8. Абрашина-Жадаева Н. Г., Тимощенко И. А. Конечно-разностные схемы для уравнения диффузии с производными дробных порядков в многомерной области // Дифференциальные уравнения. 2013. Т. 49. № 7. С. 819-825.

9. Абрашина-Жадаева, Н.Г., Тимощенко И.А. Многокомпонентные методы для аномальных процессов диффузии // Труды 1--ого международного семинара АМАДЕ-2021, 2022. --- С.~ 5--11.

10. Abrashina-Zhadaeva N.G., Romanova N.S.. A splitting type algorithm for numerical solution of PDEs of fractional order. // Mathem. Modeling and analysis. V. 12(4) (2007) 399-408.

11. Абрашина-Жадаева, Н.Г. Совершенствование численных методов расчета задач математической физики с уравнением влагопереноса. // Материалы международного конгреса по информатике: информационные системы и технологии (CSIST'2022), 2022, Ч.2.

НЕКОТОРЫЕ МЕТОДИЧЕСКИЕ ВОПРОСЫ ПРОВЕДЕНИЯ РАЗЛИЧНЫХ ВИДОВ ПРАКТИКИ ПО АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЮ

Аленский Н. А.

Белорусский государственный университет, г. Минск

В 2023 / 2024 учебном году произошли изменения в преподавании алгоритмизации и программирования на первом и втором курсах механико-математического факультета БГУ. В докладе рассматриваются методические особенности проведения различных видов практики по специальности «Математика и компьютерные науки» на **первом курсе** в связи со следующими изменениями. ***Во-первых***, сохранив лабораторные работы и лекции в том же объеме по дисциплине «Методы программирования», по новому учебному плану добавили дисциплину «Практикум по программированию», относящуюся к дополнительным видам обучения компоненты учреждения образования объёмом 2 часа в неделю, для которой лекции не предусмотрены. ***Во-вторых***, учебная (вычислительная) практика объемом 72 часа, которая долгие годы проводилась по алгоритмизации и программированию в течении первых двух семестров по 2 часа в неделю, с прошлого учебного года проводится после летней сессии две недели по 6 часов ежедневно. Поэтому возникает вопрос, как эффективно согласовать эти три вида практических занятий, чтобы не повторять одно и то же. Более того, в одной группе (подгруппе) их не обязательно проводит один и тот же преподаватель.

Все задания по программированию можно разделить на **два больших класса**: «задачи-программы» и упражнения [1]. В ***первом*** из них надо составить алгоритм и программу на выбранном языке, отладить и протестировать её. Этот класс задач является более важным, и его необходимо чаще практиковать, особенно для закрепления одной или нескольких тем и при выполнении индивидуальных заданий.

В ***упражнениях*** требуется записать один или несколько вариантов элемента языка (выражение, оператор, заголовок или вызов функции) или части программы и (или) проанализировать их. При анализе программы (или её части) надо ответить на ряд вопросов, например, есть ли ошибки в написанном коде, объяснить их, всегда ли ошибки будут проявляться, как повлияет на результат какое-нибудь изменение, а если ошибок нет, что получим после выполнения кода. На начальном этапе изучения сложных тем нельзя игнорировать такие упражнения, которые играют подготовительную, вспомогательную роль и особенно полезны при работе с отстающими студентами. Одним из этапов обучения и важным условием умения составлять качественные программы является понимание готовых программ, умение их «читать». Поэтому

полезно выполнять упражнения на восстановление постановки задачи, для которой записана программа или её часть. В случае затруднения можно рекомендовать исполнить её для одного или нескольких тестов, или составить блок-схему. Необходимо обратить внимание также на такие упражнения, в которых требуется исследовать программу или её часть, элемент языка программирования, выбрать наилучший из правильных вариантов или записать, если можно, более эффективный вариант.

Частным случаем упражнений являются ***тесты***, которые полезны не только для контроля и оценки знаний, но и как эффективный дидактический материал для изучения программирования. В практической части соответствующего курса ***Moodle*** по всем темам дисциплины «Методы программирования» кроме контрольного электронного тестирования [2] подготовлены тренировочные упражнения и тесты, которые используются во время любых практических занятий или предлагаются в качестве домашних заданий.

