу, используя полученные теоретические знания по конкретному языку программирования (как правило, это задачи вычислительного типа). Эта

методика достаточно эффективна при обучении людей с солидной математической подготовкой либо ориентированных на то, чтобы стать профессиональными программистами.

Какие задачи необходимо решить, обучая будущих учителей информатики программированию? Еще совсем недавно все было предельно ясно — круг пользователей вычислительной машине практически на 100% состоял из программистов, а значит освоение «компьютерной грамотности» сводилось к изучению языка программирования. При этом наиболее характерным подходом было обучение по принципу «что знаем, то и преподаем», не заботясь о вопросе, «а зачем это нужно?». Результатом такого обучения является то, что большинство студентов забывает о программировании на следующий же день после сдачи рейтинги или тестов.

Сегодня очевидно, что программирование является в лучшем случае частью проблемы компьютерной грамоты, а методика его преподавания должна поменяться самым решительным образом. За последние несколько лет технология разработки программ претерпела настоящие инновационное изменения (визуальное программирование, событийная логика программы, компонентная технология, использование макросредств и пр.), что должно отразиться на методике обучения.

Существует два подхода к изучению языка программирования:

формальный и «программирование по образцу». Первый основан на формальном описании конструкций языка программирования (синтаксиса языка и его семантики) тем или иным способом (с помощью синтаксических диаграмм, метаязыка или формального словесного описания, в частности, семантики) и использовании при решении задач только изученных, понятных элементов языка.

При втором подходе обучаемым сначала выдаются готовые программы, рассказывается, что именно они делают, и предлагается

написать похожую программу или изменить имеющуюся, не объясняя до конца ряд «технических» или несущественных, для решения задачи деталей. При этом говорится, что точный смысл соответствующих конструкций вы узнаете позднее, а пока поступайте аналогичным образом. Второй подход дает возможность так называемого «быстрого старта», но создает опасность получить полуграмотных пользователей среды программирования, т.е. людей, которые используют в своей практике достаточно сложные конструкции, но не могут четко объяснить, почему в том или ином случае нужно применять именно их, и как они работают. В результате рано или поздно такие «программисты» сталкиваются с ошибками, исправить которые они просто не в состоянии - им не хватает знаний.

Основу методической системы обучения программированию составляет теоретический и практический материал курса «Программирование», обеспечивающий профессиональные знания в области информатики и компьютерной техники, которые необходимы студентам в их будущей деятельности. Существенных изменений требует методика обучения различным парадигмам программирования.

Под методом обучения мы понимаем упорядоченные способы взаимосвязанной деятельности преподавателя и студента, направленные на достижение поставленных целей обучения конкретной научной дисциплине. По способу передачи информации от преподавателя к студенту различают вербальные, наглядные и практические методы обучения.

При обучении программированию, мы используем вербальные (при изложении лекционного материала) и практические (выполнение лабораторных работ, практикумов, решение задач) методы, причем основной акцент делаем на практические методы, в процессе применения

которых студенты не только, получают новые знания, но и приобретают практические навыки.

Преподаватель при этом инструктирует, указывает цели работы, направляет и проверяет ход ее исполнения. В деятельности студентов преобладает практическая работа, в ходе которой особую роль играет самостоятельный мыслительный процесс, позволяющий осуществить поиск данных и парадигмы решения задачи.

По основным видам дидактических проблем, решаемых на занятии, можно выделить методы приобретения знаний, формирования умений, применения знаний, методы творческой деятельности и методы проверки знаний, умений и навыков. Отметим, что все перечисленные методы приемлемы для использования при обучении программированию.

Часто учебная деятельность представляет собой итерационный поступательный процесс (речь идет об итерационном методе обучения).

Рассматривая итерацию как пошаговое приближение к определенной цели, можно применить этот метод, как при изложении лекционного материала, так и в процессе выполнения лабораторных работ по программированию.

Специфика заданий, предназначенных для выполнения на лабораторном практикуме, вполне соответствует поступательному итерационному процессу, который выражается в построении ряда алгоритмов и программ решения задачи, причем каждый следующий алгоритм является уточнением или расширением предыдущего. Каждая лабораторная работа содержит следующие разделы:

краткий теоретический материал (основные сведения);

эксперименты с программами (в готовые программы нужно внести некоторые изменения и проанализировать полученный результат);

задания для самостоятельной работы.

Таким образом, построение итоговой программы представляет собой итерационный процесс, на каждом шаге которого происходят некоторые изменения, что и позволяет нам применить итерационный метод обучения.

Последовательность изложения лекционного материала зависит от порядка практических и лабораторных работ, поэтому лекционный курс целесообразно строить на основе итерационного метода. В случае практического применения подобной методики обучение реализуется не на основе постепенного изучения новых структур и операторов одной из возможных парадигм программирования, а с помощью поступательного итерационного процесса уточнения и расширения возможностей программной реализации моделируемой системы. Причем введение новых структур данных и возможностей языка программирования обосновывается с точки зрения их необходимости для решения новой задачи.

Используя итерационный метод обучения программированию при работе со студентами, мы используем в основном формальный подход. При этом некоторыми неформальными умениями студенты чаще всего уже обладают. Но и без хороших примеров (образцов) при обучении программированию не обойтись. Чем больше в группе студентов с несформированными неформальными умениями, тем больше примеров необходимо приводить при описании языка (иногда даже заменяя ими строгое определение). Необходимо добиваться того, чтобы в результате обсуждения примера все его детали оказались понятны студентам (обязательно нужно объяснить, как и почему это работает, в том числе опираясь на уже изученный формальный материал). В этом случае сильные студенты получат возможность понять все досконально и смогут использовать полученные знания в дальнейшем, а средние — приобретут конкретные навыки и оставят для себя возможность вернуться при необходимости к формальным определениям позже.

ВЫВОД.

Таким образом, делая выводы из реального опыта можно сказать, что исследование в плане организации процесса обучения программированию в условиях проектной деятельности показали положительный результат и тем самым приобретает перспективу исследования в данном направлении в плане фундаментализации методики обучения.

REFERENCES

1. M. Aripov, M.Muhammadiyev. Informatika, information texnologiyalar. Darslik.T.: TDYuI, 2005 y.

2. A.R. Azamatov, B.Boltayev. Algoritmlash va dasturlash asoslari. O‘quv qo‘llanma.T.:”Cholpon”. 2013 y.

3. Sattorov A. Informatika va axborot texnologiyalari. Darslik. T.:” O‘qituvchi”, 2011 y.

4. Sh. A.Nazirov, M.M.Musayev, A.Ne`matov, R.V.Qobulov. Delphi tilida dasturlash asoslari. Toshkent. 2007- yil.

5. P.Karimov, S.Irisqulov, A. Isabayev. Dasturlash. Toshkent. “O‘zbekiston”. 2003-yil.

6. П. Дарахвелидзе, Э. Марков. Программирование в Delphi7. Учебник. Санкт- Петербург, “БХВ-Петербург”, 2003 г.

7. Вигерс Карл. Разработка требований к программному обеспечению. /Пер, с англ. - М.: Издательско-торговый дом "Русская Редакция", 2004. -576с

8. Окулов, С.М. Основы программирования. – М.: ЮНИМЕДИАСТАЙЛ, 2002. – 424 с.: ил.

9. Иванова, Л.В. Язык программирования Pascal. Часть II. Лабораторный практикум. Книга 2. Уч.-метод. Пос. для студентов педагог. вузов. Елабуга, ЕГПУ, 2007, 136 с.

10. Роберт У. Себеста. Основные концепции языков программирования = Concepts of Programming Languages / Пер. с англ. – 5-е изд. – М.: Вильямс, 2001. – 672 с. – 5000 экз. ISBN: 5–8459–0192–8