

СИСТЕМА КОМПЬЮТЕРНОЙ АЛГЕБРЫ КАК ПЕДАГОГИЧЕСКАЯ ЗАДАЧА

Поздняков С.Н.¹

¹СПбГЭТУ «ЛЭТИ», Санкт-Петербург, Россия

Аннотация

В статье представлен опыт организации командной деятельности студентов по разработке компонент системы компьютерной алгебры как один из видов учебной деятельности, сопровождающий чтение курса дискретной математики и связанный с алгоритмами работы над длинными целыми числами и многочленами. Особенностью организации этой работы является целенаправленный подбор «граничного объекта», который является основой общего информационного пространства.

В качестве граничного объекта была выбрана структура технического задания на разработку системы компьютерной алгебры, в котором фиксировались имена модулей и связи между ними, но не фиксировался ни язык, ни структура данных. Использование такого граничного объекта для организации общего информационного пространства, с одной стороны, обеспечило достаточную свободу командам студентов в принятии решений по архитектуре создаваемой системы, с другой стороны, точно определило параметры выполняемой работы, что позволило сравнивать результаты работы различных групп и оценивать качество выполненных работ и качество организации совместной деятельности в командах.

Результаты работ и анкетирование студентов показали, что такой метод учебной работы даёт хорошие результаты по качеству выполненных проектов и высоко оценивается самими студентами, оправдывая их ожидания от обучения в техническом университете.

Ключевые слова: *система компьютерной алгебры, обучение математике, связи математики и информатики, современные технологии обучения.*

Цитирование: Поздняков С.Н. Система компьютерной алгебры как педагогическая задача // Компьютерные инструменты в образовании. 2017. № 2. С. 25–41.

ВВЕДЕНИЕ. О ВЗАИМОВЛИЯНИИ ПРЕПОДАВАНИЯ МАТЕМАТИКИ И ПРОГРАММИРОВАНИЯ

Преподавание дискретной математики в технических вузах на факультетах, связанных с информатикой, ставит вопрос о взаимовлиянии курсов дискретной математики и информатики и программирования как по содержанию, так и по видам учебной деятельности, связанной с практической работой по этим курсам. Студенты, которые приходят на «компьютерные» факультеты, как правило, мотивированы к деятельности в области информационных технологий, имеют некоторый опыт программирования и использования информационных технологий, однако не имеют хорошей теоретической основы в алгоритмической и дискретной математике.

Заметно несоответствие ожиданий недавних абитуриентов и традиций организации учебной деятельности в технических университетах. Студенты что-то слышали о технологиях работы в современных компьютерных фирмах, командной работе над проектами, у них есть возможность устроиться в компьютерные фирмы тестерами или «кодерами». Эта готовность к конструктивной деятельности наталкивается на традиционную организацию учебной работы по математике в форме лекций и практических занятий. Как правило, высокого уровня в иерархии специалистов в области информационных технологий достигают только те, кто хорошо владеет дискретной и алгоритмической математикой. Тем не менее, лишь часть студентов готова к тому, чтобы заинтересоваться теоретическими предметами, не вполне понимая, как они пригодятся в будущем.

О роли математики в подготовке специалистов в области информационных технологий говорит опрос специалистов, представляющих разные направления компьютерной индустрии. Вот несколько характерных суждений по этому вопросу.

«Нужно читать математику по мере столкновения с проблемами при изучении самого программирования... Например, программисту обычно не интересно в чистом виде слушать курс комбинаторики. Но прослушать его на фоне анализа быстродействия только что изученных алгоритмов обработки данных намного интересней!» [1].

«Математика дает основу для структурного мышления, нужного для системного анализа, а в программировании без системного анализа становится все труднее, ведь сложность систем начинает превышать возможности одного отдельно взятого человека» [2].

«Программисту знать математику в принципе важно, потому что те задачи, которые он решает, очень часто связаны со знаниями основных математических понятий... закладывать красивые решения [нужно] в самом начале проектирования системы — это основа основ. Для того чтобы получить это красивое решение, ты должен иметь базовые фундаментальные знания, понимать, в чем красота, в чем не красота, как посмотреть, что данное решение оптимально для конкретного круга задач, которые решает твое программное обеспечение» [3].

Реже, но встречаются и иные мнения, например: «Связь математики и программирования — очень популярное заблуждение. Есть очень мало задач, где знание некоторых направлений математики вам пригодится. А вот умение общаться в команде или понимать пользователя нужно в любом направлении программирования» [2].

Суммируя различные мнения специалистов, можно выделить несколько важных рекомендаций для проектирования новых видов учебной деятельности, в большей степени учитывающих ориентацию на профессиональные интересы студентов:

- важным является возможность участия не только в разработке программного продукта, но и в его проектировании,
- изучение курса математики студентом, интересующимся программированием, будет эффективнее, если курс будет поддержан деятельностью по реализации изучаемых алгоритмов,
- нужно познакомить студентов не только с отдельными алгоритмами, но и со сложными системами, основанными на множестве алгоритмов, когда разрабатывается не одна простая программа, а система — программный комплекс, в котором отдельные модули должны взаимодействовать,
- даже для тех, кто не интересуется математикой, работа в команде может послужить мотивационным импульсом к изучению определенных разделов математики, алгоритмов, которые нужно реализовать программно.

1. СИСТЕМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ В УЧЕБНОМ ПРОЦЕССЕ

Учитывая содержание программы дискретной математики для студентов первого курса, которая включает в себя изучение алгоритмов для работы с длинными целыми числами и многочленами, в качестве объекта для реализации межпредметных связей и организации активной деятельности по разработке программного продукта, связанного с тематикой курса, можно взять системы компьютерной алгебры.

