

ОСНОВНЫЕ ПОДХОДЫ К ОБУЧЕНИЮ СТУДЕНТОВ ОБЪЕКТНО- ОРИЕНТИРОВАННОМУ ПРОГРАММИРОВАНИЮ И ПРОЕКТИРОВАНИЮ

Петров А.Н.

*Московский государственный областной
университет
Москва, Россия*

Информатика является одной из самых динамично изменяющихся и обновляющихся учебных дисциплин. В настоящее время методология объектно-ориентированного программирования является ведущей в информатике. Методика обучения программированию должна соответствовать современным разработкам в компьютерных науках, поэтому существует насущная потребность в глубоком изучении методологии объектно-ориентированного программирования.

Обучению объектно-ориентированному программированию и проектированию посвящены работы зарубежных исследователей, таких как Г. Буч, Б. Мэйер, А. Якобсон, Дж. Рамбо, И. Грэхем, М. Фаулер, а также исследования отечественных ученых Г.С. Ивановой, Т.Н. Ничушкина, Е.К. Пугачева, И.А. Бабушкиной и др.

В процессе обучения объектно-ориентированному программированию (ООП) преимущественно используются традиционные формы обучения, такие как лекционные и семинарские занятия, которые хорошо себя зарекомендовали в процессе обучения структурному программированию, но в современных условиях явно недостаточны для изучения такой сложной методологии программирования, как ООП.

Необходимо ответить на следующий вопрос. Если студенты не будут учиться реализовывать преимущества методологии ООП на практике, то будет ли смысл в изучении ими этой методологии программирования? На этот вопрос мы однозначно отвечаем: "Нет". Представим наши рассуждения о том, почему так важно рассматривать преимущества ООП на практике в процессе обучения студентов в вузе.

Освоение студентами на практике принципов ООП вне контекста преимуществ, которые эта методология программирования предлагает, может привести к формализму в процессе их обучения. Студенты будут обладать необходимыми теоретическими знаниями об ООП, но не смогут эффективно применять эти знания при выполнении сложных проектов на практике. Следовательно, в процессе обучения объектно-ориентированному программированию необходимо обратить особое внимание студентов не только на рассмотрение преимуществ ООП, но и на обучение реализации этих преимуществ на практике.

В процессе обучения студентов объектно-ориентированному программированию должны решаться следующие образовательные задачи:

- формирование представления об ООП;
- обучение принципам ООП;
- обучение объектно-ориентированному проектированию;
- обучение объектной декомпозиции;
- обучение созданию повторно используемого программного кода.

Для решения поставленных образовательных задач необходимо использовать не только такие организационные формы обучения, как индивидуальная и групповая, но и коллективную форму организации процесса обучения, которая наиболее приемлема в работе над проектами. Известный ученый Б. Мэйер указывал на то, что факультеты университета должны создавать многосеместровые объединяющие проекты, которые обеспечат "лучшее обучение, разработку новых курсов, факультетские исследования" [2, с. 926].

На многих кафедрах идет разработка проектов, например, исследования по электронному и дистанционному обучению. К работе над проектами необходимо привлекать преподавателей и студентов других кафедр, тогда студенты в процессе обучения в вузе будут иметь возможность участвовать в разработке сложного программного обеспечения, реализующего на практике теоретические исследования, проводимые на кафедрах университета. Это позволит формировать у них опыт коллективного создания программного обеспечения, опыт, который можно будет получить в работе над долгосрочными проектами, что будет способствовать их профессиональной подготовке. Все это требует продуманной организации, как самостоятельной работы студентов, так и организации их коллективной деятельности.

Обучение ООП можно разделить на четыре вида:

- 1) задания (1-5 минут), включающие 1-2 класса, на которых проверяются знания синтаксиса языка программирования или нотаций диаграммы классов языка UML, (индивидуальная форма);
- 2) лабораторные работы (1-2 часа), использующие в среднем 3-5 классов, позволяющие на практике изучать применение принципов ООП (групповая форма);
- 3) проекты (1-3 месяца), использующие 5-10 классов, дающие возможность студентам осваивать объектную декомпозицию (не рассматриваемую в полной мере на лабораторных работах). Объектная декомпозиция - это разделение программы на объекты, она помогает осуществлять процесс поиска классов. Студенты должны иметь представление о том, как осуществляется процесс поиска классов, отбор классов и "отбраковка" не подходящих классов (индивидуальная и групповая формы обучения);

4) долгосрочные проекты (1-3 года), организуются не преподавателями, а несколькими кафедрами университета (по прикладной тематике возможно участие кафедр педагогики, психологии и др.), и основываются на тесном сотрудничестве, наличии координационного центра, планирующего и организующего всю работу. Примером долгосрочного проекта может быть создание программного обеспечения для электронного обучения, при реализации которого студенты получают знания и опыт коллективной разработки программного обеспечения, они смогут учиться создавать повторно используемый программный код (коллективная форма обучения).

