

РАЗДЕЛ IV ТЕХНОЛОГИИ ПРЕПОДАВАНИЯ ДИСЦИПЛИН КОМПЬЮТЕРНОЙ МАТЕМАТИКИ И МОДЕЛИРОВАНИЯ В ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ

УДК 378.147

О ТЕСТОВЫХ ЗАДАНИЯХ КАК ДИДАКТИЧЕСКИХ ИНСТРУМЕНТАХ В КУРСАХ ДИСЦИПЛИН КОМПЬЮТЕРНЫХ НАУК

Н. А. Аленский¹⁾, Д. В. Филимонов²⁾

*¹⁾Белорусский государственный университет, Беларусь, Минск,
Alensky@bsu.by,*

*²⁾Белорусский государственный университет, Беларусь, Минск,
FilimonovDV@bsu.by*

На примере преподавания методов программирования для студентов естественнонаучных специальностей ММФ БГУ показано использование упражнений и тестов для контроля и оценки знаний в качестве дидактического материала для объяснения и закрепления наиболее сложных тем или вопросов на практических занятиях различных видов, на лекциях и в качестве домашних заданий.

Ключевые слова: тренировочные упражнения и тесты; репетиционное, контрольное тестирование.

ON TESTING AS DIDACTING TOOL IN COMPUTER SCIENCE COURSES

N. A. Alensky¹⁾, D. V. Filimonov²⁾

*¹⁾Belarusian State University, 4, Niezavisimosti pr., Minsk, Belarus, 220030,
Alensky@bsu.by,*

*²⁾Belarusian State University, 4, Niezavisimosti pr., Minsk, Belarus, 220030,
FilimonovDV@bsu.by*

The example of teaching programming methods to students of natural science specialties at the MMF BSU shows the use of exercises and tests for monitoring and assessing knowledge as didactic material for explaining and reinforcing the most complex topics or questions in practical classes of various types, in lectures and as homework.

Keywords: training exercise and tests; rehearsal, control testing.

Введение

Описываемый авторский подход к тестированию, апробирований в течение 4 лет при обучении основам программирования студентов специальностей «Научно-производственная деятельность» и «Компьютерная математика и системный анализ» на механико-математическом факультете БГУ показал, что использование упражнений и тестов полезно не только для контроля и оценки знаний, но и в качестве дополнительного дидактического материала. Такой вывод сделан на основе преподавания методов дискретной математики, алгоритмизации и программирования для студентов старших курсов.

Актуальность темы связана, во-первых, с тем, что методы программирования на первом курсе механико-математического факультета (ММФ) БГУ изучаются на примере языка C++ в достаточно сложной профессиональной среде MS Visual Studio. Вторая причина заключается в том, что в заимствованном студентами из Интернета и сгенерированном ИИ программном продукте не всегда выполняются требования и ограничения к заданиям, которые по каждой лабораторной работе предусмотрены в соответствии с учебным планом и программой. В-третьих, уделить достаточно времени действительно важным, творческим или исследовательским задачам математического моделирования не представляется возможным в рамках практических и лабораторных занятий, поскольку они предполагают освещение типовых задач, умение решать которые более приоритетно.

Использование упражнений и тестов на лекциях

Приведем несколько основных причин, которые заставляют вносить коррективы в традиционную классическую методику чтения лекций:

1) На факультете большое внимание уделяется подготовке **электронных вариантов лекций**, которые по каждой теме размещены в теоретической части разработанного курса [1] образовательного портала ММФ. Кроме этого, в *elib* есть учебные пособия, в которых приведен как теоретический материал, так и лабораторные работы, упражнения и тесты;

2) Для специальностей «Компьютерная математика и системный анализ» и «Механика и математическое моделирование» ММФ на первом курсе по учебному плану есть только лекции и лабораторные занятия, в отличие от остальных специальностей, где проводятся также занятия по дисциплине «Практикум по программированию»;

3) В программировании есть много вопросов (логические операции, операторы ветвления и цикла, функции и виды их параметров, операции над указателями и другие), которые теоретически можно изложить быстро, но этого недостаточно для понимания контекста их применения в проектировании оптимизированных решений;

4) Мы придерживаемся того, что аудиторная лекция не всегда и не по каждой теме должна полностью дублироваться в электронном конспекте. В аудитории основное внимание необходимо уделить более важным сложным вопросам, а справочный материал и логически простые вопросы, которые подробно изложены в электронном виде, можно изложить предельно кратко, чтобы не потерять связь между ними, или не озвучивать совсем, а лишь объяснить, где их можно найти. Оставшееся время можно использовать для рассмотрения упражнений и тестов по более сложным вопросам.