Целесообразность использования такого объекта при обучении студентов, готовящихся стать программистами, подробно раскрыта в работах Н.А. Калининой [4, 5]. В этих работах разработка систем компьютерной алгебры (СКА) рассматривалась в двух аспектах: применение СКА при изучении общих математических курсов и использование СКА при подготовке студентов, специализирующихся в информатике: «Особый интерес представляет разработка систем компьютерной алгебры как метод обучения. Освоение методов СКА идет успешнее, если студент приобретает опыт участия в разработке СКА». В работах А.Н. Калининой испытательным полигоном для студентов явилась разработка системы FABULA, функциями которой было проведение аналитических преобразований булевых выражений в символьной форме и решение оптимизационных задач и систем уравнений в булевых алгебрах [5]. Однако педагогический эффект от разработки систем компьютерной алгебры силами студентов не зависит от характеристик системы, а определяется системностью подхода, доступностью такой работы для студентов и связью с теоретическими курсами.

Работа с системами компьютерной алгебры в небольшом объеме реализована в проектах по использованию символьных калькуляторов (проекты фирмы Texas Instruments — [6]). Однако эти проекты не предполагают создания систем компьютерной алгебры, а лишь имеют в виду использование базовых инструментов. Проект Wolfram Mathematica существенно ближе к программированию, и его можно использовать для разработки программных продуктов на основе языка Wolfram Mathematica [7], в который уже встроены алгоритмы для работы с математическими объектами. Использование языка Wolfram Mathematica для поддержки курса дискретной математики можно рассматривать как другое направление использования возможностей интегрирования математики и программирования, что, однако, сдерживается коммерческим характером продукта.

Для пользователей систем компьютерной алгебры представляет интерес проект MathPartner [8], который можно использовать подобно системе Mathematica. Он позволяет создавать математические курсы с включением в них примеров по работе со сложными математическими объектами, которые нельзя просчитать вручную и которые вставляются в текст лекции автоматически.

2. ОРГАНИЗАЦИЯ ПРОЕКТНОЙ КОМАНДНОЙ ДЕЯТЕЛЬНОСТИ СТУДЕНТОВ В РАМКАХ УЧЕБНОГО ПРОЦЕССА

Важным аспектом организации учебной деятельности студентов технического вуза является проектная деятельность, особенно выполняемая командами студентов и базирующаяся на межпредметных связях общих и специальных предметов. Проектную работу со студентами первого курса вуза необходимо рассматривать как переходный этап от обучения в школе к более самостоятельному и целенаправленному обучению в вузе. Проектная работа студентов первых курсов должна сочетать в себе как элементы про-

ектной деятельности со школьниками, так и элементы проектной деятельности со студентами старших курсов, которая приближена к реальной профессиональной деятельности.

Основы проектного метода связывают с именем американского педагога Дьюи, который предлагал отталкиваться в обучении не от формализованных и разделенных между учебными предметами знаний, а от практической деятельности и возникающих в её процессе задач, требующих интегральных знаний из разных предметных областей. Такой подход к обучению подвергался критике из-за несистематичности и излишней конкретности получаемых знаний, однако неоднократно возрождался в новых формах при появлении новых технологических возможностей для обучения и расширения спектра форм обучения, так как он хорошо соответствует естественной мотивации к обучению, возникающей в процессе столкновения молодых людей с проблемными практическими ситуациями, и решению проблемы их социальной адаптации.

В настоящее время этот подход позиционируется как «метод проектов». Его внедрение связано с развитием технологий передачи, хранения и обработки информации. Обучаемый имеет доступ к большому числу источников, в том числе и к обучающим курсам, компьютерным моделям и инструментам. Преподаватель перестал быть главным и единственным источником информации, а его функции стали включать организацию самостоятельного обучения и построения индивидуальных маршрутов обучения.

Так, в работе [9, с. 58–59] приводятся следующие требования к применению метода проектов к обучению в школе:

1. Наличие значимой в исследовательском, творческом плане проблемы/задачи, требующей интегрированного знания, исследовательского поиска для ее решения.
2. Практическая, теоретическая, познавательная значимость предполагаемых результатов.
3. Самостоятельная (индивидуальная, парная, групповая) деятельность учащихся.
4. Структурирование содержательной части проекта (с указанием поэтапных результатов).
5. Использование исследовательских методов: определение проблемы, вытекающих из нее задач исследования, выдвижение гипотезы их решения, обсуждение методов исследования, оформление конечных результатов, анализ полученных данных, подведение итогов, корректировка, выводы (использование в ходе совместного исследования метода «мозговой атаки», «круглого стола», статистических методов, творческих отчетов, просмотров и др.).

Сравним эти требования с требованиями к организации проектной работы студентов старших курсов, где метод проектов рассматривается и как способ подготовки студентов к работе с реальными техническими задачами.

Так в работе [10] выделены следующие основные требования к проектному обучению:

1. Наличие значимой в теоретическом, исследовательском, техническом плане задачи, приводящей к созданию определенного продукта.
2. Разрабатываемый продукт и пути решения должны отличаться оригинальностью и новизной.
3. Для выполнения работы над проектом должна быть создана команда.
4. Работа над проектом должна выполняться в рамках жизненного цикла любого проекта (обзорная работа, исследовательская, техническое задание, стадия проектирования и др.).

5. Результаты проектирования должны представляться на выставках, а также в виде публикаций и докладов на семинарах и конференциях.
6. Работа, как правило, имеет возможность продолжения, с целью коммерциализации результатов, а участники — перспективу создания собственного предприятия.

Как видно, общая структура требований в школе и вузе одинакова.

Отличия проектной работы студентов от проектной работы школьников и от профессиональной проектной деятельности представлены в работе Ю.А. Коваленко [11]: «Задачей студента является осуществление перехода к самостоятельной профессиональной деятельности за время обучения в вузе. Он как бы вступает в другой тип деятельности: накопления знаний, умений осуществлять проектную деятельность и организовывать её в будущей профессиональной деятельности. Проектная деятельность студента носит сложный, многофункциональный характер. С одной стороны, его основная задача — приобретение знаний. С другой стороны, студент должен подготовить себя к самостоятельной профессиональной деятельности после окончания вуза. Следовательно, проектная деятельность студентов связывает две стороны процесса познания. С одной стороны, она является методом обучения, с другой — средством практического применения усвоенных знаний и умений в области будущей профессиональной деятельности».

