

**ОБЗОР ПОДХОДОВ К ИЗУЧЕНИЮ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО
ПРОГРАММИРОВАНИЯ В ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ**

Конюхов Сергей Леонидович
ст. преподаватель,

*Мелитопольский государственный педагогический
университет имени Богдана Хмельницкого,
Мелитополь, Украина*



**REVIEW OF APPROACHES TO THE STUDY OF OBJECT-ORIENTED
PROGRAMMING IN HIGHER EDUCATION INSTITUTIONS**

Konyukhov Sergiy Leonidovich
senior lecturer,

*Melitopol State Pedagogical
University named after Bohdan Khmelnytsky,
Melitopol, Ukraine*

АННОТАЦИЯ.

В статье выполнен краткий обзор современного состояния преподавания объектно-ориентированного программирования в высших учебных заведениях. Формулируется понятие и основные принципы ООП. Рассматриваются проблемы, которые возникают в процессе изучения ООП, а также подходы к определению содержания и структуры, выбору формы и методов обучения студентов высших учебных заведений.

Ключевые слова. Объектно-ориентированное программирование, инкапсуляция, наследование, полиморфизм, класс, язык программирования, метод проектов.

ABSTRACT.

The article presents a review of the current state of teaching object-oriented programming in universities. It is formulated the definition and basic principles of the OOP. It is spoken about the problems that arise in the course of the OOP, as well as approaches to the determination of content and structure, the choice of forms and methods of teaching students in universities.

Keywords. Object-oriented programming, encapsulation, inheritance, polymorphism, class, programming language, project-based method.

Актуальность. Существенные изменения в структуре занятости мирового населения во многом обусловлены развитием информационного общества, которое, в частности, характеризуется повышением ценности информации и знаний, а также усилением требований к надежности и эффективности методов хранения и обработки информации. В связи с этим, современные предприятия заинтересованы в квалифицированных сотрудниках сферы информационных технологий и предъявляют жесткие требования к их профессиональным компетенциям. С другой стороны, предполагается, что срок подготовки таких специалистов должен быть небольшим, а дальнейшее повышение квалификации возлагается на самих работников, так как технологии в этой отрасли развиваются очень динамично. Данная ситуация привела к появлению множества коммерческих организаций, которые предлагают слушателям качественное обучение в сжатые сроки в форме различных курсов, конференций, онлайн и офлайн семинаров, создавая серьезную конкуренцию для высших учебных заведений.

Важной особенностью обучения будущих специалистов сферы информационных технологий в вузах является необходимость оптимального сочетания фундаментальной теоретической подготовки с реальной практикой разработки программных продуктов. Такая возможность может быть обеспечена только при условии формирования учебных планов с жесткой внутренней логикой, ориентированной на профессиональную специализацию, наличия хорошей материально-технической базы и связей с предприятиями-работодателями.

Эта проблема остро проявляется в процессе изучения объектно-ориентированного программирования (ООП), которое на сегодня является наиболее актуальной парадигмой разработки программного обеспечения. Обучение ООП, как правило, включает ознакомление с теоретическими основами и более-менее серьезное изучение инструментария конкретной среды разработки, что не способствует формированию целостного профессионального мировоззрения. Таким образом, необходимо усовершенствовать

методическую систему обучения дисциплине "Объектно-ориентированное программирование" с целью более основательной подготовки студентов к будущей деятельности по специальности.

Постановка задач статьи. С целью формирования теоретической основы для усовершенствования преподавания курса «Объектно-ориентированное программирование» необходимо изучить и проанализировать существующий опыт. В связи с этим, в статье будет проведен обзор подходов к определению содержания и структуры данной дисциплины, а также будут рассмотрены некоторые формы и средства формирования соответствующих компетенций, применяемые в практике высших учебных заведений.

Основные понятия объектно-ориентированного программирования.

Приведем определение, сформулированное Г. Бучем: «Объектно-ориентированное программирование – это метод программирования, основанный на представлении программы в виде совокупности взаимодействующих объектов, каждый из которых является экземпляром определенного класса, а классы являются членами определенной иерархии наследования.» [3, с. 69].

Он подчеркивает, что программа будет объектно-ориентированной, если соответствует трем условиям: использует в качестве основных логических конструктивных элементов объекты, а не алгоритмы; каждый объект является экземпляром определенного класса; классы образуют иерархии [3, с. 69].

В основе ООП лежат три принципа [1, с. 189-190]:

1. «Инкапсуляция (encapsulation) – это механизм, который объединяет данные с методами их обработки и обеспечивает их защиту от постороннего вмешательства и неправильных действий пользователя.»

