

Опыт преподавания программирования в пандемию

В. Н. Касьянов, email: kvn@iis.nsk.su

Е. В. Касьянова, email: kev@iis.nsk.su

Институт систем информатики им. А.П. Ершова СО РАН
Новосибирский государственный университет

***Аннотация.** В докладе представлен курс по программированию для студентов механико-математического факультета Новосибирского государственного университета (НГУ) и опыт его преподавания в весенний семестр 2019/2020 учебного года в условиях пандемии.*

***Ключевые слова:** курс программирования, пандемия, практикум по программированию, преподавание в вузе, экзамен.*

Введение

Дисциплина «Программирование», представленная в данном докладе, традиционно реализуется на механико-математическом факультете НГУ во 2 семестре в рамках основной профессиональной образовательной программы высшего образования по направлению подготовки «01.03.02 Прикладная математика и информатика», профиль «Прикладная математика и информатика» и «01.03.03 Механика и математическое моделирование», профиль «Механика и математическое моделирование» очная форма обучения.

Задачи освоения дисциплины — это овладение студентами навыками точного формулирования алгоритмов на языках высокого уровня и развитие у студентов алгоритмического мышления посредством упражнений в пошаговой разработке программ. В результате обучения студенты должны иметь представление о проблемах и понятиях программирования, о различных сторонах программирования — как инженерной деятельности, как методологии и как научной дисциплины — и их взаимосвязи; владеть методологией разработки эффективных и корректных алгоритмов на языках высокого уровня, а также практически освоить все этапы разработки надежной и наглядной интерактивной (диалоговой) программы для решения несложной задачи, требующей разработки алгоритма, обработки сложных структур данных и создания дружеского интерфейса.

Преподавание данной дисциплины ведется для порядка 120 студентов механико-математического факультета, разбитых на 9

учебных групп, и предусматривает следующие виды учебной работы: лекции одновременно для всех студентов (2 часа в неделю), практические занятия для студентов каждой группы (2 часа в неделю), лабораторные занятия для студентов каждой группы (2 часа в неделю), а также самостоятельная работа студентов. Самостоятельная работа включает: разбор лекционного материала, выполнение индивидуальных домашних и лабораторных заданий, подготовку к контрольным работам, подготовку к устному экзамену, который проводится в конце 2 семестра.

Поэтому ситуация, вызванная пандемией, привела к необходимости перевести в дистанционный формат такие виды учебной работы как лекции, практические занятия, лабораторные занятия и экзамен.

1. Лекции и семинарские занятия

Традиционно в основу нашего курса положен принцип концентрического изложения материала (см., например, [1, 2]). Предполагается, что с первых занятий студенты начинают упражняться в составлении программ (в последние годы на языке C), которые могут реально выполняться на доступных компьютерах, и постепенно овладевают навыками разработки на языках высокого уровня линейных, ветвящихся и итеративных алгоритмов, алгоритмов со структурами данных и подпрограммами. Также постепенно, одновременно с расширением класса решаемых задач, студенты углубляют свои знания о языках высокого уровня. Другими основными методическими и технологическими принципами, на которых также традиционно базируется наш курс, являются следующие.

1. Принцип обучения конструированию программ на подробно комментированных образцах решения тщательно подобранных задач. Назначение примеров — не только дать образцы и описать основные схемы алгоритмов, но и на сравнительном анализе разных решений одной и той же задачи познакомить студента с такими понятиями, как эффективность, наглядность и надежность решения.

2. Принцип доказательного программирования, когда программа строится вместе с доказательством ее правильности. Для этого в курсе вводятся понятия промежуточных утверждений и инвариантов, а в разрабатываемых алгоритмах решения задач такие утверждения записываются в форме программных комментариев.

3. Принцип пошаговой разработки программ, когда программа строится из формальной спецификации задачи с помощью мелких формально проверяемых шагов преобразования.