Таким образом, на некоторых лекциях рекомендуется использовать упражнения и тесты как дополнение к классическому традиционному изложению материала, чтобы понятнее, проще и быстрее объяснить наиболее сложную тему или ответить на возникшие во время лекции вопросы. Часть лекции можно проводить в форме беседы, предупредив студентов об этом заранее. Для подготовки к такой лекции, к выполнению лабораторных работ и к контрольным мероприятиям по всем темам дисциплины «Методы программирования» кроме контрольного тестирования разработаны и размещены во второй практической части соответствующего курса ОП ММФ **тренировочные упражнения и тесты** с открытым доступом для студентов. Их использование на лекции с включенным проектором или интерактивной электронной доской экономит время, вызывает больший интерес у студентов и частично решает проблему посещаемости.

Использование упражнений и тестов на лабораторных занятиях

Все задания по программированию, которые студенты должны выполнить во время практических аудиторных занятий и управляемой самостоятельной работы, можно разделить на **два больших класса** [2]. В **первом** из них надо разработать эффективный алгоритм и программу на выбранном для изучения языке, отладить и протестировать её с использованием компьютера.

Общепринятая методика проведения лабораторных занятий заключается в том, что преподаватель в основном использует только задания такого класса. Выдает их, индивидуально или для всей подгруппы (команды) комментирует размещенные в практической части того же элек-

тронного курса условие задачи, требования и ограничения, помогает, если надо, отлаживать программу и проверяет её.

Есть также *несколько причин*, которые побуждают авторов не ограничиваться во время практических занятий только такими задачами.

Так как аудиторных часов немного, а изучаемый со студентами профессиональный язык C++ не простой и объемный, с помощью только хрестоматийных задач-программ невозможно закрепить на практике все многообразие элементов, возможностей и методов, предусмотренных учебной программой.

Вторая причина заключается в том, что нет гарантии, что студент при выполнении лабораторной работы по разработке программного продукта не воспользовался чужими работами или не обратился к ИИ.

Эти и некоторые другие причины не всегда позволяют объективно оценить знания, умения и навыки студентов – которых современным востребованным молодым специалистам недостаточно. Поэтому при изучении некоторых, не обязательно всех, а более сложных тем нельзя на практических занятиях игнорировать упражнения и тесты [2]. В *тестах* требуется записать один или несколько вариантов элемента языка (объявление, выражение, оператор, заголовок или вызов функции и т. п.) или части программы и (или) проанализировать их. Такие и другие виды упражнений играют подготовительную, вспомогательную роль и особенно полезны при работе с отстающими студентами.

Частным случаем тестов являются *задачи с вариантами ответа*, которым может быть отведена роль вспомогательного дидактического материала. Такое тестирование нацелено не только на проверку знания формальных синтаксических правил того или иного элемента системы программирования, а прежде всего, понимания использования таких правил при разработке программ, формирования компетенций анализа и синтеза, знакомства с важными проблемами, не рассмотренными в основном курсе. Выполнение каждой такой задачи, по причине их исследовательской природы, не ограничено по времени, но ограничено по числу попыток. В качестве вариантов ответа для таких тестов выбираются значения, достаточно близкие к правильным и возникающие при выполнении подпрограмм в неправильном порядке – то есть обучающиеся могут получить в некоторой мере правдоподобные, но неверные решения. В дальнейшем по желанию преподавателя такие ответы могут сопровождаться комментариями с причинами возникновения ошибки.

В практической части ОП ММФ по всем темам дисциплины «Методы программирования» кроме контрольного электронного тестирования подготовлены тренировочные упражнения и тесты [1]. Они используются во время специальных практических занятий в форме беседы по их

обсуждению, и (или) предлагаются в качестве общих или индивидуальных лабораторных работ. Упражнения и тесты эффективны также и на консультациях. Кроме этого, по некоторым темам для таких занятий разработано электронное **репетиционное тестирование**, размещённое там же. При его использовании в отличие от контрольного у всех студентов одновременно высвечиваются одинаковые вопросы, чтобы можно было при необходимости обсудить с группой ответы на них. Посещаемость таких занятий и активность на них влияют на текущую оценку в большей степени, чем во время занятий по отладке программ.

