А.А. Дивинец, А.А. Маркина, Р. В. Разумейчик
Брест, Брестский государственный технический университет

О модернизации подхода к преподаванию основ алгоритмизации и программирования

Аннотация

Рассмотрен набор подходов к интенсивизации изучения основ алгоритмизации и программирования студентами первой ступени высшего образованиия. Приводится перечень методических приёмов и их результативность в привязке к изучению теоретического материала и практическим занятиям для языка С. Рассматривается использование свободного программного обеспечения в свободной части курса и его связь с последующими курсами соответствующих специальностей.

Введение

Кафедра Электронных вычислительных машин и систем Брестского технического университета ведет подготовку студентов первой и второй ступеней высшего образования, и одной из основных специальных дисциплин, читаемых на первой ступени, является предмет «Основы алгоритмизации и программирования» (ОАиП). Цель преподавания данной дисциплины - освоение студентами методики постановки, подготовки и решения инженерно-технических задач на современных компьютерах. В качестве задач ставятся изучение процедурно-ориентированного языка программирования (при этом в базовых учебных программах для всех специальностей жёстко закреплен язык Си) и автоматизированных средств разработки (IDE), закладывание основ стиля программирования, приобретение навыков разработки и отладки, развитие алгоритмического мышления. Дополнительно особенностями преподавания дисциплины для специальностей первого курса являются: обучение студентов самоконтролю, формирование интереса к программированию, создание мотивации для саморазвития.

Традиционными проблемами являются низкая активность студентов в плане их участия в учебном процессе, адаптация к требованиям ВУЗа после школы, а также низкие показатели алгоритмического мышления и навыков разработки программ. Недостаточно качественное решение этих проблем на первом курсе оказывает сильный негативный эффект на последующие дисциплины. Более того, актуальность эффективного решения данных проблем в рамках дисциплины предсказуемо обострились после разделения высшего образования на первую и вторую ступени привело к сокращению числа учебных часов (причем значительно пострадали лабораторные занятия).

В ходе попыток решения обозначенных проблем в курс были внесены методические изменения, которые будут рассмотрены ниже.

Технологии преподавания

На протяжении последних двух лет преподавания данной дисциплины мы используем ряд методов активного обучения на лекциях в виде символической и текстовой наглядности для улучшения восприятия и усваиваемости материала:

- Лекция-визуализация позволяет схематично и наглядно представить опорную информацию, дополняется видеоматериалом. Этот вид лекции наиболее важен для студентов-визуалов, т. к. с помощью него происходит максимальное усвоение материала. Кроме этого, он повышает знания в области алгоритмизации, развивает системное мышление, позволяет получить наглядность в оформлении и оптимизации кода. Применён на всех темах лекционных занятий.

- Лекция-провокация. Данный вид лекции позволяет проконтролировать осознанность воспринимаемого материала слушателями, развивает у них критическое мышление, повышает дух соперничества и удовлетворенности от образовательного процесса. В начале лекции преподаватель озвучивает сколько ошибок будет допущено в лекции, в конце – проверяются найденные студентами упущения, проверяются и обсуждаются. Применён для следующих тем лекционных занятий: «Циклы», «Массивы», «Рекурсия», «Списки», «Деревья двоичного поиска».

- Лекция-конференция. Построена в форме научно-практического занятия с заслушиванием докладов и выступлений студентов (в малых группах) по ранее подготовленной проблеме в рамках учебной программы, а преподаватель подводит итоги, уточняет информацию, формулирует основные выводы. Данный вид лекции позволяет развить коммуникационный навык, здоровую конкуренцию, критичность и удовлетворение от процесса. Применён для итоговых лекций, завершающих отдельные разделы дисциплины.

- Проблемная лекция проводится в виде диалога со студентами, в процессе которого происходит совместный поиск решения задачи, разбираются типичные ошибки в работах. Такой подход позволяет развить коммуникационный навык молодых людей, а также увеличить осознание алгоритмов и основ процедурного программирования. Применён для лекций, проходящих после промежуточного контроля.

Кроме этого, в ходе лекций происходит комбинирование прослушивания и проверки знаний студентов (2 раза за лекцию), что позволяет более качественно удерживать внимание слушателей.

В ходе проведения лабораторных занятий нами используются:

- Метод «дискуссия», которая применяется во время обсуждения выданных заданий. Она позволяет максимально активизировать мыслительную деятельность студентов и увеличить усвояемость теоретических выводов. Кроме этого, данный метод позволяет увеличить мотивацию учения.
- Метод мозгового штурма направлен на снижение самокритичности студентов, повышение уверенности в себе и обучение навыку творческого подхода к решению проблемы. Применяется в ходе лабораторных задач по темам: «Циклы», «Массивы», «Рекурсия», «Стек», «Деревья двоичного поиска».
- Эвристическая беседа применяется нами при защите лабораторных работ. Этот активный метод обучения строится на вопросах проблемного характера, на который студент дает ответы. В итоге происходит активизация мышления и приобретаются новые знания.

Заметим, что наименее эффективной среди перечисленных показала себя эвристическая беседа: в ряде случаев студенты терялись в ходе попыток ответа (возможно, внешняя причина - язык программирования, изначально создававшийся отнюдь не для обучения студентов). В меньшей степени проблемы возникали с методом мозгового штурма для части студентов, испытывающих затруднения с импровизацией и свободным изъяснением своих мыслей.

В ходе изучения дисциплины имеются формы текущего (два раза в семестр) и итогового контроля. Для повышения заинтересованности и мотивации учебной деятельность мы используем кредитно-рейтинговую систему оценки в течение учебного семестра: студенты знакомятся со списком обязательных работ, требованиями преподавателей, ведётся учет активности и посещаемости. В конце семестра выстраивается рейтинг студентов по результатам их работы, и далее он учитывается в итоговом контроле.

Используемые программные средства

Кросс-платформенность языка C позволяет студентам использовать любые ОС и среду разработки на собственных устройствах. Однако в качестве основных систем в учебных классах установлены Ubuntu Linux и IDE Qt Creator. Преимуществом использования данных инструментальных сред является изначальное формирование у

студентов привычки к многоликости графических оболочек и интерфейсов. Кроме того, элементы работы с командной строкой и средой облегчает преподавание ряда дисциплин на следующих курсах, включая такие как «Конструирование программ и языки программирования» (курс включает создание графических приложений в Qt Creator), «Архитектура ЭВМ» (в рамках которой приобретаются навыки написания элементарных модулей ядра Linux), и др.

Заключение

Хотя используемые меры не являются панацеей, прошедшие два года обучения демонстрируют положительную динамику в овладении знаниями студентов, повышается интерес к будущей деятельности, возрастает мотивация к учебе, легче переносится адаптационный период обучения.