4. Принцип модульного программирования, позволяющий проектировать, разрабатывать и собирать программу по частям и с использованием библиотек уже готовых частей.

5. Принцип объектно-ориентированного программирования, позволяющий разработчикам программ легко создавать все более сложные приложения с помощью инкапсуляции, наследования и полиморфизма.

Семинарские занятия проходят по следующему плану.

1. Понятие алгоритма и программы. Построение линейных программ обработки числовых данных.

2. Стандартные и конструируемые простые типы данных. Построение ветвящихся программ.

3. Операторы цикла с условием на продолжение и окончание. Обработка текстов и числовых последовательностей переменной длины.

4. Типы структур. Обработка последовательностей структур переменной длины.

5. Типы массивов. Операторы цикла с параметрами. Задачи сортировки и поиска.

6. Обработка таблиц и матриц.

7. Промежуточная контрольная на составление двух программ, одна из которых требует обработки матрицы, а другая — обработки таблицы.

8. Файлы. Задача внешней сортировки.

9. Подпрограммы: процедуры и функции.

10. Рекурсия. Рекурсивные методы поиска и сортировки. Порождение комбинаторных объектов.

11. Указатели. Работа со списками, очередями и стеками.

12. Древоподобные структуры и их реализация. Обходы деревьев.

13. Графы и их реализация. Алгоритмы обработки графов.

14. Модули и модульное проектирование программ.

15. Итоговая контрольная. Состоит из двух задач, первая из которых требует построения программы для обработки последовательности матриц с использованием подпрограмм с параметрами, а вторая — рекурсивной подпрограммы для обработки дерева.

Все занятия по курсу в 2019/2020 учебном году начались в начале февраля в обычном очном формате и в аудиториях НГУ, но уже в середине марта перешли в дистанционный формат.

Сначала это коснулось лишь лектора В.Н. Касьянова и семинаристов возраста 65+, которые стали приходить в здания НГУ, чтобы в пустых аудиториях читать лекции и вести занятия дистанционно через Zoom или Google Meet. А затем уже к концу марта для всех преподавателей и студентов университет закрыли для

посещения, и все лекции и семинарские занятия стали проходить удаленно и вне университета через Zoom или Google Meet.

Оказалось, что данные программы достаточно удобны с точки зрения поддержки нашей традиционной практики очного проведения лекций и семинаров, когда преподаватель либо рассказывает новый материал, используя доску и отвечая на вопросы студентов, либо вызывает одного из студентов к доске и вместе с другими студентами помогает ему решать некоторую задачу.

2. Лабораторные занятия

Основной задачей лабораторного практикума является не только и не столько обучение студентов собственно записи (кодированию) известного алгоритма на языке программирования высокого уровня, а практическое закрепление знаний, получаемых в курсе по программированию, и овладение общими методами, приемами и навыками технологии решения задач на компьютере.

Во время лабораторных работ студент выполняет домашние задания в виде решения десяти несложных задач, заданных ему в ходе семинарских занятий, а также выполняет индивидуальное семестровое задание, подробные рекомендации по выполнению которого содержатся в учебном пособии [3], а также в главе 8 книги [2].

Каждая домашняя задача и каждая задача, предлагаемая студентам в рамках контрольных работ, — это небольшая задача по программированию, взятая из первых семи глав книги [2]. Каждое индивидуальное семестровое задание состоит из трех индивидуальных более сложных задач, взятых из разных тематических разделов (графы и системы дорог; грамматики, языки и автоматы; формулы и программы; геометрия; игры и модели) учебного пособия [3] или главы 8 книги [2].

Каждая сданная домашняя задача оценивается в 1 балл, а полностью сданное индивидуальное лабораторное задание — в 30 баллов (по 10 баллов за каждую из трех задач, составляющих задание). За каждую из двух контрольных работ, выполняемых студентами в ходе семинарских занятий, студент может получить 0, 3, 4 или 5 баллов.