Одним из важных преимуществ ООП является возможность создания повторно используемого программного кода. Применение методологии ООП не гарантирует то, что программный код будет являться повторно используемым, но реализация принципов ООП способствует написанию повторно используемого кода. Очень сложно создавать программный код, который действительно является повторно используемым, но затраты на его создание со временем окупаются, так как этот код можно будет применять в большом числе проектов. В повторно используемом коде будут сохраняться уже имеющиеся разработки. Хорошо отлаженный и проверенный программный код при повторном использовании может обеспечить большую надежность программного обеспечения.

Анализ образовательных программ по специальности информатика показал наличие ряда проблем, связанных с недостаточным использованием в процессе обучения объектно-ориентированному программированию средств объектно-ориентированного проектирования.

В практике обучения ООП в вузе наблюдаются следующие противоречия:

- поверхностно представлена методология объектно-ориентированного программирования, в то время как она является одной из основных;

- студентами изучается преимущественно алгоритмическая декомпозиция, а объектная декомпозиция рассматривается недостаточно;

- студенты не накапливают опыт в использовании средств объектно-ориентированного проектирования, столь необходимый им в будущей работе.

Таким образом, необходимо обратить внимание на формирование у студентов представления об ООП и объектно-ориентированном проектировании.

В процессе обучения ООП рекомендуется использовать средства объектно-ориентированного проектирования (язык UML и пакет BlueJ). Язык UML в процессе обучения ООП помогает студенту:

- представлять взаимодействия и отношения между классами наглядно;

- абстрагироваться от конкретной реализации классов и концентрироваться в целом на классах и отношениях между ними;

- управлять созданным программным кодом;

- реализовывать объектную декомпозицию на практике.

Пакет BlueJ предназначен специально для обучения объектно-ориентированному программированию. Он включает только самые необходимые элементы диаграммы классов языка UML, что позволяет студентам сконцентрироваться на выполняемой ими задаче и не затрачивать время на изучения интерфейса и возможностей учебного инструмента разработки. Обучение студентов объектно-ориентированному программированию с использованием языка UML и пакета BlueJ будет способствовать освоению ими методологии объектно-ориентированного программирования.

В ООП одинаково важную роль играет как объектная декомпозиция, так и алгоритмическая. Но, как показал анализ литературы [1], [4] и др., в современном процессе обучения ООП преобладает алгоритмическая декомпозиция, а объектная декомпозиция занимает второстепенное положение. Доминирование алгоритмической декомпозиции можно объяснить тем, что она достаточно подробно изучается в школе, она доступнее и проще объектной декомпозиции. С другой стороны, наблюдается недостаток в учебной и учебно-методической литературе по обучению объектной декомпозиции.

Несмотря на трудности обучения объектной декомпозиции, следует помнить, что организовать обучение ООП без нее невозможно, потому что объектная декомпозиция является важным фактором в формировании представлений об объектно-ориентированном программировании.

До тех пор, пока у студентов не сформировалось представление об объектно-ориентированном программировании и проектировании, объектная декомпозиция должна преобладать в учебном материале и, по-возможности, необходимо сокращать рассмотрение алгоритмической декомпозиции. Когда у студентов сформируются представления об ООП, и они смогут самостоятельно осуществлять процесс поиска классов, тогда можно будет более детально рассматривать алгоритмическую декомпозицию.

