

# МОДЕЛЬ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ НА ЭКЗАМЕНЕ ПО ДИСЦИПЛИНАМ ПРОГРАММИРОВАНИЯ

О.А. Кравченко

*Учреждение образования*

*«Гомельский государственный технический университет имени П.О. Сухого»*

*Кафедра «Информационные технологии»*

При изучении дисциплин «Основы алгоритмизации и программирования», «Конструирование программ и языки программирования», «Объектно-ориентированное программирование», «Программирование для мобильных устройств» и других дисциплин, связанных с программированием, студенты должны получить представление об основных структурах данных, об использовании их при проектировании и разработки алгоритмов задач по их математическим моделям, приобрести навыки грамотной постановки задач, возникающих в практической деятельности, для их решения с помощью ЭВМ, получить опыт определения теоретической и экспериментальной оценок временной и емкостной сложности алгоритмов, уяснить связь сложности алгоритма со свойствами структур данных.

Контроль знаний традиционно осуществляется на практических и лабораторных занятиях, зачетах и экзаменах в следующих формах:

- устный опрос;
- самостоятельная работа;
- контрольная работа;
- использование тестирующей системы проверки правильности программ, написанных студентом на лабораторных занятиях, зачетах и экзаменах;
- защита отчета по лабораторной работе;
- проведение коллоквиума;
- защита курсовой работы;
- текущее и итоговое тестирование, в том числе, интерактивное тестирование;
- проведение зачета и экзамена.

Отметим, что с помощью ныне широко используемого компьютерного тестирования контролируются, в первую очередь, знания студентов. В меньшей степени тесты позволяют контролировать умения, приобретенные студентом, для решения задач данной учебной дисциплины. Однако тесты не предоставляют возможность экзаменатору контролировать умения, необходимые для решения практических задач. Особенно это касается учебных дисциплин, связанных с программированием для компьютеров. Здесь очень важно научить студентов составлять необходимый набор тестов и овладеть технологией тестирования и отладки программ. Последнее зависит в огромной степени от сложности алгоритма решаемой задачи.

На экзаменах по дисциплинам физико-математического цикла традиционно контроль знаний, умений и навыков включает практическое задание – решение студентом некоторой задачи. Сложность задачи определяется экзаменатором на основе тематики практических и лабораторных работ и специальности и специализации контингента студентов. Данный подход не приемлем для дисциплин цикла программирования, поскольку потребует много времени – студент должен разработать алгоритм решения задачи, набрать на компьютере текст программы, подготовить набор тестов и провести тестирование и отладку программы. Такое решение задачи за реальное время проведения экзамена возможно только для простых задач.

Суть нашего предложения по содержанию практического экзаменационного задания по дисциплине цикла программирования состоит в следующем. Для оценки

степени освоения студентом методов и технологии применения некоторого языка программирования при решении определенного класса задач студенту предлагается условие задачи и приводится программа, реализующая данную задачу. Причем, известно, что программа не содержит синтаксических ошибок, но содержит, по крайней мере, одну семантическую или логическую ошибку.

Дополнительные условия:

1) программа содержит минимальное количество комментариев, необходимых для понимания смысла используемых основных идентификаторов;

2) сохраняется традиция использования преподавателем обозначений при проведении практических и лабораторных занятий.

Таким образом, цель предлагаемой модели практического задания состоит в экономии экзаменационного времени за счет того, что студент не программирует (как это традиционно делается), а демонстрирует свои способности по составлению полного набора необходимых тестов и применения методов отладки.

Приведем классификацию специально допущенных ошибок в программах практического задания экзамена:

1. Прогон программы при запуске: практически любого теста показывает наличие ошибки. Например, в программе, которая должна из заданного массива (длина  $n$ ) помесечных объемов материальных затрат находить месяц с максимальным объемом выплат, в заголовке цикла просмотра вместо *for i:=1 to n do* написано *for i:=1 to 1 do*. Как правило, такие ошибки легче всего локализируются студентом.

2. Ошибочная работа программы выявляется примерно для половины заготовленных тестов. Например, в программе, работающей с двумя массивами размерности  $n$  и  $m$ , в соответствующих заголовках циклов дважды указывается конечное значение  $n$ . В этом случае при прогоне тестов, в которых будет задано одинаковое число элементов массивов, не выявят ошибки программы.

3. Наиболее трудный для локализации ошибки случай, когда при прогоне большинства тестов программа работает правильно. Например, в заголовке цикла некоторой проверки элементов массива вместо *while (i<=n) and W do* написано *while (i<n) and W do*, что приведет к тому, что последний элемент массива не будет проанализирован.

Опыт использования данной методики при проведении экзаменов и зачетов по дисциплинам «Основы алгоритмизации и программирования» и «Конструирование программ и языки программирования» для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)», 1-40 04 01 «Информатика и технологии программирования», 1-53 01 07 «Информационные технологии и управление в технических системах» Гомельского государственного технического университета имени П.О. Сухого показывает, что студенты, самостоятельно выполнявшие и защитившие отчеты по лабораторным работам курса, успешно справлялись с экзаменационными задачами.

При оценке экзаменационного ответа преподаватель обязательно учитывает полноту подготовленного студентом набора тестов, доказывающего правильность функционирования программы.