Каждая задача, входящая в индивидуальное семестровое задание, — это самостоятельная, как правило, комбинаторная или логическая задача с краткой и четкой формулировкой, не содержащей описания алгоритма.

Предполагается, что студент, решая задачу, создает интерактивную C-программу, которая за один запуск может обрабатывать не отдельный входной набор, а последовательность входных наборов произвольной длины. Эти входные наборы либо заранее размещаются в специальных «входных» файлах программы, предъявляемых преподавателю вместе с

разработанной программой, либо задаются пользователем программы в процессе ее исполнения.

Таким образом, разработанная студентом программа после завершения обработки очередного входного набора в зависимости от указания пользователя может либо завершить свою работу (остановиться), либо продолжить ее и перейти к обработке следующего входного набора. В последнем случае программа может вступить в диалог с пользователем, в процессе которого пользователь программы может подготовить и инициировать новый счет по программе. Для этого он может либо осуществить весь очередной ввод с помощью клавиатуры и мышки, либо дать указание программе, из какого файла ей нужно взять необходимые данные.

При этом, создавая программу обработки последовательности наборов входных данных, совсем не требуется создавать программу так, чтобы при ее переходе к обработке следующего входного набора этот набор каждый раз целиком задавался заново. В частности, во многих заданиях удобно разделить входной набор на две части, выделяя более общие (как правило, более объемные) исходные данные задачи в отдельную часть, и предусмотреть специальный вид реализации счета по программе на последовательности входных данных набора, различающихся второй частью, без необходимости повторного задания первой.

Например, в задаче нахождения кратчайшего пути между двумя заданными вершинами заданной системы дорог такой более общей частью входного набора является система дорог. Поэтому естественно решение, при котором система дорог является общей для нескольких следующих друг за другом счетов по программе. В этом случае каждый раз, переходя к новому счету по программе, пользователь может выбрать один из следующих двух вариантов продолжения работы: осуществить ввод очередной системы дорог из заданного файла с выводом ее изображения на экран; запустить нахождение кратчайшего пути между двумя заданными городами в текущей системе дорог.

Для проведения лабораторных занятий в дистанционном формате большинство преподавателей использовало Skype. Практика показала, что данная программа позволяет в наибольшей степени обеспечивать поддержку нашего традиционного сценария очного проведения лабораторных работ, когда все студенты группы работают над своими заданиями под наблюдением преподавателя и могут в любой момент обратиться к нему за консультацией, а преподаватель наблюдает за работой студентов, взаимодействует с ними по очереди, консультирует

их и принимает задания у студентов, которые демонстрируют работу созданных ими программ на своих компьютерах.

3. Экзамен

Проведение консультации, экзамена и пересдач осуществлялось также дистанционно и потребовало разработки соответствующих сценариев проведения экзамена и пересдач, причем, в двух разных вариантах: для студентов и для преподавателей.

Ниже мы приводим сценарий проведения экзамена для преподавателей, который состоит из трех последовательных этапов: до экзамена, проведение экзамена и после экзамена. Отметим, что каждая задача, включенная в список задач для экзамена, упоминаемый в сценарии, требует построения программы для обработки последовательностей матриц произвольной длины. Формулировка каждой такой задачи содержит унарную функцию Φ , реализация которой имеет вид гнезда из двух вложенных циклов с вычислениями не только вне внешнего цикла и внутри вложенного цикла, но и вне вложенного цикла, но внутри внешнего цикла.

3.1. До экзамена

Всем участвующим в экзамене следует ознакомиться с функционалом Zoom и со сценарием экзамена. В процессе экзамена нужно организовывать все конференции через приложение Zoom, установленное на вашем компьютере (не используйте для этого браузер).

Не позже чем за день до экзамена лектор (куратор) рассылает семинаристам (экзаменаторам) их аккаунты в Zoom, список задач для экзамена, а также число N .

