АСПЕКТЫ МЕТОДИКИ ПРОВЕДЕНИЯ ЛАБОРАТОРНЫХ РАБОТ ПО ОСНОВАМ ПРОГРАММИРОВАНИЯ ДЛЯ СТУДЕНТОВ 1 КУРСА

Кравченко О.А.

Учреждение образования
«Гомельский государственный технический университет имени П.О. Сухого»,
Кафедра «Информационные технологии»

В настоящем докладе рассматриваются проблемы обучения основам программирования студентов 1 курса и предлагаются некоторые рекомендации по проведению лабораторных работ.

Цель обучения программированию состоит в том, чтобы научить первокурсников использовать эффективные методы для создания эффективно работающих надежных программ.

Будем использовать следующие неформальные термины:

- под словом «эффективный» следует понимать «как можно лучший», но, ни в коем случае, не «оптимальный», т. е. не «самый лучший»;
- «эффективный метод (способ, прием)» выбирается не из всего возможного набора, определяемого средой программирования, а из набора методов, рассмотренных на учебных занятиях к данному моменту;
- под «эффективно работающей программой» будем понимать программу, которая лучше «неэффективной» по следующим критериям:

а) время выполнения,

б) объем используемых системных и технических ресурсов;

- «надежная программа» – программа, протестированная на максимально возможном наборе тестов.
- для языков программирования используется понятие «надежность»: степень «надежности языка» можно определить, как отношение количества ошибок, допущенных программистом при кодировании алгоритма (с точки зрения программиста – это синтаксические ошибки), к количеству выдаваемых при этом сообщений об ошибках средой программирования, чем меньше значение такого отношения, тем легче процесс тестирования и отладки, т. е. язык более надежен.

Современный этап изучения основ программирования характеризуется следующим:

1. Из учебного процесса исключен созданный специально для обучения надежный язык Паскаль (Delphi), а используются языки класса C и Python, которые предоставляют профессиональным программистам богатые возможности по проектированию и разработке программ, но их использование для обучения основам программирования привело к резкому усложнению процессов разработки и отладки студентами своих программ на лабораторных занятиях.
2. Современных студентов, вчерашних школьников, отличает плохая математическая подготовка.
3. Проблема – отсутствие у студентов логического мышления, которое сказывается не только при разработке схем алгоритмов, но даже при формировании простых логических выражений, включающих только одну логическую операцию.
4. Отметим, что современные студенты очень плохо выражают свои мысли и в устной и письменной форме.

Таким образом, надо найти ответы на следующие методические вопросы:

- какие возможности языка программирования не следует рассматривать при изучении основ программирования с целью повышения надежности подмножества этого языка?
- какова должна быть последовательность выполнения лабораторной работы?

- какие должны быть требования заказчика (преподавателя) к функциональности и оформлению программы?
- по каким критериям следует принимать у студента лабораторную работу, т. е. разработанную программу?

Наш опыт позволяет выделить следующие аспекты методики проведения лабораторных работ по основам программирования для студентов-первокурсников:

1. Для своего варианта задания по лабораторной работе студент сначала должен разработать и представить преподавателю максимально возможный набор тестов (если он не сможет работать программистом, то хотя бы получит навыки тестировщика). Разработка максимально возможного или полного набора тестов очень трудная проблема для наших студентов. Но, по моему мнению, необходимо на лабораторных занятиях требовать у всех студентов выполнения задания по разработке тестов до набора программы. Только по предъявленным тестам можно убедиться, что студент понимает условие задачи.
2. Разработанная на втором этапе решения задачи блок-схема алгоритма должна быть структурной. Таким образом, каждый цикл должен иметь только один выход, а это означает, что не допускается использовать операторы типа break, return, continue и т. д. при написании программы. В блоках не надо записывать операторы программы, а следует указывать математические формулы, краткие описания выполняемых действий и пр.
3. Несмотря на богатые возможности языков C и Python на начальном этапе обучения, по нашему мнению, программы студентов должны иметь простую структуру – описания всех структур данных должны быть сосредоточены в начале программы. Тем самым не будет проблем с локальными и глобальными объектами. Не следует допускать динамические описания, использование которых не дает возможность студентам уяснить разницу между резервированием памяти и определением значений переменных. Обязательная запись программы «лесенкой», при которой конструкции языка располагаются в строках в соответствии с логической подчиненностью.
4. Современный программист – это сотрудник фирмы, а не предприниматель-одиночка. Следовательно, его программа должна быть понята коллегами с минимальными усилиями. Поэтому необходимо учить студентов эффективно располагать комментарии в своей программе. Как правило, комментарии бывают двух типов: к используемым структурам данных (статика) и к функциональности программы (динамика). Многие студенты заменяют «статические комментарии» длинными названиями используемых структур данных и при этом никак не комментируют действия программы. На наш взгляд, длинные названия ухудшают читаемость программы. Все комментарии (статические и динамические) целесообразно единообразно форматировать и располагать в тексте программы единообразно.
5. Разнообразные возможности циклов for и while языков класса C резко снижают надежность студенческих программ. Целесообразно на начальном этапе обучения формализовать использование указанных операторов следующим образом:
   - цикл for использовать в тех и только в тех случаях, когда число повторений заранее определено;
   - в заголовке цикла for не следует записывать операторы, не относящиеся к управлению циклом;
   - цикл while следует использовать тогда, когда истинность условия повторения цикла определяется в процессе выполнения цикла.