

10. Найдите все непрерывные на числовой прямой функции f , удовлетворяющие тождеству $f(x) = f(\sin x)$.

В плане развития творческого потенциала студентов помимо работы над тематикой стандартных университетских математических курсов необходимо осуществлять работу по расширению их математических горизонтов.

Существует целый пласт классической математической культуры, к которому не прикасаются в школьном курсе математики из-за «неэлементарности» или «ненужности», а в университетском курсе – из-за «элементарности». Многие из этих результатов принадлежат величайшим мировым мыслителям: Л. Эйлеру, И. Ньютону, К. Гауссу, О. Коши, П. Дирихле, Б. Паскалю.

В связи с их «не элементарностью» эти классические результаты элементарной математики называют «олимпиадными», хотя они получены задолго до появления олимпиад. Студентам, проявляющим особый интерес к математике, необходимо дать шанс познакомиться с этой классикой, которая, несомненно, послужит развитию их творческого потенциала.

Библиографический список

1. Саженов А.Н., Саженова Т.В. Математический анализ в задачах студенческих олимпиад. Практикум. Часть 1. – Барнаул: Изд. АлтГУ, 2011.
2. Саженов А.Н., Саженова Т.В., Плотникова Е.А. Математическое творчество: классические олимпиадные темы и задачи высокого уровня сложности. – Барнаул: Изд-во АлтГУ, 2013.
3. Саженов А.Н., Саженова Т.В., Плотникова Е.А. Математический анализ в задачах студенческих олимпиад // Труды семинара по геометрии и математическому моделированию: сб. ст. – Вып. 2. / гл. ред. Е.Д. Родионов. – Барнаул: Изд-во Алт. ун-та, 2016. – С. 50-54.

УДК 378.1

Методика преподавания курса «Основы программирования»

*Л.Л. Смолякова
АлтГУ, Барнаул*

Изучение курса основ программирования является одним из основных при подготовке студентов бакалавриата специальностей, связан-

ных с математикой и информационными технологиями. И это обоснованно. Обучение программированию – дело непростое. Многолетний опыт автора в обучении студентов и школьников позволяет выделить основные причины встречающихся трудностей на этом пути. Студенты приходят в вуз, имея низкий уровень начальной подготовки в этой области. Отсутствует навык самостоятельной работы с учебным материалом. Несмотря на то, что изучение основ структурного программирования входит в государственный образовательный стандарт общего и среднего образования по информатике, значительная часть первокурсников имеет весьма слабое представление о программировании вообще. И то, что единый государственный экзамен по информатике в настоящее время сдается письменно, а не за компьютером говорит об уровне информационных технологий в стране. Государственный образовательный стандарт высшего профессионального образования по дисциплине «Основы программирования» предусматривает изучение объектно-ориентированного подхода к программированию, что оказывается практически невозможным без предварительной базовой подготовки обучаемых. Поэтому, как правило, обучение программированию, приходится начинать с азов.

По окончании данного курса, большинство студентов считают курс «Основы программирования» одним из наиболее сложных и испытывают затруднения при его освоении. Все вышеперечисленное объясняется тем, что программирование является специфическим видом человеческой деятельности, для успешной реализации которой необходимо не только применение приобретенных в процессе обучения знаний и умений, но требуется и наличие определенного стиля мышления, прежде всего, абстрактного, но связанного с решением конкретной задачи. Программирование следует рассматривать как средство развития мышления. Задача преподавателя – не только научить студента записывать алгоритм на языке программирования, а обучить его самостоятельно конструировать сам процесс решения задачи. При правильном подборе учебных задач от более простых к сложным, появится понимание единства принципов написания и функционирования информационных систем.

Следует отметить, что методика обучения на данном курсе представляет собой итерационный процесс. Рассматривая итерацию как пошаговое приближение к определенной цели, можно применить метод итерации, как при изложении лекционного материала, прохождения тестов, так и в процессе выполнения лабораторных работ по данной дисциплине. Тем более что 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. Таким образом, построение итоговой программы с применением одной из парадигм программирования представляет собой итерационный процесс, на каждом шаге которого происходят некоторые изменения, что и позволяет нам применить итерационный метод обучения. Согласно отмеченным положениям, последовательность изложения лекционного материала зависит от порядка практических и лабораторных работ. Учитывая их итерационный характер, изложение лекционного курса также имеет смысл строить на основе итерационного метода.

Все эти аспекты отражены в электронном учебно-методическом комплексе дисциплины (ЭУМКД). На странице курса «Основы программирования» можно обнаружить материалы разного рода. Однако следует помнить, что возможность их использования упрощает, но ни в коем случае не заменяет живого общения с преподавателем. Курс содержит:

1. Слайды к лекциям

Слайды к лекциям могут служить кратким конспектом лекций. Используют их как план освоения прочих материалов: видеолекций, текстов и пр. В начале каждой презентации содержатся задания для самопроверки, которые могут дать представление о наиболее сложных вопросах в тестах.

