

## **КОНТРОЛЬ ЗНАНИЙ В КУРСЕ «ПРОГРАММИРОВАНИЕ»**

*В.Ю. Сакович, О.М. Кондратьева*

*Рассматриваются формы контроля знаний, используемые при изучении дисциплины «Программирование» для студентов специальности «Информатика» факультета прикладной математики и информатики Белорусского государственного университета. Представлены схемы текущего и выходного контроля, выполнен анализ текущего контроля и даны рекомендации по улучшению успеваемости студентов.*

### **Введение**

Объективный контроль знаний, получаемых студентами, является одной из основных проблем управления качеством образовательного процесса. Необходимо уделять особое внимание проведению контроля знаний и увеличению объективности оценки. Используются различные формы контроля знаний:

- входной контроль;
- контроль текущей успеваемости;
- выходной контроль;
- контроль остаточных знаний.

Данная работа посвящена контролю текущей успеваемости и выходному контролю в курсе «Программирование».

Дисциплина «Программирование» ориентирована на обучение студентов базовым знаниям, умениям и навыкам в области проектирования программ. Изучаемые темы базируются на использовании современных информационных технологий, новейшего программного и технического обеспечения компьютеров.

Изучение программирования преследует две основные цели: во-первых, выработать у студентов компетенции, необходимые для усвоения материала последующих учебных дисциплин в области информатики, и, во-вторых, сформировать составную часть банка знаний, необходимых студентам для успешной дальнейшей работы.

В соответствии с типовым учебным планом специальности «Информатика» программа третьего семестра предусматривает для изучения дисциплины 102 аудиторных часа, в том числе лекции – 34 часа, лабораторные занятия – 68 часов [1].

### **Организация занятий в третьем семестре**

Третий семестр является заключительным в курсе «Программирование».

Лекции строятся по следующей схеме:

1. Сентябрь – библиотека МФС;

2. Октябрь – язык программирования Java;
3. Ноябрь – библиотеки Java;
4. Декабрь – декларативные языки (в настоящее время изучаются HTML и XML). Данный теоретический материал используется в четвертом семестре во время проведения учебной практики.

На практике изучаются три раздела:

1. Консольные и оконные приложения на Ассемблере под Windows;
2. Программирование с использованием библиотеки MFC (Microsoft Foundation Classes);
3. Консольные и оконные приложения на Java.

Практические занятия традиционно делятся на занятия в аудитории и в компьютерном классе. Студент обязан выполнить плановый набор заданий. Если студент по неуважительным причинам не сдает задания в отведенный срок, имеется возможность «поощрить» его работоспособность дополнительными задачами или иным способом.

### **Текущий контроль**

Контроль текущей успеваемости организован двумя способами:

1. Проведение в семестре трех контрольных работ по каждому из изучаемых на практике разделов. Текущие знания студента оцениваются преподавателями, ведущими практические занятия. Каждая контрольная работа проверяется двумя преподавателями: первый непосредственно работает в группе, второй – знающий материал, но с группой не работающий.
2. Определение рейтинга знаний студента в процессе семестра (своевременная сдача планового набора заданий, качество выполненных заданий и полученные оценки во время работы в аудитории). Контроль осуществляется преподавателем, ведущим дисциплину.

В конце изучения каждого раздела проводится контрольная работа на компьютере. Студентам предварительно предлагается примерный вариант задания, что позволяет им лучше подготовиться к контрольной работе. Предоставляется три попытки для получения аттестационной оценки.

1. Первая попытка. Максимальная оценка – 10 баллов. Для проверки такого варианта используется многокритериальный подход, что требует от преподавателя значительных затрат времени. Выдаются замечания по реализации задачи (например, эффективность алгоритмов, использование структур данных, правила именования). Таким образом, после первой попытки студент получает не только оценку, но и рекомендации, проработка которых позволит ему улучшить знания по разделу.

2. Вторая попытка. Максимальная оценка – 6 баллов. Здесь преподаватель оценивает функциональность приложения и выявляет случаи списывания. Если студент демонстрирует действительно отличное решение, оценка может быть повышена.

3. Третья попытка. Максимальная оценка – 4 балла.

Ниже приведен пример условия контрольной работы по MFC первой попытки (максимальная оценка 10 баллов).

*Разработать SDI-приложение с поддержкой архитектуры «Документ/Вид». Требуется:*

- реализовать операции с документом: отображение, сохранение/загрузка, редактирование/удаление/добавление;
- выполнить операции редактирования/удаления/добавления данных в одном из двух вариантов: в диалоговом окне (более сложный вариант) и непосредственно в виде (более простой вариант);
- обеспечить проверку вводимых данных (использовать механизм DDX/DDV);
- построить диаграмму в диалоговом окне.