Поддержка проектной деятельности, в которой участвуют несколько сотен студентов, требует особого внимания к ее организации. Преподаватель уже не может разобратся детально во всех тонкостях реализации проекта. Решение этой проблемы может быть достигнуто разными способами. Один из них — автоматизация проверки работ. Однако это возможно только, если работы типизированы. Проектная работа предполагает большую свободу студентов в принятии решений, поэтому автоматизация проверки работ не представляется возможной. Другой способ — общественная приемка работ, которая включает в себя выступление с результатами на итоговой конференции и предварительное перекрестное рецензирование работ другими студентами [12]. Перекрестное рецензирование широко используется за рубежом и имеет технологическую поддержку [13]. Наиболее распространенный способ — получение трех независимых рецензий от других участников.

В течение 2015–2017 годов на первом курсе факультета компьютерных технологий и информатики СПбГЭТУ «ЛЭТИ» в рамках курса дискретной математики был проведён эксперимент по разработке систем компьютерной алгебры. Работа проводилась в рамках дополнительной деятельности по курсу дискретной математики, результаты заслушивались на защите выполненных проектов. В эксперименте ежегодно участвовало более 200 студентов.

Перейдем к подробному описанию работы.

3. ПОСТАНОВКА ПРОБЛЕМЫ

Современные студенты компьютерных специальностей хотят решать программистские задачи в современном контексте. Студенты стремятся «примерить на себя» разные профессиональные роли: архитектора системы, организатора работ и ответственного за их качество, разработчика, тестера, сдатчика работ и пр. У студентов первого семестра читается интенсивный курс программирования, который выравнивает знания студентов и учит основам программирования тех из них, кто не овладел программированием в школе. Курс программирования продолжается и во втором семестре. Параллельно с изучением программирования студентам читаются математические курсы. Во вто-

ром семестре сразу после интенсивного курса программирования студентам читается курс дискретной математики, который в определенной степени можно рассматривать как курс теоретической информатики. Среди студентов можно выделить две категории: «программистов» и «теоретиков». Первые склонны к практической деятельности в области программирования, хотя не всегда понимают неэффективность реализуемых ими решений и не склонны к системному взгляду на предмет, в школе они не отличались хорошим знанием математики, и, используя сленг, их можно было бы отнести к категории «хакеров». Вторая категория недавних школьников включает тех, кто был более успешным в школе и имеет интерес к занятиям математикой, физикой, решению теоретических задач, но при этом не изучал в школе программирования. Такая ситуация подталкивает к поиску повышения эффективности обучения за счет использования межпредметных связей между математикой и программированием.

4. ПРЕДЛАГАЕМЫЙ ПОДХОД К РЕШЕНИЮ ПОСТАВЛЕННОЙ ПРОБЛЕМЫ

Предлагаемый подход состоит в том, чтобы после изучения части курса предложить студентам самостоятельно разработать систему компьютерной алгебры, включающую более 40 модулей, которые описаны своими именами и связями между ними. Для реализации этого подхода необходимо наличие в курсе достаточного числа понятий и алгоритмов, которые, будучи реализованы программно, представляют другую форму математических знаний и, следовательно, будут влиять на овладение соответствующими математическими идеями. Для этого оказалось удобным выбрать первую часть курса дискретной математики, которая читается в течение первых двух месяцев семестра [14]. В ней обсуждается ряд алгоритмов с целыми числами, таких как алгоритм нахождения наибольшего общего делителя, алгоритм решения линейных диофантовых уравнений, алгоритмы вычисления степеней, основанные как на малой теореме Ферма и Эйлера, так и на бинарном алгоритме быстрого возведения числа в степень, арифметика остатков и китайская теорема об остатках, алгоритмы факторизации чисел и многочленов, в том числе алгоритм разложения на свободные от квадратов множители и пр.

Работа выполняется командами, каждая из которых должна реализовать все модули системы. В то же время, за командами остается свобода в выборе языка реализации, структур данных, интерфейсов. Команды формируются самими студентами на базе учебных групп (кроме выделения команд из группы допускаются и межгрупповые команды). В созданной командой системе должен присутствовать пользовательский интерфейс, и его выбор находится в компетенции студента-архитектора. Фактически, результатом являются два пакета: один — с операциями над длинными числами и многочленами, второй пакет — интерфейс. Эти два пакета образуют вместе систему компьютерной алгебры. Взаимодействие двух пакетов — зона ответственности студента-архитектора.

В качестве главных действующих лиц группа выбирает архитектора системы и человека, ответственного за качество. Первый, по согласованию с командой, выбирает язык реализации и структуру данных, проектирует интерфейсы между модулями, организует тестирование и сборку модулей, второй — следит за тем, чтобы все студенты выполняли доступную им работу и получали пользу от выполнения проекта. Также в работу ответственных за качество работ входит совместное определение метрик для оценки размера модулей, проведение анкетирования студентов для получения требуемой преподавателем информации о характеристиках процесса работы (содержание вопросов преподавателем не регламентируется).

На работу отводятся три недели. Первая неделя — знакомство с задачей, определение состава групп и выбор архитектуры системы. Вторая — выполнение работ. Третья — сборка, тестирование, анкетирование. На четвертой неделе в режиме конференции проходит прием работ. Предварительно отчеты по работам выкладываются в общий для студентов доступ, и происходит перекрестное рецензирование выполненных работ: каждая команда должна дать рецензии на работы трех других команд (и, тем самым, каждый созданный продукт тестируется трижды представителями других команд). Докладчиками от каждой команды выступают двое случайно выбранных преподавателем студентов (не архитектор, не ответственный за качество). Им дается по 5 минут на выступление и по 5 минут для ответов на вопросы, которые задаются из аудитории (в основном, архитекторами других команд). Отдельно 5 минут даются ответственному за качество, чтобы он рассказал об организации работ и привел результаты анализа созданного продукта и результаты анкетирования студентов.

5. ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ПРЕДЛОЖЕННОГО ПОДХОДА: СОЗДАНИЕ ОБЩЕГО ИНФОРМАЦИОННОГО ПРОСТРАНСТВА НА ОСНОВЕ ПРАВИЛЬНОГО ВЫБОРА ГРАНИЧНОГО ОБЪЕКТА

Для того чтобы организовать совместную деятельность студентов, нужно правильно организовать общее информационное пространство.

Понятие общего информационного пространства впервые было изучено в работе Vannon L, Vodker S. [15]. Было показано, что для некоторых видов деятельности общее информационное пространство может служить равноправным инструментом организации коллективной работы. Общее информационное пространство используется для поддержки работы практикующих сообществ (practical societies), которые имеют общие знания (контекст, context), достаточные для того, чтобы упростить передачу содержательной информации между членами практикующего сообщества. Для поддержки передачи контекстно-зависимой информации используются так называемые граничные объекты — структуры, достаточно жёсткие, чтобы однозначно определить суть передаваемой информации и, в то же время, достаточно гибкие, чтобы в них можно было представить все особенности конкретной ситуации в области совместной деятельности. В качестве примера граничного объекта авторы приводят библиографические карточки в библиотеке. Правильно разработанный граничный объект является основой эффективности конструируемого общего информационного пространства. Применительно к поставленной задаче требовалось выбрать или сконструировать граничный объект для передачи задания студентам. В качестве такого граничного объекта выбрана структура модулей: их точные названия и связи между ними (табл. 1, 2, 3, 4). Такая структура задания является достаточно гибкой, так как в ней не фиксируется ни язык для реализации системы, ни структура данных для представления объектов. Это дает студентам возможность сгруппироваться по типам языков, какими они владели или хотели овладеть, и по умениям работать с теми или иными структурами данных в этих языках.

Дадим обоснование правильности выбранного граничного объекта. Для этого проведем теоретический анализ педагогических последствий неправильно выбранного граничного объекта и рассмотрим случаи, когда студентам дано больше или меньше свободы в принятии технических решений:

1. Если студентам предложить создать систему компьютерной алгебры без какой бы

Таблица 1. Граничный объект задания на создание системы компьютерной алгебры: имена модулей и связи между ними

	Натуральные числа с нулем: ($n; A[..]$) — номер старшей позиции и массив цифр, цифра D (тип — целое)	Имена	Базовые модули, на которые ссылается данный модуль
N-1	Сравнение натуральных чисел: 2, если первое больше второго; 0, если равно; 1 — иначе	COM_NN_D	
N-2	Проверка на ноль: если число не равно нулю, то «да» иначе «нет»	NZER_N_B	
N-3	Добавление 1 к натуральному числу	ADD_1N_N	
N-4	Сложение натуральных чисел	ADD_NN_N	COM_NN_D
N-5	Вычитание из первого большего натурального числа второго меньшего или равного	SUB_NN_N	COM_NN_D
N-6	Умножение натурального числа на цифру	MUL_ND_N	
N-7	Умножение натурального числа на 10^k	MUL_Nk_N	
N-8	Умножение натуральных чисел	MUL_NN_N	MUL_ND_N, MUL_Nk_N, ADD_NN_N
N-9	Вычитание из натурального числа другого натурального числа, умноженного на цифру для случая с неотрицательным результатом	SUB_NDN_N	SUB_NN_N, MUL_ND_N, COM_NN_D
N-10	Вычисление первой цифры деления большего натурального числа на меньшее, домноженное на 10^k , где k — номер позиции этой цифры (номер считается с нуля)	DIV_NN_Dk	MUL_Nk_N, COM_NN_D
N-11	Частное от деления большего натурального числа на меньшее или равное натуральное число с остатком (делитель отличен от нуля)	DIV_NN_N	DIV_NN_Dk, SUB_NDN_N
N-12	Остаток от деления большего натурального числа на меньшее или равное натуральное число с остатком (делитель отличен от нуля)	MOD_NN_N	DIV_NN_N, SUB_NDN_N
N-13	НОД натуральных чисел	GCF_NN_N	MOD_NN_N, COM_NN_D, NZER_N_B
N-14	НОК натуральных чисел	LCM_NN_N	GCF_NN_N, MUL_NN_N

то ни было спецификации, командами могут быть созданы совершенно разнородные системы, которые нельзя сравнить ни по объему, ни по функциональности. Также нужно учесть отсутствие у студентов первого курса культуры управления проектом, умения разделить работу на независимые части небольшого размера.

2. Если зафиксировать большее число параметров, что позволило бы упростить проверку выполненных работ и частично ее автоматизировать, например, зафиксировать язык и структуры данных, вести общение напрямую между студентами и преподавателем без введения руководящих ролей, студенты воспримут это как «обычное» домашнее задание, которое надо «сдать», будут искать в интернете модули прошлых лет, просить написать модули более сведущих в программировании студентов. В предложенной ситуации последнее не имеет смысла, так как отчитываются они не перед преподавателем, а перед руководителями команды, которые заинтересованы в создании продукта, а не в «сдаче» обязательного задания.

Таблица 2. Граничный объект задания на создание системы компьютерной алгебры: имена модулей и связи между ними

	Целые числа ($b, n; A[...]$) — знак числа (1 — минус, 0 — плюс) номер старшей позиции и массив цифр	Имена	Базовые модули, на которые ссылается данный модуль
Z-1	Абсолютная величина числа, результат — натуральное	ABS_Z_N	
Z-2	Определение положительности числа (2 — положительное, 0 — равное нулю, 1 — отрицательное)	POZ_Z_D	
Z-3	Умножение целого на (-1)	MUL_ZM_Z	
Z-4	Преобразование натурального числа в целое	TRANS_N_Z	
Z-5	Преобразование целого неотрицательного числа в натуральное число	TRANS_Z_N	
Z-6	Сложение целых чисел	ADD_ZZ_Z	POZ_Z_D, ABS_Z_N, COM_NN_D, ADD_NN_N, SUB_NN_N, MUL_ZM_Z
Z-7	Вычитание целых чисел	SUB_ZZ_Z	POZ_Z_D, ABS_Z_N, COM_NN_D, ADD_NN_N, SUB_NN_N, MUL_Z- Z
Z-8	Умножение целых чисел	MUL_ZZ_Z	POZ_Z_D, ABS_Z_N, MUL_NN_N, MUL_Z- Z
Z-9	Частное от деления большего целого числа на меньшее или равное натуральное число с остатком (делитель отличен от нуля)	DIV_ZZ_Z	ABS_Z_N, POZ_Z_D, DIV_NN_N, ADD_1N_N
Z-10	Остаток от деления большего целого числа на меньшее или равное натуральное число с остатком (делитель отличен от нуля)	MOD_ZZ_Z	DIV_ZZ_Z, MUL_ZZ_Z, SUB_ZZ_Z, MUL_Z- Z