Контрольное тестирование

Тесты предлагается использовать прежде всего для оценки знаний на контрольных работах, зачётах и экзаменах. Контрольное тестирование для этих целей можно проводить одним из следующих методов: электронное, собеседование и смешанное тестирование.

Электронное контрольное тестирование подготовлено с использованием элемента «Тест» в третьей части того же курса в ОП ММФ для разделов, включающих несколько тем [1]. Полезными являются открытые тесты, в которых приведен элемент языка или фрагмент программы, и с помощью типа вопроса «Короткий ответ» или «Числовой ответ» необходимо определить их результат. Это эффективнее, чем «Множественный выбор» и «На соответствие» с предложенными вариантами ответов. Для проверки теоретических положений удобно использовать такие типы вопросов как «Выбор пропущенных слов», «Выбор слова» и другие.

Тестирование в формате собеседования имеет смысл использовать, когда ответ на вопрос трудно или невозможно формализовать (блок-схемы программ и алгоритмов, теоретические вопросы, ответ с решениями при переводе чисел из одной системы счисления в другую). В этом случае студент в свободной форме письменно или устно отвечает на вопрос с обязательным объяснением.

Смешанное тестирование предполагает оценивание работы, состоящей из выполняемых в аудитории заданий и защиты исследовательской задачи, время на выполнение которой выделяется незадолго до проведения контрольного тестирования, обычно близко к тренировочному. В отличие от ранее упомянутых тестовых исследовательских задач, включенные в данную форму контроля не имеют вариантов ответа, введенное студентом значение сохраняется в базе данных курса и требует сопроводительного объяснения. Это позволяет более объективно оценить степень усвоения материала.

Кроме этого, можно предложить и другие способы **повышения объективности итоговой оценки**:

1) если ответ на вопрос система не в состоянии формально оценить, можно при электронном тестировании использовать тип вопроса «Эссе». Студент размещает файл или текст, который преподаватель проверяет и оценивает вручную;

2) после окончания любого из предложенных выше методов тестирования и получения компьютерной оценки можно устно проверить понимание ответов на тесты – это полезно и в том случае, если ответ учащегося не прошел валидацию системы, либо ему не хватило времени на какой-либо вопрос (например, несколько секунд). Оценка за конкретный тест, а, значит, и итоговая повышается, исключая «штраф» за некорректную форму ответа. И наоборот, оценку можно понизить, если студент не в состоянии объяснить свой ответ;

3) при любом виде контрольного тестирования целесообразно подготовить несколько банков вопросов с большим количеством для разных категорий, из которого вопросы будут выбираться случайным образом. Это удобно и для задач, в которых небольшие изменения начальных условий приводят к значительным изменениям ответов.

Заключение

Использование упражнений и тестов в процессе обучения программированию не только способствует лучшему усвоению материала студентами, но также помогает развивать критическое мышление и навыки анализа. В условиях быстро меняющегося мира технологий, где программирование становится все более важным навыком, необходимо. Применение таких адаптированных активных методов обучения как тестирование в сочетании с решением практических задач позволяет студентам не только запоминать информацию, но и применять ее на практике. Это особенно важно в контексте программирования, где ошибки могут быть неочевидными и требовать глубокого анализа. Студенты учатся не только находить решения, но и обосновывать свои выборы, что является ключевым аспектом профессиональной деятельности программиста. В результате обучающиеся получают более полное представление о предмете и развивают навыки работы с современными инструментами разработки. Это и означает, что адаптация традиционных методов обучения к современным требованиям образовательного процесса приводит к повышению качества образования в области программирования. Таким образом, внедрение предложенного подхода может значительно повы-

сить уровень подготовки студентов и их готовность к реальным вызовам профессиональной деятельности.

Библиографические ссылки

1. *Аленский Н.А.* Методы программирования. Первый и второй семестры [Электронный ресурс]. URL: <https://edummf.bsu.by/course/view.php?id=36> (дата обращения: 30.03.2025)

2. *Аленский, Н. А., Травин В.В., Филимонов В.Д.* Методы программирования: учеб.-метод. пособие [Электронный ресурс]. URL: <https://elib.bsu.by/bitstream/123456789/300305/1/Alenski.pdf>. (дата обращения: 30.03.2025).