Каждый экзаменатор начинает готовить отчет об экзамене, заполняя всю информацию о студентах, сдавших ему экзамен досрочно. Для каждого студента экзаменатор заполняет информацию о его билете, состоящем из теоретического вопроса $((XYZ + N) \text{ MOD } K)+1$ и задачи $((XYZ + N) \text{ MOD } M)+1$, где XYZ — последние три цифры номера зачетки студента, K — количество вопросов по курсу, M — количество задач для экзамена, а N — число, присланное куратором.

При оценивании ответа студента на экзамене экзаменатор использует следующие правила:

- оценка не может быть положительной, если студент не набрал 46 баллов за работу на семинаре,
- оценка не может быть положительной, если нет правильного описания унарной функции Φ , присутствующей в формулировке задачи,
- ответ нельзя оценить на «хорошо» или «отлично», если студент не предоставил такого правильного решения задачи, которое может

обрабатывать входную последовательность матриц, длина которой превышает размер используемой программой памяти (в частности, не признается решение, в котором вся входная последовательность вводится в некоторый массив) и содержит помимо функции Ф процедуру с параметрами,

– оценка не может превышать оценки за ответ на теоретический вопрос.

3.2. Проведение экзамена

8:30. Каждый студент убеждается, что у него есть и работают Интернет, наушники, микрофон, камера, есть Zoom.

8:45. Каждый экзаменатор организует свою конференцию в Zoom с включенным залом ожидания и приглашает на нее студентов своей группы. Каждый студент при регистрации представляется, называя себя, показывает на камеру свой паспорт, студенческий билет или зачетную книжку.

9:00. Экзаменатор каждому зарегистрированному и набравшему не менее 46 баллов студенту (и куратору) на университетскую почту высылает его задачу в письме с темой «Экзамен по программированию» следующего содержания «Здравствуйте! Отправьте до 9.45 ответное письмо мне и лектору с текстом программы для решения следующей задачи:...». Всех остальных зарегистрированных студентов экзаменатор отправляет на пересдачу, заполняет отчет и просит студентов покинуть конференцию.

Студент решает полученную им задачу на языке C или Pascal, не выключая камеру, и отправляет ответным письмом экзаменатору и куратору написанный им в любом редакторе текст программы. Получив решение, экзаменатор в чате сообщает студенту его теоретический вопрос. После этого студент отключается от конференции, готовит ответ на теоретический вопрос, пользуясь литературой, и до 10:15 отправляет экзаменатору и куратору подготовленный им файл с ответом. Затем студент опять подключается к конференции.

Экзаменатор фиксируют все моменты, когда изображение студента «пропадает» во время решения задачи. Если проблемы с изображением повторяются или время отсутствия изображения достигает 15 минут, студент отправляется на пересдачу, а в ведомость ему проставляется «неявка». При предоставлении документов в деканат, подтверждающих проблемы со связью, «неявка» будет по уважительной причине.

9:45. Всех тех студентов, кто не отослал решения задачи до этого времени, экзаменатор отправляет на пересдачу, заполняет отчет и просит студентов покинуть конференцию.

Если найдется студент, который не пожелает покидать конференцию, экзаменатор организует новую конференцию (с включенным залом ожидания) и приглашает на нее продолжающих сдавать экзамен студентов и куратора.

10:15. Всех тех студентов, кто не прислал ответа на теоретический вопрос до этого времени, экзаменатор отправляет на пересдачу.

Экзаменатор включает запись, называет себя, название предмета и дату, проводит собеседование с оставшимися студентами, «запуская» из их зала ожидания в порядке их второго подключения и сообщая им о том, что ведется видеозапись сдачи:

- студент представляется, называя себя и демонстрируя на камеру свой документ,

- экзаменатор выводит на экран присланную студентом программу и обсуждает с ним решение задачи, выводит на экран присланный студентом ответ на теоретический вопрос и слушает комментарии студента, задает студенту дополнительные вопросы, на которые студент отвечает без подготовки,

- по окончании собеседования экзаменатор называет студенту его оценку, заполняет отчет, просит студента покинуть конференцию и «запускает» следующего.