2. «Наследование (inheritance) – это процесс, когда один объект наследует свойства и методы другого объекта, и добавляет к ним оригинальные свойства и методы. Благодаря наследованию удается поддерживать иерархию классов, что, в свою очередь, делает управляемыми большие потоки информации.»

3. «Полиморфизм (polymorphism) – это возможность использования одного и того же метода (или свойства) разными объектами для обработки данных разных типов. Определяется единая последовательность действий, которая не зависит от конкретных данных. Как правило, идентичные свойства и методы принадлежат классу в целом, и каждый экземпляр класса (объект) может ими пользоваться.»

Использование объектно-ориентированной парадигмы наиболее целесообразно для разработки больших проектов, так как она позволяет создавать модели сущностей, которые рассматриваются в процессе выполнения задачи. В результате алгоритмы становятся более естественными, программа максимально приближается к

рассматриваемой предметной области и разработчик может оперировать соответствующими понятиями, а не формальными конструкциями языка программирования, что существенно облегчает дальнейшее расширение программы. Применение такого подхода для решения небольших задач только неоправданно увеличит размер кода и его сложность.

Некоторые проблемы в изучении ООП.

Рассмотренные особенности и преимущества ООП усложняют подготовку программистов. Согласимся с В.Л. Дмитриевым, который отмечает, что в соответствии с данной парадигмой фрагменты одного алгоритма могут быть распределены по многим методам, принадлежащим разным объектам, поэтому их трудно понять и составить целостное представление о программе [5, с. 70]. Это создает серьезные проблемы для студентов, которые еще не научились мыслить соответствующими категориями и абстракциями и зачастую не готовы к возможным трудностям.

Как отмечает Д.Н. Лавров: «Необходимо сломать установку учащегося, которая скрывается за высказыванием: «Но ведь программа работает правильно». «Работает правильно» - это необходимое условие хорошей программы, но, к сожалению, недостаточное» [6, с. 181].

Обучение будущих ИТ-специалистов также осложняется тем, что система образования имеет достаточно жесткую структуру, которая тяжело адаптируется к новым условиям. Так, в высших учебных заведениях зачастую отсутствуют возможности для специализации преподавателей и студентов, организации совместной исследовательской деятельности, а также для практического ознакомления и применения актуальных технологий разработки программного обеспечения.

Резюмируя сказанное, назовем основные, на наш взгляд, причины низкой эффективности обучения студентов компьютерных специальностей объектно-ориентированному программированию: достаточно низкий уровень развития у них абстрактного и образного мышления; сложность формирования положительной внутренней мотивации; ориентация на использование готовых классов без понимания их внутренней природы; отсутствие условий для формирования навыков разработки больших программных проектов, которые демонстрируют преимущества объектно-ориентированного подхода.

Подходы к выбору содержания и средств обучения объектно-ориентированному программированию.

Объектно-ориентированная парадигма подробно изложена в классических работах зарубежных авторов, таких как: «Объектно-ориентированный анализ и проектирование» Г. Буча, «Объектно-ориентированное программирование» Т. Бадда, «Объектно-ориентированное программирование на С++» А. Пола, «Язык программирования С++» Б. Страуструпа, «Объектно-ориентированное

программирование в С++» Р. Лафоре, «Объектно-ориентированное конструирование программных систем» Б. Мейера. Считаю, что при изучении курса ООП целесообразно придерживаться содержания и структуры, предложенных в этих учебниках.

Так, например, в книге Г. Буча [3] предлагается следующая последовательность изложения теоретического материала:

Объектная модель (глава 2):

1.1. Эволюция объектной модели: Поколения языков программирования; Топология языков первого поколения и ранних языков второго поколения; Топология языков программирования позднего второго и раннего третьего поколения; Топология поздних языков третьего поколения; Топология объектных и объектно-ориентированных языков.

1.2. Основные положения объектной модели: Объектно-ориентированное программирование; Объектно-ориентированное проектирование; Объектно-ориентированный анализ.

1.3. Составные части объектного подхода: Абстрагирование; Инкапсуляция; Модульность; Иерархия; Контроль типов; Параллелизм; Персистентность.

1.4. Применение объектной модели: Преимущества объектной модели; Нерешенные вопросы.

Классы и объекты (глава 3):

1.5. Природа объектов: Что является и что не является объектом; Состояние; Поведение; Операции; Роли и обязанности; Объекты как автоматы; Индивидуальность.

1.6. Отношения между объектами: Связи; Видимость; Синхронизация; Агрегация.

1.7. Природа класса: Что такое класс; Интерфейс и реализация; Жизненный цикл класса.

1.8. Отношения между классами: Ассоциация; Наследование; Агрегация; Зависимости.

1.9. Взаимосвязь классов и объектов: Отношения между классами и объектами; Роль классов и объектов в анализе и проектировании.

1.10. Качество классов и объектов: Измерение качества абстракции; Выбор операций; Временная и пространственная семантика; Выбор отношений; Выбор реализации; Упаковка.

В высших учебных заведениях России и Украины существуют различные подходы к отбору и структурированию информации для изучения в курсе объектно-ориентированного программирования, а также к выбору базового языка программирования, что отражается в действующих учебных программах, учебниках и методических пособиях.

Так, например, в пособии, подготовленном коллективом авторов (И.А. Волкова, А.В. Иванов, Л.Е. Карпов), в качестве основного языка выбран С++ и рассматриваются такие основные темы [4, с. 3-4]:

1. Обзор основных парадигм программирования, основные принципы ООП, абстрактные типы данных.

2. Отличия С и С++, стандартная библиотека С++.

3. Классы и объекты С++ (синтаксис описания класса, управление доступом к членам класса, классы и структуры, класс как область видимости, объявление и определение методов класса, указатель на текущий объект, указатель на член класса).

4. Конструкторы и деструкторы (конструктор умолчания, конструктор преобразования и конструкторы с двумя и более параметрами, конструктор копирования, автоматическая генерация конструкторов и деструкторов, список инициализации, порядок вызова конструкторов и деструкторов).

5. Статические члены класса.

6. Константные члены класса.

7. Друзья классов.

8. Статический полиморфизм (перегрузка бинарных и унарных операций, перегрузка функций).

9. Виды отношений между классами.

10. Одиночное наследование (правила наследования, преобразования указателей, правила видимости при наследовании, закрытое наследование, перекрытие имен, наследование и повторное использование кода).

11. Динамический полиморфизм, механизм виртуальных функций.

12. Средства обработки ошибок, исключения и обработка исключений.

13. Множественное наследование, интерфейсы (видимость при множественном наследовании, виртуальные базовые классы, интерфейсы).

14. Динамическая информация о типе (RTTI).

15. Параметрический полиморфизм (параметры шаблона, шаблоны функций, специализация шаблонной функции, шаблонные классы, эквивалентность типов).

16. Стандартная Библиотека шаблонов STL.

Л.Р. Камалова, составитель учебно-методического комплекса по дисциплине «Объектно-ориентированное программирование» для студентов Сыктывкарского лесного института, выбирает в качестве языка программирования С++ (среда разработки MS Visual C++) и предлагает такой перечень тем и вопросов [9, с. 7-8]:

1. Методологии программирования (структурное, модульное, объектно-ориентированное программирование).

2. Объектно-ориентированные возможности языка программирования С++ (понятие класса, доступ к данным и методам классов, конструкторы и деструкторы, реализация методов класса, постоянные функции, объявление и реализация классов).

3. Работа с указателями и ссылками (понятие указателя, объявление и цели использования указателей, операторы для работы с указателями,

понятие ссылки, объявление ссылок, указатель this, паразитные указатели, возвращение функцией нескольких значений, передача ссылок как средство повышения эффективности программ).

4. Перегрузка функций и операторов (перегрузка функций и конструкторов, конструктор копий, перегрузка операторов, перегрузка префиксных и постфиксных операторов, перегрузка бинарных операторов).

5. Наследование, виртуальные функции (понятие наследования, наследование конструкторов и деструкторов, переопределение функций в производных классах, сокрытие методов базового класса, виртуальные функции, одиночное наследование, множественное наследование, виртуальное множественное наследование, абстрактные типы данных).

В отличие от рассмотренных выше, коллектив авторов Харьковского национального экономического университета (Ю.Э. Парфёнов, А.В. Щербаков, В.Н. Федорченко) предлагает изучать ООП с использованием языка программирования C# и платформы Microsoft .NET Framework. Перечень тем, которые выносятся на изучение [11, с. 6-7]:

1. Объектно-ориентированный анализ и проектирование: Основные положения объектно-ориентированного подхода; Основы объектно-ориентированного проектирования на языке UML.

2. Технология ООП: Введение в платформу Microsoft .NET и язык C#; Реализация главных концепций объектно-ориентированного программирования на языке C#.

3. Обработка исключений библиотеки классов: Обработка исключительных ситуаций; Стандартные библиотеки классов Microsoft .NET Framework.

4. Объектно-ориентированное программирование Windows-приложений на платформе Microsoft .NET: Разработка DLL-библиотек; Основы событийно-управляемого программирования; Основы использования технологии Windows Forms; Разработка графических интерфейсов пользователя с помощью технологии Windows Forms; Использование графических возможностей платформы Microsoft .NET; Разворачивание программного продукта; Обзор других технологий разработки Windows-приложений на платформе Microsoft .NET.

В рассмотренных курсах, как и во многих других, в качестве средства обучения используется язык C++ и C#. Отметим, что в работах зарубежных авторов, посвященных объектно-ориентированному программированию, в качестве основного языка также часто рассматривается C++, так как он был создан для написания сложных проектов, реализован в составе многих сред разработки и является основой для других объектно-ориентированных языков (Java, C# и т.д.). В то же время, в отдельных работах (И.А. Ломазова [7], Д.Н. Лавров [6]) предлагаются язык Java, что связано с ростом его популярности и кроссплатформенностью; Delphi (Л.Н. Паламарчук, В.Н. Гладков, И.А. Гринёв [10]), так как он более

легкий для изучения; параллельное изучение C++ и Delphi (И.В. Синельникова, Л.А. Калиникова [12]); параллельно изучение Java и Python (Н.А. Мясникова [8]).

Формы и методы обучения объектно-ориентированному программированию.

Вопросы определения наиболее эффективных форм и методов обучения объектно-ориентированному программированию рассматривают П.И. Алексеевский, В.Н. Гладков, И.А. Гринёв, Л.Н. Паламарчук, З.С. Сейдаметова, С.А. Семериков, В.Я. Талгатова, М.А. Умрик и др. Проведенные исследования позволили разработать ряд методических систем, которые могут быть применены в практике высших учебных заведений.

П.И. Алексеевский предлагает организовать обучение на основе методологии унифицированного процесса разработки программного обеспечения. При этом дисциплина изучается в порядке, соответствующем порядку следования фаз унифицированного процесса с возможностью возврата на предыдущий этап: стартовый этап, проектирующий этап, этап внедрения, этап представления. Курс рассматривает такие основные вопросы: структура, особенности и средства унифицированного процесса; основы объектного моделирования с использованием языка UML, основные теоретические принципы объектно-ориентированного программирования; языки программирования C и C++, реализация различных алгоритмов на этих языках, использование сторонних библиотек для разработки программ, методы оценки сложности алгоритмов, способы оптимизации программного кода на различных уровнях, средства отладки и профилирования; средства генерации программной документации по объектной модели и исходному коду программы. На разных этапах учебного процесса автор применяет комплекс методов обучения, таких как: проблемные лекции, имитационные упражнения, игровое проектирование, ситуационные методы и т.д. Формирование специальных умений и навыков происходит путем выполнения учебного проекта по разработке компьютерной игры в небольших группах студентов [2, с. 151-153].

Обучение ООП с использованием мини-проектов предлагают Л.Н. Паламарчук, В.Н. Гладков, И.А. Гринёв. В основе методической системы лежит итерационный подход к разработке программного обеспечения, который предполагает написание отдельных мини-проектов (итераций) с последующим их объединением. В ходе обучения студенты создают небольшие программы для поддержки графического интерфейса пользователя с использованием среды Delphi [10, с. 46].

Другой подход к реализации метода проектов – коллективную дистанционную разработку с использованием системы управления обучением Moodle – предлагают В.Я. Талгатова и О.И. Богомолова. Студенты должны написать библиотеку классов «Геометрия» на языке C# в среде Visual C# Express Edition. Каждый из них должен самостоятельно разработать собственный

класс. На следующем этапе выполняется совместное тестирование полученных классов, проверка на наличие алгоритмических и математических ошибок, оценка соответствия различным критериям (время выполнения, полнота описания класса и т.д.). С этой целью применяется блок обмена сообщениями системы Moodle. На последнем этапе разработки производится объединение созданных классов в библиотеку, тестирование ее работы и добавление недостающих методов. Как указывают авторы, такой подход позволяет продемонстрировать преимущества объектно-ориентированной парадигмы разработки, а также формирует у студентов навыки коллективной разработки программных продуктов [13, с. 658-660].

В.Л. Дмитриев предлагает для изучения ООП применять поэтапный подход к разработке классов (на примере графических объектов – прямоугольников) в среде Delphi, сущность которого заключается в создании и последовательном совершенствовании класса, что позволяет продемонстрировать преимущества объектно-ориентированного подхода. На каждом этапе создания класса и его экземпляров добавляются новые методы, что позволяет неограниченно усложнять задачу [5, с. 71-76].

Интересный методический прием, который также позволяет показать преимущества ООП, предлагает Д.Н. Лавров: студенты должны решить одну и ту же задачу нахождения корней квадратного уравнения с использованием императивного и объектно-ориентированного подходов. Несмотря на большие затраты времени и сложность, становится понятно, что во втором случае будет намного проще усовершенствовать программу в дальнейшем [6, с. 179-181].

Выводы и перспективы дальнейших исследований. Проведенный анализ существующих подходов и собственный опыт преподавания дисциплины „Объектно-ориентированное программирование” в университете позволяет сделать вывод о том, что изучению ООП должно предшествовать овладение методологией структурного программирования в курсе „Программирование”. Начинать изучение объектно-ориентированного подхода нужно с формирования умений использовать готовые классы в одной из сред визуальной разработки, после чего переходить к теоретическим основам ООП и далее к практическому применению рассмотренных

механизмов на примере достаточно крупных программных проектов.

СПИСОК ЛИТЕРАТУРЫ

1. Агальцов В.П. Языки программирования и их преподавание в учебных заведениях / В.П. Агальцов // Ученые записки Российского государственного социального университета. – 2009. – Выпуск № 2. – С. 186-193.
2. Алексеевский П.И. Обучение программированию студентов на основе методологии унифицированного процесса разработки программного обеспечения / П.И. Алексеевский // Педагогическое образование в России. – 2014. – № 8. – С. 150-153.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд.: Пер. с англ. / Г. Буч, Роберт А. Максимум и др. – М.: ООО «ИД. Вильямс», 2008. – 720 с.
4. Волкова И.А. Основы объектно-ориентированного программирования. Язык программирования С++. Учебное пособие для студентов 2 курса / И.А. Волкова, А.В. Иванов, Л.Е. Карпов. – М.: Издательский отдел факультета ВМК МГУ, 2011. – 112 с.
5. Дмитриев В.Л. Поэтапная разработка классов в среде Delphi при обучении программированию / В.Л. Дмитриев // Смалта. – 2014. – № 3. – С. 70-76.
6. Лавров Д.Н. От императивного к объектно-ориентированному программированию вместе с Java и NetBeans: объектная декомпозиция и инкапсуляция / Д.Н. Лавров // Математические структуры и моделирование. – 2009. – Вып. 20. – С. 178-190.
7. Ломазова И.А. Почему Java? О выборе языка программирования при обучении будущих учителей информатики / И.А. Ломазова // Ученые записки российского государственного социального университета. – 2008. – № 5(61). – С. 21-23.
8. Мясникова Н.А. Проблемы изучения объектных технологий в ВУЗе / Н.А. Мясникова // Объектные системы. – 2011. – № 1(3). – С. 86-90.
9. Объектно-ориентированное программирование [Электронный ресурс]: учеб.-метод. комплекс по дисциплине для студентов специальности 230201 "Информационные системы и технологии" всех форм обучения : самост. учеб. электрон. изд. / Сыкт. лесн. ин-т; сост.: Л.Р. Камалова. – Электрон. дан. – Сыктывкар: СЛИ, 2012. – Режим доступа: <http://lib.sfi.komi.com>. – Загл. с экрана.
10. Паламарчук Л.Н. Об использовании мини-проектов в обучении студентов-программистов / Л.Н. Паламарчук, В.Н. Гладков, И.А. Гринёв // Вестник южно-уральского профессионального института. – 2013. – № 3(12). – С. 45-50.
11. Робоча програма навчальної дисципліни "Об'єктно-орієнтоване програмування" для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання / Укл.: Ю.Е. Парфонов, О.В. Щербаков, В.М. Федорченко. – Х.: Вид. ХНЕУ, 2012. – 48 с. (Укр. мова).
12. Синельникова И.В. Проблемы изучения объектно-ориентированного программирования в высших учебных заведениях / И.В. Синельникова, Л.А. Калиникова // Объектные системы. – 2010. – № 2(2). – С. 72-75.
13. Талгатова В.Я. Использование метода коллективной разработки в дистанционном обучении программированию в техническом вузе / В.Я. Талгатова, О.И. Богомолова // Образовательные технологии и общество. – 2014. – Выпуск № 4. