

УДК 37.01

DOI 10.58649/1694-8033-2023-4(116)-18-21

БОЛОТБАЕВ Д.С., АЖЫМАНБЕТОВА Г.И.

КУУ атындагы Ж. Баласагын

БОЛОТБАЕВ Д.С., АЖЫМАНБЕТОВА Г.И.

КНУ им. Ж. Баласагына,

BOLOTBAEV D.S., AJYMANBETOVA G.I.

KNU named after J.Balasagyn,

ORCID: 0009-0009-6787-9404; SPIN-код: 9806-31902

ПРОГРАММАЛООГО ҮЙРӨТҮҮДӨ ОКУТУЛУП ЖАТКАН
ДИСЦИПЛИНАЛАРДЫН ЧӨЙРӨСҮН КЕҢЕЙТҮҮНҮН ЗАРЫЛДЫГЫ

**НЕОБХОДИМОСТЬ РАСШИРЕНИЯ КРУГА ИЗУЧАЕМЫХ ДИСЦИПЛИН
ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ**

THE NECESSITY OF BROADENING THE RANGE OF DISCIPLINES IN
PROGRAMMING EDUCATION

Кыскача мүнөздөмө: Макалада болондук окутуу системасына ИТ адистерин даярдоодогу маселелерге басым жасалып, аларды окутууда колдонулган окуу планын иштеп чыгуудагы кездешкен кыйынчылыктар талданды.

Аннотация: В статье рассмотрены вопросы, связанные с обучением ИТ-специалистов в условиях Болонской системы образования. В ней анализируются проблемы, возникающие при разработке учебных планов для обеспечения эффективного обучения.

Abstract: The article focuses on issues related to the education of IT specialists in the context of transitioning to the Bologna education system. It analyzes the challenges that arise in developing a curriculum to ensure their effective learning.

Негизги сөздөр: алгоритм; программисттердин окуу процессине дисциплиналарды интеграциялоо; билим берүүнү интеграциялоо; фундаменталдык билим берүү; ИТ адистери; окутуу; моделдөө; функционалдык программалоо; Питон; Фортран.

Ключевые слова: алгоритм; интеграция дисциплин в образовательный процесс программистов; интеграция образования; фундаментальное образование; Итспециалисты; обучение; моделирование; функциональное программирование; Питон; Фортран.

Keywords: algorithm; integration of disciplines into the learning process for programmers; integration of knowledge acquisition; fundamental education; IT specialists; education; modeling; functional programming; Python; Fortran.

«Чем опытнее программист, тем лучше он осознаёт
всю скудность своих знаний и навыков»

Продуктом общественной материальной и духовной деятельности людей являются знания. Под знанием мы понимаем совокупность сведений, полученных нами в процессе обучения, воспитания и опыта, приобретенного в процессе взросления. Знания позволяют нам познавать мир и совершенствоваться самим. Поэтому, каким путем мы приобретаем эти знания, является жизненно необходимой потребностью.

Этот процесс начинается с рождения ребенка, когда он получает воспитание от родителей. Затем следуют ясли, детский сад, где уже опытные воспитатели дают детям азы начального обучения. Далее следует школа, где в основном закладываются знания об окружающем нас мире, и формируется разносторонняя личность. Здесь молодые люди в большинстве своем определяют направление своей будущей деятельности, а именно выбор профессии.

Хорошо, если на этом этапе они понимают, что профессия программиста предполагает, я не говорю глубокие, об этом остается только мечтать, а хорошие знания в области естественных наук, а именно: математики, физики, информатики и других наук. Многие молодые люди полагают, что достаточно овладеть какими-либо языками программирования, чтобы делать в дальнейшем карьеру. Они не понимают, что на практике им придется столкнуться с решением проблем, где требуется моделировать какие-то процессы или явления, а затем их разрешения. А для этого необходимы знания многих фундаментальных дисциплин.

Не будем говорить о недостатках в системе среднего образования, хотя они оказывают влияние на подготовку специалистов высшего звена. Переход на контрактную систему обучения вынуждает вузы набирать студентов с достаточно невысокими баллами по ОРТ. Если максимальными для ОРТ являются 240 баллов, а проходным для поступления в вузы считаются 110 баллов, что в процентном соотношении составляет приблизительно 46%, то о каком качестве подготовки специалистов приходится говорить.

При подготовке специалистов мы должны опираться на те знания, которые они получили в школе. Исходя из этого, и строится стратегия их дальнейшего обучения.

В настоящее время, когда компьютеры получили широкое распространение повсеместно, возникает большая потребность в IT-специалистах. От уровня их подготовки зависит развитие экономики страны.

В основе навыков программиста лежат алгоритмический стиль общения и понятие алгоритма. Алгоритм легче выполнить, чем разработать и записать. Программисту необходимо выработать приемы умственных действий. Необходимо постоянно учиться в силу непрерывно меняющегося характера задачи. Способность к обучению является необходимым качеством программиста. Программист должен уметь варьировать решения и оценивать их.

Знание для программиста приобретается путем освоения различных смежных с математикой дисциплин. После перехода системы высшего образования Кыргызстана на Болонскую систему, была утрачена взаимосвязь с бывшей, советской системой образования. Если советская система образования делала упор на получение фундаментальных знаний в высшем образовании, то Болонская система предполагает более узкую специализацию, которая рассчитана на потребности подготовки конкретных специальностей и не

предполагает обширных знаний. При этом возникает необходимость в изучении дополнительных дисциплин, не входящих в общий (основной) или профессиональный цикл.

Удивительно, но в настоящее время такая дисциплина, как “Программирование” находится в стесненном положении. В чем это выражается, рассмотрим на обычном классическом примере решения уравнений вида $f(x)=0$, где в общем случае функция $f(x)$ нелинейна. Рассмотрим алгоритм решения таких уравнений.

Во-первых, надо определить, имеет ли данное уравнение решение?

Уже на этом этапе у студентов возникают проблемы. Они связаны с тем, что для этого надо владеть аппаратом математического анализа, алгебры, но этим дисциплинам либо мало выделяется времени, либо их вообще не изучают в связи с сокращением часов на эти дисциплины. Даже на такой простой вопрос: что будет, если дискриминант квадратного уравнения меньше нуля, следует ответ, что уравнение не имеет решений. Когда говоришь студентам, что квадратное уравнение всегда имеет решение, то это приводит их в недоумение. Приходится объяснять, что все зависит от множества, на котором ищется решение.

Это свидетельствует о том, что студентам также необходимы знания по теории множеств и теории функций комплексной переменной. Но об этом только приходится мечтать по тем же причинам, которые были изложены выше.

Во-вторых, необходимо установить, если уравнение имеет решение, то сколько их: конечное число или бесконечное? Опять для студентов становится загадкой, как это сделать?

В-третьих, после выяснения второго вопроса, необходима процедура отделения границ корней, что тоже вызывает у них вопросы.

И, наконец, в-четвертых, надо уточнить корни в пределах заданных границ, а для этого необходимо знание методов решения таких уравнений. А это возможно только при овладении такой дисциплиной, как численные методы.

Таким образом, видим, что для решения математических или задач, связанных с математическими моделями, необходимо будущему программисту знание многих дисциплин, которые даже не предусмотрены в учебных планах многих ИТ-направлений.

На четвертом этапе мы и столкнемся с проблемой решения задачи на конкретном языке программирования.

Как говорил известный программист Андерс Хейлсберг, «не бывает хороших языков, бывают хорошие программисты».

Выбор языка программирования в таких случаях также является предметом спора. Предполагается, что студент не использует ППП, а самостоятельно пишет программу.

Парадигма функционального программирования говорит о том, как писать программы. Цель данной парадигмы – оформлять программы как математические функции.

Конечно, за этим стоят какие-то большие цели. Кто-то считает, что это единственный верный способ программирования, кто-то поклоняется данной парадигме, как богу, однако истина находится «между». Иначе говоря, функциональное программирование позволяет сделать код более компактным, красивым, быстрым, но использование только этой парадигмы, игнорируя остальные, идет только во вред.

Важно понять, что требуется от языка? В данном случае, требование одно – большая скорость при небольших ресурсах, которые не будут увеличиваться.

Например, Фортран – старейший язык программирования, предназначенный для решения математических задач, однако он не изучается в курсе программирования. Остаются С++, С# и Java, которые популярны в настоящее время.

Однако, по моему мнению, для решения таких задач более подходит, получивший в последнее время широкое распространение, язык программирования Python. Этот язык является более понятным и доступным для студентов на начальном этапе программирования, и к тому же Python умеет работать с такими языками, как Fortran, С и С++, которые уже широко используются в научных расчетах.

Подытоживая сказанное, я свожу свои соображения к тому, что для формирования хороших навыков будущих программистов необходимо расширять круг изучаемых дисциплин, а также обратиться к опыту советской высшей школы обучения таких специалистов.

Вывод. Подготовка IT-специалистов в современных условиях требует особого подхода при разработке учебного плана для их обучения. Нужно четко очертить круг обязательных, а также профессиональных дисциплин, которые необходимы для подготовки таких специалистов.

Список использованных источников:

1. В.П. Гладков. Методы обучения программированию. – Пермь: Пермский госуд., технич., универ., 2009. – Режим доступа: <https://cyberleninka.ru/article/n/metody-obucheniya-programmirovaniyu>
2. Большаков А.А. Новые методы математического моделирования динамики и управления формированием компетенций в процессе обучения в вузе / А.А. Большаков, И.В. Вешнева, Л.А. Мельников, Л.Г. Перова. – Москва: ГЛТ, 2014, 248 с.