Экзаменатор завершает экзамен не позже 14:00. После завершения конференции он должен дожидаться, пока обработается и сохранится видеозапись, и затем переписать ее себе.

Если в ходе экзамена запись пришлось прервать (например, наступило время запланированного перерыва), нужно сказать вслух, что завершена очередная часть записи, а затем при ее восстановлении сказать, что начата следующая часть записи.

3.3. После экзамена

Экзаменатор готовит и (не позже чем через 3 часа) отправляет куратору отчет об экзамене, в котором он указывает следующую информацию о каждом студенте своей группы:

- ФИО студента;
 - номер зачетки;
 - задача;
 - теоретический вопрос;
 - оценка;
 - обоснование оценки;
 - описание конфликтных ситуаций, если они возникали,
- а также дает свои предложения по улучшению сценария экзамена, если они есть.

Например, для оценки «неявка» возможны следующие варианты обоснования: «неявка студента», «неоднократное отключение студента», «отключение студента более чем на 15 минут», «неоднократное отключение камеры студента» или «отключение камеры студента более чем на 15 минут»), а для оценки «неудовлетворительно»: «недобор баллов», «несамостоятельность студента при написании программы», «непредставление программы в срок», «неправильность функции Ф», «непредставление в срок ответа на теоретический вопрос», «неудовлетворительный ответ на теоретический вопрос» или «неудовлетворительный ответ на такой-то дополнительный вопрос».

Заключение

Отметим, что наш опыт преподавания программирования в пандемию был довольно успешным, хотя и потребовал дополнительных усилий как от студентов, так и от преподавателей. Тем не менее, у нас получилось перенести в дистанционный формат все формы обучения, сохранив их диалоговый характер. Это во многом определилось, на наш взгляд, следующими тремя основными причинами. Во-первых, тем, что все преподаватели, участвующие в занятиях со студентами, занимаются программированием профессионально, что позволило им без особых сложностей овладеть новыми технологиями цифрового обучения. Во-вторых, тем, что все преподаватели и практически все студенты вне зданий университета и без привлечения университетских ресурсов были обеспечены техническими средствами и возможностями, позволяющими проводить дистанционное обучение. Третья причина — это развитость сайта НГУ с точки зрения поддержки процесса обучения, его удобства для организации как индивидуального, так и группового взаимодействия по почте и через локальные диски между всеми студентами и преподавателями, вовлеченными в процесс обучения.

Понятно, что дистанционное обучение не во всем хуже очного, поскольку обладает известными достоинствами, главным образом, связанными с его большей доступностью. В нашем случае они тоже проявились, хотя и не без некоторых негативных последствий. Например, каждый студент или преподаватель легко может записать и в дальнейшем произвольным образом использовать запись любого этапа дистанционного обучения. Понятно, что возможность повторного просмотра лекции или семинара полезна большинству студентов, но на практике такая возможность приводит к тому, что существенная часть студентов перестает участвовать в текущем учебном процессе, откладывая просмотр и изучение материала на потом. Также при дистанционном формате преподавателю становится проще проверять программы студентов, поскольку нет необходимости разбирать каракули

студентов, которые теперь пишут контрольные работы и программы на экзамене с использованием языкового редактора, но при этом труднее добиться, чтобы студенты писали их самостоятельно.

Список литературы

1. Касьянов, В. Н. Вопросы подготовки специалистов по программированию // Проблемы информатики и ее применения в управлении, обучении и научных исследованиях. — София: СУ им. Климента Орхидского, 1985. — С.138–148.

2. Касьянов, В. Н. Курс программирования на Паскале в заданиях и упражнениях. — Новосибирск: НГУ, 2001. — 448 с.

3. Касьянов, В. Н. Практикум по программированию / В. Н. Касьянов, Е. В. Касьянова. — Новосибирск: НГУ, 2013. — 198 с.