Для более объективной оценки выполненных работ рекомендуется давать задания нескольких **уровней алгоритмической сложности**, что очень важно при контроле и оценке знаний. Например, по теме «Матрицы» можно предложить следующие задачи: ***первый уровень*** с максимальной оценкой шесть баллов — во всей целочисленной матрице найти количество чисел с каким-нибудь простым условием (например, кратных заданному числу); ***второй уровень*** (на восемь баллов) — в каждой строке найти количество чисел с более сложным условием, например, простых; ***третий уровень*** (десять баллов) — обойти квадратную матрицу по спирали, то есть вывести её в таком порядке. На одном уровне можно предусмотреть два ***подуровня***. Например, в теме «Строки» второго уровня сложности условие задания может быть таким: в строке найти один любой символ [все символы], который повторяется [которые повторяются] чаще других.

Чтобы не переключаться с одного уровня сложности на другой, почти по каждой теме можно сформулировать **требования к заданиям.** Например, по теме «Матрицы» во втором семестре можно потребовать следующее.

1) Матрица должна быть динамической.

2) Составить несколько функций: для ввода, вывода матрицы и в зависимости от варианта и уровня алгоритмической сложности одну или несколько функций для её обработки. Если можно, составить функцию для работы с одномерным массивом и использовать её для матрицы. В main должны быть необходимые объявления, вызов функций и минимум вспомогательных операторов.

3) Там, где это можно, использовать указатели для организации циклов.

4) Перестановки строк матрицы, если это надо по условию, выполнять с помощью указателей.

5) Матрицу вводить с экрана, используя управление курсором.

6) Предусмотреть вывод результатов в удобном для анализа виде, разными цветами (например, каждое простое или наибольшее число, или строку (столбец) с каким-нибудь условием вывести другим цветом), а полученные для каждой строки параметры вывести справа от матрицы.

В докладе показано, как учитывать выполнение или невыполнение такого рода требований при оценке заданий. Кроме этого, оценка зависит от эффективности алгоритма и программы, качества тестирования, своевременности выполнения задания или времени, затраченного на его выполнение во время контрольной работы, зачета или экзамена. В ***Moodle*** за этим легко проследить. С учетом всего этого при необходимости задания по некоторым темам можно оценить и по 20-тибалльной шкале.

При **выборе заданий** по алгоритмизации и программированию для любого из трех перечисленных в начале тезисов видов практик необходимо учитывать, что во время

практических занятий важно не только и не столько изучить правила конкретного языка программирования и обучить школьника записывать с его помощью алгоритм. Для этого есть Интернет, учебные пособия в ***Elib*** и электронный материал в ***Moodle.*** Прежде всего, и это самое главное, предлагаемые задания должны способствовать развитию ***алгоритмического, логического, математического, программистского стилей мышления***, помочь овладеть методами и приемами эффективного программирования. С помощью задач студент должен освоить основные типы алгоритмов (ветвления, циклы с известным и неизвестным количеством повторений, вложенные циклы, итерационные и рекурсивные алгоритмы), научиться использовать и обрабатывать данные различных типов и структур (целые, вещественные, логические, символьные, строки, массивы, структуры, списки, классы, файлы). Сказанное выше можно использовать при проведении любых из трёх видов перечисленных в начале практик. Рассмотрим кратко их особенности.

Лабораторные занятии проводятся параллельно с лекциями по дисциплине «Методы программирования» и должны на практике закрепить лекционный материал. Опыт работы показал, что при изучении некоторых тем, не сложных теоретически, можно нарушить классическую схему «Лекция, а затем практика». Сначала во время лабораторного занятия предлагается выполнить общее или индивидуальное задание, возможно, с помощью преподавателя, а позже послушать лекцию или, так как их немного, ограничиться только практикой, а с теорией ознакомиться самостоятельно, используя электронные материалы.

При изучении дисциплины **«Практикум по программированию»** по учебному плану не предусмотрены лекции. Поэтому она мало чем отличается от лабораторных занятий. Если лабораторные работы закрепляют возможности языка программирования (операции, операторы, функции, массивы, строки, структуры, классы, списки, файлы), то с помощью этой дополнительной дисциплины появилась возможность больше решать на компьютерах ***математические задачи*** изучаемых на первом курсе соответствующих дисциплин. Вот некоторые их типы: работа с массивом точек и геометрическими фигурами на плоскости или в пространстве; задачи целочисленной арифметики, работа с векторами, решение системы линейных алгебраических уравнений, вычисление определителей и другие матричные задачи из алгебры; вычисление рядов Тейлора из математического анализа; решение уравнений, вычисление определенных интегралов из численных методов и другие.

