

МЕСТО ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Миронов Иван Александрович

магистрант Московского Государственного Областного Университета, РФ, г. Москва

Основной тенденцией образования является подготовка всесторонне развитых специалистов, которые могут применять свои знания, умения и накопленный опыт для решения различного рода производственных задач и способных к дальнейшему самостоятельному развитию. Она свойственна для любой области и информационные технологии не являются исключением.

Учащиеся получают обширные умения и навыки, которым всегда найдется применение в повседневной деятельности. Будь то работа с текстовым, табличным редакторами, организация и управление базами данных, создание и редактирование графических изображений или интерактивных презентаций. Но, к сожалению, мало кто находит применение программированию. В чем причина? Можно ли утверждать то, что программирование необходимо лишь специалистам, чья профессиональная деятельность связана непосредственно с ним или же программирование является неотъемлемой частью компьютерной грамотности?

Причиной является подход к преподаванию. В школе он сводится к изучению семантики языка, синтаксиса и рассмотрению, реализованных на нем алгоритмов. Учащемуся показывают абстрагированный от внешнего мира программ редактор кода, который в лучшем случае умеет показать окно операционной системы и запросить через поле ввода или другие элементы управления информацию от пользователя, в худшем же это строка консольного ввода принимающая на ввод текст и выдающая в ответ так же текст. Никто не рассказывает о том каким образом можно организовать программу, на какие компоненты следует ее разделить, как она может взаимодействовать с внешним миром компьютерных программ расположенных на компьютере пользователя или вне его, различного рода внешних устройств. И тем более не уделяют внимания тому, каким образом обычный пользователь может оптимизировать свою работу, в существующих программах.

В вузе, образовательная программа может содержать в себе изучение нескольких языков программирования. Для каждого из них основной упор будет сделан на один и тот же материал: типы данных, управляющие конструкции, циклы, функции, процедуры. Для объектно-ориентированных языков добавятся классы, объекты, методы, в упрощенном варианте программирование пользовательского интерфейса. Без внимания останутся и их особенности, и то, как перейти от реализации отдельных учебных алгоритмов к полноценным функционирующим программам.

Такой подход к изучению программирования сводится лишь к формированию теоретической базы, недостаточной для дальнейшего самостоятельного развития.

На практике людям занимающимся программированием не приходится сталкиваться с реализацией алгоритмов на том уровне на котором знакомят обучающихся. Работа программиста сводится более к организации алгоритмов в единые, функционирующие механизмы.

Возникает задача, как организовать обучение программированию таким образом чтобы развить алгоритмическую культуру учащихся, практические умения и навыки применения программирования и избавиться от дублирования в содержании.

Для изучения данной проблемы проведено исследования, на примере трех групп учащихся:

Первая группа — учащиеся изучившие процедурный язык программирования Pascal [1]. В рамках образовательного курса программирование второй группы был объектно-ориентированный язык программирования Java [2].

В содержание образовательной программы третьей группы добавлено изучение разного рода проблем проектирования — паттернов программирования.

Для того, чтобы выявить уровень знаний, умений и навыков учащихся каждой группе предложено решить задачу следующего содержания. Имеется файл, который содержит информацию о некоторой выдуманной группе школьников, в формате: имя, фамилия, год рождения, класс и баллы за ряд предметов. Данный файл программа должна прочитать и преобразовать в соответствии с поставленными задачами: отсортировать учеников по столбцу (имя, фамилия, год рождения и др.), распределить стипендию среди участников группы руководствуясь правилами (отличникам — повышенную стипендию, школьникам, имеющим удовлетворительные отметки — поощрительную). Помимо этого, отдельной задачей, требующей от школьников самостоятельного изучения протокола передачи данных HTTP/HTTPS и библиотек по работе с ним, организовать получения файла с веб сервера.

По ходу решения учащиеся столкнулись с рядом проблем проектирования, требующих от них решения задач:

1. Получить из командной строки имя команды и её параметры и в соответствии с значениями организовать внутри программную маршрутизацию (Например: отсортировать учеников по убыванию по имени — `app — sort=name.desc`);
2. Преобразовать сущности, хранящиеся в файле в структуры данных;
3. Разработать механизм, предусматривающий различные форматы вводимых данных (Например: год рождения может быть указан как 2014, так и 14);
4. Реализовать и внедрить в программу алгоритмы поиска и сортировки;
5. Декорировать значение стипендий в зависимости от содержания правил их распределения.

Результаты первой и второй группы очень похожи. Они справились с первыми двумя задачами.

Для решения задач внутри программной маршрутизации и декорирования данных использовали условные конструкции. Разница между ними заключалась в организации хранения данных. Первая группа использовали многомерные массивы, вторая объекты классов.

Если для первой группы, используемый ими язык программирования не предлагал никакой альтернативы, то для второй возникает вопрос, почему они не использовали такие особенности как полиморфизм, наследование, перегрузка методов и др.?

С дополнительным, третьим заданием никто из учеников этих групп не справился.

Учащиеся третьей группы наоборот старались использовать все возможности, которые предлагал им их язык программирования. Полиморфизм на уровне внутренней маршрутизации, на уровне реализаций структур данных, на уровне организаций алгоритмов. Программа проектировалась на уровне интерфейсов, а не реализаций и когда речь зашла о том чтобы видоизменить часть её — это не составило большого количества труда.

Заслуживает внимания подход к поиску решения. Чтобы реализовать загрузку файла, используя протокол HTTP, учащиеся искали конкретную реализацию — вполне уместно сказать конкретный синтаксис. Ход мысли третьей группы был иной. Они рассматривали эту задачу как некоторую абстракцию: для загрузки файла следует обратиться к серверу, для того чтобы обратиться к серверу следует использовать какой-то язык, этим языком в рамках конкретной реализации является HTTP. — Значит каким-то образом следует на языке HTTP передать запрос серверу и тогда он даст ответ. — Каким образом следует передать запрос

серверу? — Записать что в сокет и прочитать ответ. — А на что это похоже? — На работу с файлами? — Так вот что делает эта библиотека, она дает возможность передать параметры серверу и получить от него ответ не задумываясь о реализации.

Таким образом, можно утверждать следующее, при обучении программированию недостаточно изучить семантику и синтаксис языка, важно рассмотреть широкий спектр практических задач и методику их решения. Паттерны программирования в данной ситуации являются прекрасным инструментом обучения программированию. Они позволяют сформировать знания и умения, опираясь на которые учащиеся могут самостоятельно изучать новые технологии, разрабатывать свои собственные программы неограниченными кругом образовательных задач и оптимизировать повседневную деятельность путем использования в ней программирования.

Их следует принять в качестве отправной точки в интересный мир высоких технологий.

Список литературы:

1. Программирование на языке Pascal // Intuit.ru: Национальный Открытый Университет «ИНТУИТ» URL: <http://www.intuit.ru/studies/courses/41/41/info> (дата обращения: 01.02.2014).
2. Программирование на Java // Intuit.ru: Национальный Открытый Университет «ИНТУИТ» URL: <http://www.intuit.ru/studies/courses/16/16/info> (дата обращения: 01.02.2014).
3. Фримен Э., Фримен Э. Паттерны проектирования. СПб.: Питер, 2011.