Таблица 3. Граничный объект задания на создание системы компьютерной алгебры: имена модулей и связи между ними

	Рациональные числа (дроби) — пара (целое; натуральное), первое имеет смысл числителя, второе — знаменателя	Имена	Базовые модули, на которые ссылается данный модуль
Q-1	Сокращение дроби	RED_Q_Q	ABS_Z_N, GCF_NN_N DIV_ZZ_Z
Q-2	Проверка на целое, если рациональное число является целым, то «да», иначе «нет»	INT_Q_B	
Q-3	Преобразование целого числа в дробное	TRANS_Z_Q	
Q-4	Преобразование дробного числа в целое (если знаменатель равен 1)	TRANS_Q_Z	
Q-5	Сложение дробей	ADD_QQ_Q	LCM_NN_N, MUL_ZZ_Z, ADD_ZZ_Z
Q-6	Вычитание дробей	SUB_QQ_Q	LCM_NN_N, MUL_ZZ_Z, SUB_ZZ_Z
Q-7	Умножение дробей	MUL_QQ_Q	MUL_ZZ_Z
Q-8	Деление дробей (делитель отличен от нуля)	DIV_QQ_Q	MUL_ZZ_Z

Таблица 4. Граничный объект задания на создание системы компьютерной алгебры: имена модулей и связи между ними

	Многочлен с рациональными коэффициентами, m — степень многочлена и массив коэффициентов	Имена	Базовые модули, на которые ссылается данный модуль
P-1	Сложение многочленов	ADD_PP_P	ADD_QQ_Q
P-2	Вычитание многочленов	SUB_PP_P	SUB_QQ_Q
P-3	Умножение многочлена на рациональное число	MUL_PQ_P	MUL_QQ_Q
P-4	Умножение многочлена на x^k	MUL_Pxk_P	
P-5	Старший коэффициент многочлена	LED_P_Q	
P-6	Степень многочлена	DEG_P_N	
P-7	Вынесение из многочлена НОК знаменателей коэффициентов и НОД числителей	FAC_P_Q	ABS_Z_N, TRANS_Z_N, LCM_NN_N, GCF_NN_N, TRANS_N_Z, DIV_ZZ_Z
P-8	Умножение многочленов	MUL_PP_P	MUL_PQ_P, MUL_Pxk_P, ADD_PP_P
P-9	Частное от деления многочлена на многочлен при делении с остатком	DIV_PP_P	DIV_QQ_Q, DEG_P_N, MUL_Pxk_P, SUB_PP_P, ADD_PP_P
P-10	Остаток от деления многочлена на многочлен при делении с остатком	MOD_PP_P	DIV_PP_P, MUL_PP_P, SUB_PP_P
P-11	НОД многочленов	GCF_PP_P	DEG_P_N, MOD_PP_P
P-12	Производная многочлена	DER_P_P	
P-13	Преобразование многочлена — кратные корни в простые	NMR_P_P	GCF_PP_P, DER_P_P, DIV_PP_P

6. ОБЗОР РЕЗУЛЬТАТОВ ВЫПОЛНЕННЫХ РАБОТ

Эксперимент проходил в течение 2015–2016 и 2016–2017 учебных годов со студентами первого курса во втором семестре на факультете компьютерных технологий и информатики СПбГЭТУ «ЛЭТИ». В эксперименте каждый год участвовали около 200 человек (два потока по 6 групп в каждом). Для работы студенты, как правило, выбирали язык, которому их обучали в первом семестре, это С и С++. В то же время, малые группы из 1–6 человек (таких групп в 2016–2017 учебном году было 5) для работы выбирали языки, которыми они хотели овладеть: Java, Python, С#, JavaScript. В качестве структур данных архитекторы выбирали массивы, динамические массивы и списки. Команды студентов, которые не были сильны в программировании, выбирали консольные интерфейсы для пользователя, небольшие группы создавали более удобные интерфейсы, хотя специальных требований к интерфейсу преподавателем не выдвигалось. В то же время, студенты не подумали о том, как сделать систему более практичной, например, о том, чтобы передавать результаты между разными модулями.

В основном, студенты оценивают работу как очень полезную и интересную (70–80%), практически никто не считает её ненужной. Особенно ценят то, что решали задачу совместно.

6.1. Пример анализа работ ответственным по качеству (пример межгруппового объединения)

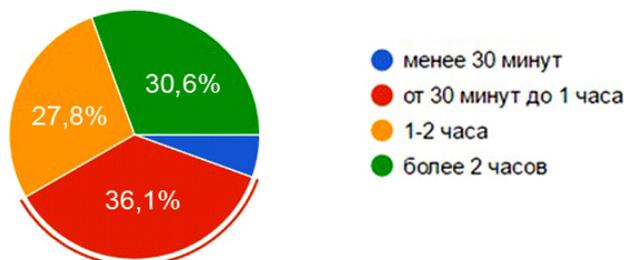
«После того как проект был выполнен, участникам необходимо было пройти анонимное анкетирование, в ходе которого можно было определить, поняли ли студенты суть задания, сколько времени было потрачено на разработку своего модуля, удовлетворены ли студенты работой команды.

Анкета была разработана в Google Forms, состояла из 8 вопросов (3 из которых были не обязательны, связано это с тем, что анкета была общая, как для обычных разработчиков, так и для архитекторов и ответственных за управление качеством).