Для студентов специальностей, которые курирует кафедра веб технологий и компьютерного моделирования, **учебная (вычислительная) практика** полвека проводится по алгоритмизации и программированию. К её началу в конце июня студенты не только изучили дисциплины «Методы программирования» и «Практикум по программированию», выполнили практические задания и, более того, сдали зачет и (или) экзамен. Поэтому для лучшего закрепления изученного материала имеет смысл давать задания не столько по конкретным темам, а предлагать в каждом проекте использовать несколько различных структур данных и методов программирования для решения одной и той же задачи, а в отчете провести их сравнительный анализ. Например, для решения ***матричных задач*** можно использовать статические (обе размерности — константы), частично динамические, или динамические двумерные массивы; однонаправленные списки, в информационной части каждого элемента которых одномерный массив; текстовые или бинарные файлы. При программировании учебных задач ***целочисленной двоичной или шестнадцатеричной арифметики*** можно предложить использовать арифметические операции целочисленного деления, битовые операции, объединения и поля битов. В докладе более подробно рассматриваются эти и другие примеры заданий.

Литература

1. Аленский Н. А., Травин В. В. Методика преподавания информатики. Уч.-мет. пособие с грифом УМО вузов РБ по естественно-научному образованию. – Минск, «Адукацыя і выхаванне». 2019 --- 104 с.
2. Аленский Н. А. Система контроля и оценки знаний по методам программирования на ММФ БГУ с использованием образовательного портала *MOODLE*. Проблемы преподавания высшей математики и информатики в условиях новой образовательной парадигмы: материалы Междунар. науч.-практ. конференции, Минск, 14–15 апреля 2022 г. / БГУ, Механико-математ. фак. ; [редкол.: С. А. Самаль (отв. ред.) и др.]. – Минск : БГУ, 2022. – 146 с.: ил., табл. – Библиогр. в тексте, стр 4 — 6 http://elib.bsu.by/handle/123456789/277837.

О НЕОБХОДИМОСТИ МОДИФИКАЦИИ КУРСА МАТЕМАТИКИ ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Асмыкович И. К.

Белорусский государственный технологический университет, Минск

Live as if you were to die tomorrow.
Learn as if you were to live forever
Gandhi

Переход на следующий этап технологической революции во всём мире требует нового подхода к уровню образования субъектов хозяйствования, особенно инженерно-технического и руководящего персонала. Ясно, что математика в этом подходе должна быть не на последних ролях. Времена, когда математику представляли только в чисто технико-технологическом плане, в виде востребованного обществом инструмента его практически-преобразовательной деятельности, ушли в прошлое [2,3]. В современную эпоху резко возросла потребность в креативной и интеллектуально развитой личности. Разумеется, что наряду с другими компетенциями она должна обладать и отвечающими требованиям нашей эпохи компетенциями в области математики: даже в повседневности сегодня практически трудно без них обойтись, хотя в реальности часто обходятся и приходят к довольно печальным результатам. Математика ставит проблемы, решение которых требует усилий мысли, упорства, воли и других качеств личности.

К сожалению в настоящее время реальные преобразования типовых и учебных программ среднего и высшего специального образования не совсем соответствуют высказанным идеям о необходимости фундаментального образования [3]. По всем инженерным специальностям существенно уменьшают объёмы учебных часов по математическим дисциплинам. И большая часть современных инженеров не знает ту математику, которая им нужна. Их учат по учебникам непрерывной математике, что было нужно инженеру 60-40 лет назад, но с тех пор всё существенно изменилось: другие области, другое применение. Отметим, что целый ряд, весьма необходимых для высшего образования инженеров, разделов математики отсутствуют в современных учебных планах [4]. Ранее, для ряда инженерных специальностей был отдельный курс «Методы оптимизации» или «Математическое программирование». Л. Эйлер, великий математик, писал: «Так как здание всего мира совершенно и возведено премудрым Творцом, то в мире не происходит ничего, в чем не был бы виден смысл какого-нибудь максимума или минимума». Составители учебных программ для высшего технического образования убирают из курса высшей математики задачи на условный экстремум, метод