

# МЕТОДИЧЕСКИЕ ОСОБЕННОСТИ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ «ПРОГРАММИРОВАНИЕ»

**О. М. Кондратьева, В. Ю. Сакович**

---

*Белорусский государственный университет*

*Минск, Беларусь*

*E-mail: kondratjeva@bsu.by, sakovich@bsu.by*

Рассматривается методика преподавания дисциплины «Программирование» в третьем семестре для студентов специальности «Информатика» факультета прикладной математики и информатики Белорусского государственного университета. Представлены схемы проведения лекционных и практических занятий, текущего контроля знаний и экзамена.

*Ключевые слова:* программирование, методика преподавания, лекция, текущий контроль знаний, экзамен.

Одним из основных противоречий современной системы ИТ-образования является диссонанс между академическим, университетским подходом к обучению и требованиями, предъявляемыми ИТ-компаниями к программистам. Поэтому важнейшая на сегодняшний день задача системы образования – не только давать готовые знания, но и учить добывать, ранжировать, создавать новые знания и применять их в определенной предметной области. На смену информационно-накопительной модели высшего образования, ориентированной на пассивное усвоение большого объема информации, приходит модель, основанная на создании целостно-ориентированного, конструктивного стиля мышления и новых способов самостоятельной творческой деятельности.

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.

Рассматриваемая дисциплина ориентирована на подготовку специалиста, умеющего проектировать эффективные алгоритмы решения поставленной задачи, выбирать наиболее подходящие структуры данных, программные и технические средства его реализации и с учетом операционного окружения разрабатывать программные приложения, отвечающие современным требованиям и компьютерным технологиям.

Изучение программирования преследует две основные цели: во-первых, дать студентам базу, необходимую для усвоения материала последующих учебных дисциплин в области информатики, и, во-вторых, сформировать составную часть банка знаний, необходимого 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].

Третий семестр является заключительным в курсе «Программирование». Он достаточно сложен и включает три самостоятельных раздела:

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

Лекции строятся по следующей схеме. В сентябре изучается библиотека MFC, затем начитываются язык программирования Java и его библиотеки [2]. Поэтому в сентябре студенты проектируют приложения на языке Ассемблера под Windows на практических занятиях, теоретический материал для которых был изложен во втором семестре. На лекции рассматриваются теоретические аспекты изучаемого языка и исходные тексты учебных программ, материал излагается с использованием проектора.

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.

Основная форма проведения аудиторных занятий – обсуждение решения типовых задач, формулировки которых известны заранее. При подготовке к занятиям студент должен отобрать, изучить необходимый теоретический материал и предложить свои варианты решений типовых задач. Это дает им следующие возможности: сильным – проанализировать различные варианты и мотивированно выбрать лучший, слабым – освоить приемлемый вариант. Для проверки подготовки студента к занятиям используются самостоятельные работы на 10-15 минут, которые могут проводиться как в аудитории, так и в компьютерном классе.

Мы считаем, что при обучении программированию преподаватель обязан оценивать различные параметры, в том числе исходный текст программы. Чтение исходного текста дает возможность привить студенту определенный уровень профессиональной культуры, начиная с реализации алгоритма и заканчивая комментариями и отступами. Программирование практически является единственной дисциплиной, где еще удастся проверить исходный текст программы и повлиять на программистский стиль студента.

Зачастую студент при реализации задач сталкивается с непреодолимыми трудностями. Учитывая ограниченность времени личного общения с преподавателем, приходится использовать другие формы взаимодействия: электронная почта, общие сетевые папки.

В конце изучения каждого раздела дается контрольная работа на компьютере, которая является частью текущего контроля знаний. Перед ее проведением выставляется примерный вариант, что позволяет студенту лучше подготовиться к контрольной работе. Предоставляется три попытки для получения оценки.

1. Первая попытка. Максимальная оценка – 10 баллов. Для проверки такого варианта используется многокритериальный подход к оцениванию, что требует от преподавателя большого объема времени. Также выставляются замечания по реализации задачи, которые не влияют на оценку (например, неэффективные алгоритмы, структуры данных, правила именования). Таким образом, при первой попытке 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. В некоторых случаях, когда студент демонстрирует действительно отличное решение, оценка может быть повышена.

3. Третья попытка. Максимальная оценка – 4 балла.

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

*Разработать параметризованный класс, производящий обработку данных. Разработать иерархию классов, которая используется в параметризованном классе. Объект параметризованного класса инициализируется элементами из текстового файла, передаваемого из командной строки. При реализации генерировать и обрабатывать исключительные ситуации, в т.ч. своих классов. Приготовить два тестовых файла. По каждому выдать результаты работы трех методов.*

По общему условию формулируются конкретные варианты иерархии классов (например, числа, животные, растения и т.п.).

Решение оценивается согласно следующим критериям:

1. Определен ли метод *toString* в своем классе;
2. Обрабатывается ли текстовый файл;
3. Построена ли иерархия классов;
4. Задача подсчета:
  - реализована ли;
  - используется ли сортировка;
  - стандартная ли сортировка;
5. Задача фильтрации:
  - реализована ли;
  - используется ли сортировка;
  - стандартная ли сортировка;
6. Генерируются ли исключения;
7. Есть ли свой класс исключений;
8. Соблюдаются ли соглашения об именовании.

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

*В файле хранятся целые числа. Вывести их в порядке убывания на консоль, предпочтительно используя стандартную сортировку.*

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

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. 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. Считаем, что главная цель курса – научить студента решать задачи. Поэтому

оценка, полученная за задачу, является отправной точкой на втором этапе экзамена. Ответы по теоретическому материалу могут повысить либо понизить оценку максимум на два балла.

Электронные учебные материалы размещены в общих сетевых папках и включают:

1. Программу курса;
2. Электронные книги;
3. Исходные тексты программ, рассматриваемые на лекциях;
4. Готовые к использованию учебные проекты, охватывающие все разделы курса;
5. Материалы к экзамену;
6. Условия контрольных и экзаменационных задач прошлых лет;
7. Методические указания;
8. Базу данных с примерами выполнения различных заданий лучшими студентами;
9. Формулировки заданий и график их сдачи;
10. Экран успеваемости.

Таким образом, в работе представлена методика преподавания дисциплины «Программирование» в третьем семестре. Методика хорошо зарекомендовала себя, уже третий учебный год мы используем ее без изменений. Управляя познавательной деятельностью, мы пытаемся стимулировать самообразование студентов, поиск ими новых знаний и способов IT-интерпретации реальности, что позволяет подготовить их для решения различных задач, связанных с компетенциями в жизненно-практической деятельности.

## ЛИТЕРАТУРА

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. *Хорстманн, Кей С., Корнелл, Гари*. Java 2. Библиотека профессионала, том 1. Основы, 7-е изд.: Пер. с англ./ Кей С. Хорстманн, Гари Корнелл. М.: Издательский дом «Вильямс», 2007. 896 с.: ил.– Парал. тит. англ.