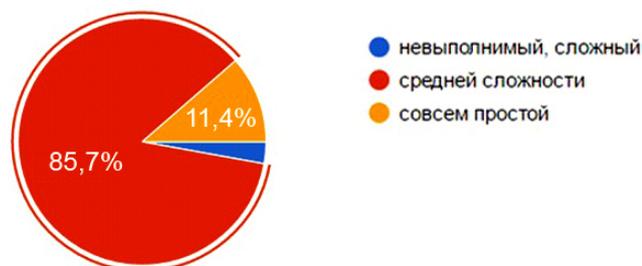
Переходим к результатам анкетирования (рис. 1, 2, 3).

Из 41 разработчика ответили на вопросы анкеты 36 человек. Предполагаем, что это связано с нехваткой времени у студентов.

Сколько времени вы потратили на выполнение своего модуля?
(36 ответов)



Модуль, который Вы выполнили, оказался для Вас: (35 ответов)



Легко ли Вам было разобраться в модулях других ребят нашей группы?
(36 ответов)

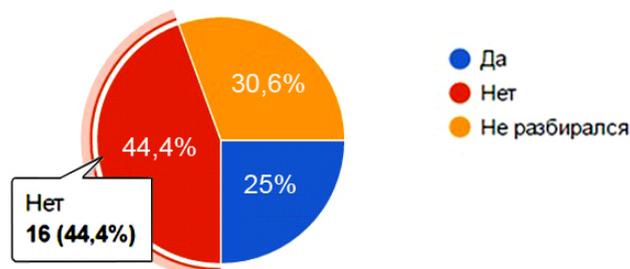
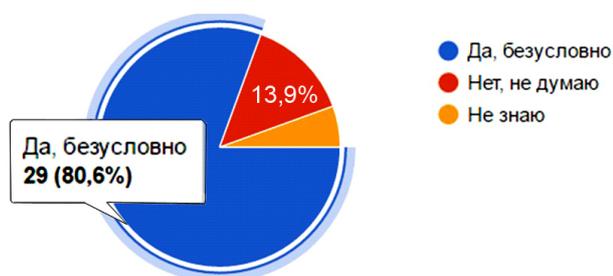
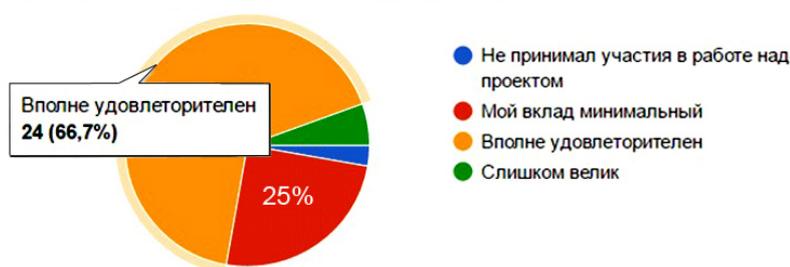


Рис. 1. Пример ответов на вопросы анкеты в 2015–2016 учебном году (команда состоит из двух групп)

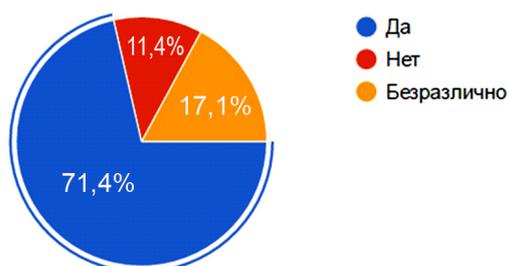
Была ли для вас полезна работа над данным проектом (36 ответов)



Как вы оцениваете ваш личный вклад в данный проект? (36 ответов)



Удовлетворены ли вы работой архитекторов? (35 ответов)



Удовлетворены ли вы работой ответственных за управление качеством? (35 ответов)

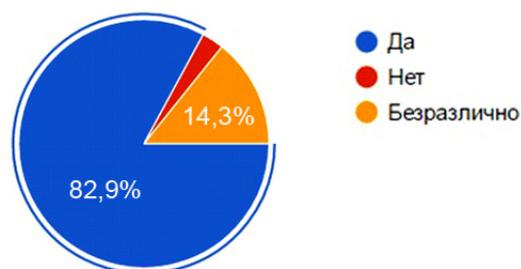


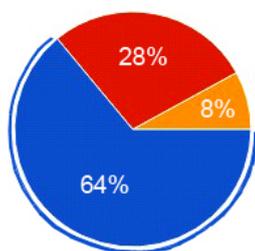
Рис. 2. Пример ответов на вопросы анкеты в 2015–2016 учебном году (команда состоит из двух групп) (продолжение)

На основе анкетирования можно сделать следующие выводы:

- 1) модули среди учащихся были распределены верно;
- 2) данная работа над проектом была полезна учащимся наших групп;
- 3) основная часть времени ушла не на разработку самих модулей, а на сборку программы, поиск ошибок и их исправление;
- 4) проблема нашей команды была в том, что только 9 учащихся смогли с лёгкостью разобраться в модулях других ребят из нашей команды».

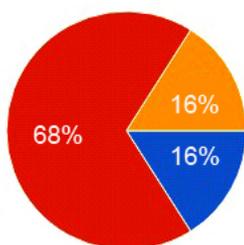
Хорошо ли была организована работа?(Справился ли архитектор со своими обязанностями?)

(25 ответов)



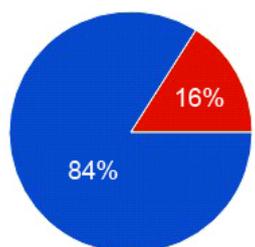
- Отлично , все задания были грамотно и вовремя розданы студ...
- Хорошо , все задания были розданы вовремя , но не всегда п...
- Удовлетворительно , все задания были розданы с незначительной...
- Плохо , задания были розданы не полностью с значительной задер...
- Ужасно , задания не были розданы.