В современном процессе обучения ООП преобладают задачи и лабораторные работы по построению графического интерфейса пользователя. Это приводит к тому, что студенты в большинстве случаев будут испытывать затруднения при решении нестандартных задач по ООП, которые не имеют отношения к графическому интерфейсу пользователя. Это можно объяснить отсутствием проблемных ситуаций, связанных с ООП,

при разработке графического интерфейса пользователя, использование которого не позволяет студентам получить достаточный опыт в разрешении проблемных ситуаций. Задания и лабораторные работы по построению графического интерфейса пользователя не подходят для обучения студентов реализации на практике объектной декомпозиции. Поэтому в процессе обучения ООП рекомендуется использовать сюжетные задания, лабораторные работы и проекты, которые помогут студентам в формировании опыта по разрешению проблемных ситуаций в ООП, обучению объектной декомпозиции и способствующих развитию устойчивой познавательной мотивации. Построение графического интерфейса пользователя желательно рассматривать как дополнение к учебным проектам.

Сюжетные задачи строятся на основе реальных объектов, процессов, ситуаций и т.д. Например, игра в шахматы, моделирование дорожного движения на перекрестке или работы лифта и т.п.

Итак, если в процессе обучения объектно-ориентированному программированию и проектированию:

- используют средства объектно-ориентированного проектирования;
- доминирует объектная декомпозиция по отношению к алгоритмической декомпозиции;
- решают сюжетные задачи и задачи, имеющие объекты, прототипами которых являются реально существующие объекты,
- изучают на практике преимущества объектно-ориентированного программирования и проектирования;
- применяют методические рекомендации, рассмотренные в нашем исследовании,

то у студентов формируются такие знания об объектно-ориентированном программировании и проектировании, которые соответствуют современным требованиям подготовки специалистов.

СПИСОК ЛИТЕРАТУРЫ:

1. Бабушкина И.А. Практикум по объектно-ориентированному программированию. / Бабушкина И.А., Окулов С.М.: - М: Бином, Лаборатория знаний, 2004. - 366 с.
2. Бертран М. Объектно-ориентированное конструирование программных систем / Пер. с англ. - М.: Издательско-торговый дом «Русская Редакция», 2005. - 1232 стр.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд. / Пер. с англ. - М.: «Издательство Бином», СПб: «Невский диалект», 1998г. - 560 с.
4. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование: Учебник для вузов. – 3-е изд., стер./Под ред. Г.С. Ивановой. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2007. – 368 с.

ПАМЯТЬ ВНИМАНИЯ - ОСНОВНОЕ ПОНЯТИЕ АНТРОПОЛОГИЧЕСКОЙ ТЕОРИИ ТВОРЧЕСТВА И КРЕАТИВНОСТИ

Петрова В.Н.

*Университет Российской академии образования
Москва Россия*

Быть творцом, а не рабом или ремесленником – суть самовоспитания современного человека, если, конечно, он хочет реализовать свой потенциал, если он стремится стать высококлассным специалистом. Одним из первых шагов в обучении мышлению и творчеству является повышение концентрации внимания. Концентрации внимания следует учить не на срисовывании узоров, не на запоминании и затем воспроизведении какого-либо предмета, а на материале, который человеку интересен, который будет служить основой либо в процессе мышления, либо в процессе творчества. Тогда будет возможно изучать свои собственные особенности, тогда будет возможно легко и быстро входить в состояние максимальной концентрации внимания. Концентрация внимания осуществляется при помощи памяти внимания, которая была впервые описана Алексеем Николаевичем Петровым в Антропологической теории творчества и креативности.

Мы выделяем три способа управления вниманием:

1. память внимания (основной инструмент),
2. прямое управление (человек сам управляет вниманием),
3. противоречие (искусственный инструмент).

Остановимся на памяти внимания. Память внимания человека хранит информацию о том, как надо поступать вниманию в каждой конкретной ситуации, в ней сохраняются его элементы: культуры, мировоззрения, представления, убеждения... Аналогию с памятью внимания можно провести, рассматривая двигательную память.

Например, когда человек поднимает руку, то он не знает, какие мышцы необходимо привести в движение, а двигательная память человека «знает», в каком направлении, с какой скоростью запустить ту последовательность опорно-двигательной системы, которая приведет к желаемому результату. Это происходит тогда, когда у человека данное действие доведено до автоматизма. Но, если он делает то, что он никогда не делал, то его движения замедлены до своей минимальной скорости. Почему? Мы считаем, что в этот момент происходит запись этого движения в двигательную память (стирание предыдущего варианта и запись на него нового, без возможности вернуться к предыдущему опыту). Существует очень интересный процесс «затирания», не такой как в памяти, где можно вспомнить, в памяти внимания «затирание» происходит навсегда, так как разрываются все предыдущие связи и