

НЕКОТОРЫЕ АСПЕКТЫ ПРЕПОДАВАНИЯ ПРОГРАММИРОВАНИЯ

*А.Г. Буймистров, И.И. Гламаздин,
Д.В. Горбачев
БИП – Университет права и социально-
информационных технологий,
Белорусский государственный университет
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Востребованность IT-специалистов делает данный вид профессиональной деятельности привлекательным для молодых людей. Об этом свидетельствует неизменно большой конкурс при поступлении на IT-специальности в учреждения образования Беларуси. С другой стороны, IT-компании предъявляют высокие требования к специалистам, даже уровня Junior, при приеме на работу. Поэтому перед УВО стоит непростая задача – подготовить конкурентноспособных специалистов, готовых к работе в реальном секторе экономики.

Задача усложняется еще тем, что IT-технологии развиваются очень быстро. Например, компания Microsoft в ноябре 2021 года выпустила релиз платформы .Net версии 6, и соответственно, язык программирования C# версии 10. А уже в ноябре 2022 года вышла 7-я версия платформы .Net и 11-я версия языка C#. В 2023 году уже анонсируется выход 8-версии платформы .Net. Для версий же платформы .Net, начиная с 5-й и ниже, уже прекращено развитие и поддержка [1].

Это требует от преподавателя, с одной стороны, быть в курсе всего, что происходит с предметом преподаваемой дисциплины. С другой стороны, нужно регулярно обновлять материалы лекций и практических (лабораторных) занятий, чтобы студенты получали актуальные знания, которые будут востребованы на рынке труда. Быстрое развитие IT-технологий также порождает еще одну проблему: отсутствие актуальных учебных печатных изданий.

Возьмем, например, фреймворк ASP.Net Core. Издательство O'Reilly в феврале 2022 года выпустило книгу Адама Фримана (Adam Freeman) «Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages» [2]. Это на 3 месяца позже официального выхода 6-й версии платформы .NET. Естественно, что русский перевод выйдет еще позже (на момент написания статьи книга еще не переведена).

Возникает вопрос: а где же брать материал для обучения? Ответ простой – из первоисточника. Например, что касается докумен-

тации по платформе .NET, всю необходимую информацию можно получить на обучающей странице компании Microsoft [3]. Кроме того, можно получать самую свежую информацию из сообщества [4], из официальных каналов на YouTube [5], из блогов ведущих специалистов сообщества компании Microsoft [6]. Но есть одна проблема: представленная там информация – это хороший справочник, бюллетень новостей, но не учебник. Да, там можно найти подробные ответы на некоторые вопросы. Но это подходит для программиста, который хорошо знает предмет, и просто ему необходимо уточнить некоторые детали, и совсем не оптимально для изучения языка/библиотеки/шаблона/фреймворка. Ведь обучение – это не просто получение пакета информации. Эта информация, как минимум, должна излагаться в «правильной» последовательности. Тем более, что новые возможности новой версии языка не упраздняют то, что предлагалось в предыдущих версиях.

И здесь возвращаемся к преподавателю, который должен уметь обработать новую информацию и правильно преподнести ее на занятиях.

Например, при переходе от версии платформы .NET 5 к .NET 6 существенно изменился шаблон проекта ASP.NET Core. Следовательно, на занятиях нужно было знакомить как со старой версией, так и с новой, поскольку поддержка версии платформы .Net 3.1 прекратилась только в ноябре 2022. Потребовалось также обновление содержания заданий к лекционным и практическим лабораторным занятиям.

Однако донести информацию – это еще не все. Нужно еще научить этой информацией пользоваться, и, что немаловажно, пользоваться правильно. Поэтому важно правильно составить задания к лабораторным работам. Особенно это важно на начальных этапах обучения, при знакомстве с языком программирования.

Большинство современных языков программирования являются объектно-ориентированными. И лабораторные работы связаны с созданием классов, интерфейсов, описанием наследования и т.д. Как правило, задания носят императивный характер: описать класс, описать конкретные методы, реализовать отношения между классами.

Например, имеем предметную область – АТС. Задание в этом случае может быть следующим: «Опишите класс «АТС». Опишите класс «Тариф», содержащий свойства «Город абонента» и «Стоимость». В классе «АТС» опишите коллекцию объектов «Тариф».

Студенты учатся работать с классами. Однако многолетний опыт преподавания показывает, что при таком подходе, умея работать с классами, студенты зачастую не знают, как этим пользоваться. Это напоминает ситуацию, когда ребенок, зная буквы, не умеет составлять из этих букв слова. До сих пор знакомство с программированием у студентов заключалось в построении алгоритмов. Дисциплины, в которых рассказывается о методах проектирования программного обеспечения, еще будут изучаться в последующих семестрах. Но изучение объектно-ориентированного языка предполагает также и развитие навыка объектно-ориентированного подхода к решению задачи. Поэтому вместо императивной постановки задачи авторы предлагают использовать декларативный подход, т.е. описывать задачу так, как если бы ее ставил заказчик, а не программист. Речь идет о том, чтобы уже сейчас научить студента анализировать поставленную задачу с точки зрения парадигм ООП – инкапсуляции, наследования и полиморфизма.

Также авторы считают полезным, когда задание очередной лабораторной работы является продолжением, развитием предыдущей. В этом случае можно поставить задачу так, что ошибки объектно-ориентированного подхода станут очевидными. Например, если приведенной выше задаче студент не выделил АТС в отдельный класс, а новое задание предусматривает наличие нескольких АТС.

В заключение хотелось бы отметить, что в современном обществе, когда информация и знания становятся высшей ценностью, а информационная культура и компетенции человека – определяющим фактором профессиональной деятельности, изменяются и требования к системе образования, происходит существенное повышение статуса образования как универсальной формы деятельности, направленной на постоянное развитие личности в течение всей жизни человека.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *.NET and .NET Core Support Policy* [Электронный ресурс]. – Режим доступа: <https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core>. – Дата доступа: 10.03.2023.

2. *O'Reilly* [Электронный ресурс]. – Режим доступа: <https://www.oreilly.com/library/view/pro-aspnet-core/9781484279571/>. – Дата доступа: 10.03.2023.

3. *Learn to code C#* [Электронный ресурс]. – Режим доступа: <https://dotnet.microsoft.com/en-us/learntocode>. – Дата доступа: 30.01.2023.

4. *.NET Developer Community* [Электронный ресурс]. – Режим доступа: <https://dotnet.microsoft.com/en-us/platform/community>. – Дата доступа: 30.02.2023.

5. *Dotnet* [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/dotnet>. – Дата доступа: 30.02.2023.

6. *.NET Blog* [Электронный ресурс]. – Режим доступа: https://devblogs.microsoft.com/dotnet/?c=37&WT.mc_id=dotnet-35129-website. – Дата доступа: 30.02.2023.