2. Видеолекции

Видеолекции смонтированы на основе звукозаписей «живых» лекций и до некоторой степени способны их заменить. Пока они не покрывают полный курс, но их коллекция неуклонно пополняется. Используются для восстановления в памяти рассмотренного материала. К сожалению, при просмотре видеолекций нет возможности задать вопрос лектору, однако его можно задать а) в личном сообщении преподавателю, б) в группе ВКонтакте vk.com, в) на очной консультации.

3. Лабораторные работы

Выполнение лабораторных работ – основа практических занятий. Основные навыки программирования вырабатываются именно в эти моменты.

Как правило, на выполнение лабораторной работы отводится 3 недели, при этом сложность работ возрастает с увеличением номера лабораторной работы.

4. Тесты

Тесты предназначены для контроля усвоения теоретического материала, а также умения читать и понимать код программ. Количество попыток на каждый тест ограничено тремя. Зачётные попытки выполнения теста осуществляются только в присутствии преподавателя (на практическом занятии или консультации). Оценка за тест определяется как максимальная из оценок, полученных в зачетных попытках.

5. Рекомендации «Что почитать?»

Элемент «Что почитать?» имеется в большинстве разделов курса и содержит ссылки на фрагменты текстов учебников, вдумчивое прочтение которых является залогом успешного выполнения как тестовых заданий, так и лабораторных работ.

6. Рекомендации «Что посмотреть?»

На страницах с названием «Что посмотреть?» содержатся ссылки на видео-материалы, служащие либо полезными иллюстрациями к материалу соответствующего раздела курса, либо излагающие дополнительные сведения, необходимые для понимания раздела, но выходящие за рамки курса.

7. Примеры программ

В папках с таким названием содержатся исходные коды программ, которые рассматривались во время лекций и фрагментарно представлены на слайдах к лекциям соответствующих разделов. Используются эти исходные коды для самостоятельной компиляции и [пошагового] выполнения программ, чтобы детально разобраться в их устройстве и принципах работы. Кроме того, исходные коды полезны в роли заготовки программ, решающих задачи, поставленные в лабораторных работах.

Каждый следующий учебный материал содержит вопросы из предыдущих тем, и на основе этих вопросов созданы материалы экзамена по дисциплине. Результат экзамена зависит от трех составляющих: среднего балла за лабораторные работы, среднего балла за тест и баллов за экзаменационную работу.

Таким образом, преподавание данного курса с использованием электронного ресурса дает студентам доступ к материалам курса в любое время, а самое главное возможность задать вопросы преподавателю вне занятий, по мере их возникновения, что стимулирует успешное обучение по данной дисциплине.

Библиографический список

1. Смолякова Л.Л., Флорю О.Н. Некоторые аспекты преемственности преподавания информатики // Сборник трудов Всероссийской

конференции по математике «МАК-2017»; Материалы молодежной прикладной IT школы «Математические методы и модели в экологии», Барнаул, 29 июня – 1 июля 2017 г. : [тексты докладов] / АлтГУ [и др.] ; [гл. ред. Н. М. Оскорбин]. – Барнаул : Изд-во АлтГУ, 2017. – С. 279–281.

УДК 372.851

**Повышение мотивации учащихся
на занятиях по математике
через наработку техники некоторых вычислений**

В.Н. Токарев, Е.В. Токарева
АлтГТУ г. Барнаул; АлтГУ г. Барнаул

Обучение математике теряет свою яркость каждый раз, когда учащиеся и студенты сталкиваются с простыми на вид операциями и выполняют их с ошибками. Типичные ошибки учащихся на выпускных экзаменах – неправильное извлечение квадратных корней, элементарные ошибки в вычислениях. Допущение таких ошибок не способствует развитию мотивации и повышению познавательного интереса на занятиях по математике.

Хотя каждый учитель математики сможет объяснить и показать примеры вычисления квадратных корней, лучшим учебником будет «Арифметика» Л. Эйлера [1] с ясным изложением правил, упрощенной техникой вычислений. Эйлер в обращении к читателю делает акцент на необходимости поиска наиболее эффективных способов вычислений: он подчеркивает, что большинство учебников не заботится «о тех способах, чрез которые счисление легче и короче учинить можно, но тем только удовольствуются, чтоб о всем основании в коротких словах показано было». О необходимости обращения к «несовременным» учебникам классиков математической науки и упоминание эйлеровского метода извлечения корней из «составных количеств» было сделано в работе авторов [2].

Данная заметка призвана открыть для читателей метод Эйлера (см. примеры на рисунке 1). Метод прост и универсален, и действительно изящен, не требуется никаких подборов, определения границ корней.

Во-первых, число делится справа на группы по два разряда (можно получать значения и иррациональных корней, на рисунке примеры для чисел 2 и 20). Далее, слева направо: извлекаем корень из самой левой части – двузначного или однозначного числа, записываем результат