Сколько времени ушло на работу? (25 ответов)



- Мало , код был написан за один вечер
- Средне , код был написан за 2-3 дня
- Много , код писался неделю с сильным затруднением

Хорошо ли вы поняли смысл общей работы? (25 ответов)



- Да , работать в команде интересно , это помогло мне набрать необходимый опыт
- Средне , работа в команде была интересной , но я мало что вынес для себя
- Плохо , работы в команде не было как таковой

Рис. 3. Пример ответов на вопросы анкеты в 2016–2017 учебном году (команда состоит из одной группы)

Из наблюдений за работой команд можно выделить следующие:

- 1) увеличение на втором году эксперимента команд с небольшим числом человек и даже маргинальных из одного человека, отсутствие на втором году межгрупповых объединений, которые были в первый год эксперимента;
- 2) неправильные организационные решения внутри команд, которые приводили к таким последствиям, как:
 - смена архитектора,
 - отпочкование несогласных в отдельную группу,
 - деление на подгруппы по реальному уровню владения программированием,
 - неучастие части студентов в эксперименте из-за того, что не смогли вписаться в команду (около 10% студентов);
- 3) увлечение некоторых команд работой и создание интересных продуктов с удобным интерфейсом и расширенным набором функций, превышающим исходное задание.

7. ВЫВОДЫ

Использованные в рамках эксперимента формы проектно-командной деятельности отличаются от описанных в литературе форм и не попадают в сложившиеся классификации (см., например, [9]). Размер команды заранее не фиксирован, работа в командах строится на формализации задания, которое по доверенности остальных студентов поручается одному из них — архитектору. Таким образом, при малом количестве инициативных студентов команды могут состоять из 15–20 человек, при большем числе студентов, которые хотят видеть себя в роли архитекторов, число команд увеличивается, а число человек в команде уменьшается до 5–7 человек и может в крайних случаях достигать 2–3 и даже одного человека, который хочет сделать всю работу сам. Вторым организатором является студент, отвечающий за качество выполнения работ. В его обязанности входит правильное распределение работ, чтобы все участники могли работать с пользой и удовольствием. Умение выполнять сложную работу путем разделения ее на маленькие части, а затем соединения этих частей в работающий продукт рассматривается как важная цель учебной работы.

Возможность использования студенческой инициативы для коллективного создания системы компьютерной алгебры во многом базировалась на правильном выборе «граничного объекта» — технического задания с описанием продукта, который должен был быть получен в результате командной деятельности, где были жестко заданы названия программных модулей и связи между ними, но не фиксировался ни язык программирования, ни структура данных, ни пользовательский интерфейс.

Список литературы

1. Программирование и математика (2008) <https://habrahabr.ru/post/37217/> (дата обращения: 12.01.2017).
2. Насколько программисту нужно знание математики в работе (2015). <https://tproger.ru/experts/2/> (дата обращения: 12.01.2017).
3. Насколько программисту нужно знать математику? Что думают в Яндексе (2014). <https://habrahabr.ru/company/yandex/blog/239339/> (дата обращения: 12.01.2017).
4. *Городня Л.В., Калинина Н.А.* (2000) Компьютерная алгебра и функциональное программирование как базовые компоненты научного инструментария // В тр. межд. конференции «Информационные технологии в науке, образовании, телекоммуникации, бизнесе». Украина, Крым, 2000 г. С. 81–84.
5. *Калинина Н.А.* (2002) Системы компьютерной алгебры как необходимая компонента образовательных технологий. Всероссийская научная конференция «Научный сервис в сети Интернет» (2002 г.). Материалы к докладу: http://www.ict.edu.ru/vconf/index.php?a=vconf&c=getForm&r=thesisDesc&d=light&id_sec=26&id_thesis=396 (дата обращения: 12.01.2017).
6. *Люблинская И.Е.* (2013) TI-Nspire™ CAS технологии для изучения математики и основных научных дисциплин // Компьютерные инструменты в образовании. 2013. № 1. С. 15–21.
7. Wolfram Mathematica: Modern Technical Computing <http://www.wolfram.com/> (дата обращения: 12.01.2017).
8. MathPartner <http://mathpar.com/ru/> (дата обращения: 12.01.2017).
9. *Полат Е.С., Бухаркина М.Ю.* Новые педагогические и информационные технологии в системе образования. М.: Издательский центр «Академия», 2008.
10. *Боков Л.А., Катаев М.Ю., Поздеева А.Ф.* Технология группового проектного обучения в вузе как составляющая методики подготовки инновационно-активных специалистов // Современные проблемы науки и образования. 2013. № 6.; URL: <https://science-education.ru/ru/article/view?id=11762> (дата обращения: 12.01.2017).

11. Коваленко Ю.А., Никитина Л.Л. Проектная деятельность студентов в образовательном процессе вуза // Вестник Казанского технологического университета. 2012. № 20. Т. 15. С. 229–231.
12. PRAZE <http://peerreview.cis.unimelb.edu.au/tools/about-praze/> (дата обращения: 12.01.2017).
13. Mulder R.A., Pearce J.M. PRAZE: Innovating teaching through online peer review. ... Proceedings ascilite Singapore 2007: <http://ascilite.org/conferences/singapore07/procs/mulder.pdf> (дата обращения: 12.01.2017).
14. Поздняков С.Н., Рыбин С.В. Дискретная математика: учебник для вузов. М.: Академия, 2008.
15. Bannon L., Bodker S. Constructing Common Information Spaces // Fifth European Conference on Computer Supported Cooperative Work, ECSCW'97, Lancaster, UK, 7–11 September 1997, Proceedings. Kluwer Academic Publishers, 1997. P. 81–96.

Приложение.

Информация для студентов на сайте курса дискретной математики

Цель работы — создать систему компьютерной алгебры (на длинных числах и многочленах одной переменной), распределив работу между всеми студентами.

При этом:

- система должна быть создана качественно и быстро;
- в проекте должны участвовать все студенты, и каждый должен хорошо разбираться во всей выполненной работе.

Для достижения поставленной цели создаются группы студентов, работу в которых обеспечивают два человека: архитектор и ответственный за управление качеством, которые сами модулей не разрабатывают.