*Документ представляет собой телефонную книгу, хранящую фамилии, имена и номера телефонов абонентов. Проверить, что номер телефона является семизначным. Диаграмма отображает количество абонентов в каждой номерной зоне, которая определяется по первой цифре номера.*

Решение оценивается согласно следующих критериев:

1. Реализация операций с документом:
  - отображение;
  - сохранение;
  - загрузка;
  - создание нового;
  - редактирование;
  - удаление;
  - добавление.
2. Качество реализации операций:
  - какой контейнер используется для хранения данных;
  - используется ли флаг модификации документа;
  - проверяется ли корректность данных при редактировании;
  - производится ли редактирование в отдельном окне;
  - проверяется ли корректность данных при добавлении;
  - корректно ли готовится документ для повторного использования.
3. Диаграмма:
  - построена ли;
  - выводится ли диаграмма в отдельное окно.

Для сравнения приведем условие контрольной работы по MFC третьей попытки (максимальная оценка 4 балла).

*Реализовать MFC-приложение на базе диалога. В текстовом файле хранится информация о работниках: фамилия, номер отдела, зарплата. Отобразить сведения о работниках в компоненте ListBox.*

Как видно, сложность задач первой и третьей попыток существенно различается.

Положительные результаты текущего контроля являются допуском к экзамену (выходному контролю).

### **Выходной контроль**

Результаты контрольных работ студентов передаются лектору вместе с оценкой преподавателя, ведущего практику. По ним рассчитывается оценка текущего контроля знаний. Контрольные работы имеют вес 0,27, оценка преподавателя – 0,19. Уменьшение веса оценки преподавателя связано с попыткой увеличения объективности итоговой оценки текущего контроля. Оценка текущего контроля на экзамене согласно рекомендации методической комиссии факультета имеет вес 0,3.

Экзамен по программированию проходит в два этапа: первый – практический, второй – теоретический. На первом этапе студенты решают задачу на компьютере в течение одной учебной пары. Задача проверяется в присутствии студента. Учитывается не только работоспособность программы, но и другие факторы: оптимальность алгоритма, дружелюбность интерфейса, соглашения об именовании и т.п. Оценка, полученная за задачу, является отправной точкой на втором этапе экзамена. Ответы по теоретическому материалу могут повысить либо понизить оценку максимум на два балла.

### **Результаты текущего контроля**

В этом учебном году нам предоставилась уникальная возможность провести занятия по программированию в двух группах одновременно. В одной группе мы работали с первого курса, и эти студенты приучены к нашей схеме текущего контроля; во второй начали вести занятия только с третьего семестра. Нас интересовал вопрос, насколько хорошо студенты работают в семестре, показателем чего являются результаты текущего контроля. Для сравнения средних баллов двух групп использовался t-критерий Стьюдента [2].

Результаты экзамена по программированию за первый курс показали, что группы не различаются по успеваемости (вероятность  $p \geq 0.99$ ). Анализ результатов текущего контроля приведен в таблице.

Таблица

	Средний балл в первой группе	Средний балл во второй группе	t по критерию Стьюдента
Ассемблер	5.80	3.87	2.5
MFC	6.94	5.66	1.6
Java	6.63	7.46	1.1

Как видно из таблицы, при написании контрольной работы по Ассемблеру первая группа дала существенно лучший результат ( $p \geq 0.95$ ). Со второй контрольной группы показали одинаковые результаты ( $p \geq 0.99$ ).

Таким образом, используемая схема текущего контроля не только позволяет получить более объективную оценку, но и стимулирует студентов к систематической работе в семестре.

### **Рекомендации**

Чтобы успеваемость студентов была достаточной, следует:

1. Обеспечить преемственность проведения курса: как видно из вышеизложенного, результаты группы, в которой такая преемственность была, лучше;
2. Систематически проводить контроль по усвоению теоретического материала;
3. Проводить экспресс-опросы в начале практических занятий.

### **Список литературы**

1. ПРОГРАММИРОВАНИЕ. Типовая учебная программа для высших учебных заведений по специальностям: 1-31 03 03 Прикладная математика (по направлениям), 1-31 03 04 Информатика, 1-31 03 05 Актуарная математика, 1-31 03 06-01 Экономическая кибернетика (математические методы в экономике), 1-98 01 01-01 Компьютерная безопасность (математические методы и программные системы) / Министерство образования Республики Беларусь. Рег. № ТД-G152/тип. 24.09.2008.

2. Автоматический расчет t-критерия Стьюдента [Электронный ресурс] / Психологическая помощь: консультация психолога, психотерапия. – Режим доступа : <http://www.psychol-ok.ru/statistics/student/>. – Дата доступа : 12.04.2011.

*Сакович Вадим Юрьевич, старший преподаватель кафедры многопроцессорных систем и сетей факультета прикладной математики и информатики Белорусского государственного университета, sakovich@bsu.by*

*Кондратьева Ольга Михайловна, ассистент кафедры многопроцессорных систем и сетей факультета прикладной математики и информатики Белорусского государственного университета, konratjeva@bsu.by*