Архитектор должен:

- хорошо разбираться в программировании, структурах данных и управлении проектами по созданию программного обеспечения,
- выбрать структуру данных, язык программирования, систему поддержки версий и способ взаимодействия всех исполнителей (например, GitHub),
- уточнить интерфейс для взаимодействия модулей и определить вид пользовательского интерфейса,
- определить требования к читаемости отдельных модулей и технологии тестирования (или к доказательству корректности работы каждого модуля их авторами). Например, архитекторы могут решить, что тестирование должно иметь несколько уровней, низший уровень — тестирование каждого модуля, используя заглушки для тех, которые этот модуль используют.

Ответственный за управление качеством обеспечивает контакт студентов с архитектором, так чтобы каждый студент группы работал с пользой для себя.

По результатам работы ответственный за управление качеством создаёт в Google Forms анонимную анкету, на которую отвечают все студенты группы, и которая даёт ответ, хорошо ли удалось распределить работу между студентами: сколько времени ушло на работу (чем меньше, тем лучше), хорошо ли поняли смысл общей работы, также ответственные (все вместе) определяют метрики, по которым будет оцениваться качество ПО.

Искусство управления проявится в том, чтобы не распределять работу случайно, но и не отдавать интересные её части тем, для кого решение этих задач не является «вызовом».

Для выполнения проекта группы могут объединяться или делиться, чтобы использовать один язык и одну систему программирования. Чем больше человек будет участвовать в создании одной системы, тем лучше (при условии, что это сократит время разработки, но не уменьшит вовлечённость каждого студента в проект).

Приёмка разработанных программ, требования и регламент приёмки:

- модули должны быть хорошо написаны и прокомментированы;
- внутри каждого модуля должны быть фамилии создателей с указанием групп студентов;
- каждый модуль должен быть как можно более коротким за счёт ссылок на программы других участников.

В качестве одного из параметров ответственный за управление качеством вычисляет и потом представляет медиану, моду и среднее объема кода 80% всех модулей, упорядоченных по возрастанию объема кода. В качестве другого параметра — число вызовов чужих модулей из каждого модуля, при этом также требуется привести медиану, моду и среднее).

Архитекторы должны договориться, как считать объем кода, чтобы это не привело к снижению читаемости программ, введению несодержательных идентификаторов и пренебрежению комментариями.

Созданная система должна иметь интуитивно понятный интерфейс и справку для работы с системой, которая компенсирует «неинтуитивность» пользовательского интерфейса. На работу отводятся две недели, третья неделя идёт на перекрестное рецензирование (все проекты выкладываются в открытый доступ, и архитекторы одних проектов вместе с любыми желающими из их команды рецензируют проект по указанным выше параметрам); на четвертой неделе архитекторы вместе с ответственными за управление качеством и двумя случайно выбранными (преподавателем) студентами каждой из групп защищают проект.

На защите могут присутствовать все желающие; на защиту даётся 20 минут (10 минут — демонстрация системы 2–3 студентами, выбранными случайно, 5 минут — демонстрация результатов анкетирования ответственным за управление качеством, 5 минут — вопросы оппонентов и ответы администраторов проекта).

В таблицах приведены модули, которые обязательно должны быть в каждой системе (см. табл. 1, 2, 3, 4 в тексте статьи). Приветствуется расширение этой системы (например, добавление разложения числа на множители по методу Ферма).

Просьба к архитекторам: по ходу работы присылать данные по новым модулям, которые будут добавлены в таблицу как дополнительные: имена и ссылки на другие модули (имя определяет первый приславший письмо архитектор). Также можно сообщать об опечатках/ошибках в таблице.

Численная оценка работ определяется по 5-балльной системе закрытым голосованием всех участников (кроме преподавателя) по двум параметрам:

- качество созданной системы;
- качество участия студентов в создании системы.

Поступила в редакцию 12.01.2017, окончательный вариант — 16.03.2017.

Computer tools in education, 2017

№ 2: 25–41

<http://ipo.spb.ru/journal>

COMPUTER ALGEBRA SYSTEM AS A PEDAGOGICAL TASK

Pozdniakov S.N.¹

¹SPbETU «LETI», Saint-Petersburg, Russia

Abstract

The experience of organizing a student group project for developing components of a computer algebra system as one of the types of educational activity in a course of discrete mathematics is presented in this article; it is related to algorithms for long integer and polynomial arithmetic. A special feature of the organization of this work is a targeted selection of a "boundary object which is the basis of the common information space.

As boundary object was chosen the structure of the technical assignment for the development of the computer algebra system, in which the names of the modules and the relations between them were fixed, but neither the language nor the data structure were fixed. The use of such a boundary object for the organization of a common information space, on the one hand, provided sufficient freedom for student groups to make decisions on the architecture of the system being created, on the other hand, accurately determined the parameters of the work performed, which allowed to compare the results of the work of different groups and assess the quality of the work performed and the quality of the organization of the joint activities in the groups.

The results of the work and students' written evaluations showed that this educational method gives good results as far as the quality of the completed projects is concerned and is highly appreciated by the students themselves, justifying their expectations from studying at a technical university.

Keywords: *computer algebra system, teaching in mathematics, connections between mathematics and informatics, modern technologies of training.*

Citation: S.N. Pozdniakov, "Sistema komp'yuternoï algebry kak pedagogicheskaya zadacha" [Computer Algebra System as a Pedagogical Task], *Computer tools in education*, no. 2, pp. 25–41, 2017 (in Russian).

Received 12.01.2017, the final version — 16.03.2017.

Sergei N. Pozdniakov, professor of Department of Mathematics, Saint-Petersburg Electrotechnical University; 197376 Saint-Petersburg, ul. Professora Popova 5, korp. 3, Dep. of Mathematics, pozdnkov@gmail.com

**Поздняков Сергей Николаевич,
доктор педагогических наук,
профессор кафедры ВМ-2 СПбГЭТУ «ЛЭТИ»;
197376, Санкт-Петербург, ул. Профессора
Попова, д. 5, корп. 3, кафедра ВМ-2,
pozdnkov@gmail.com**

©

Наши авторы, 2017.

Our authors, 2017.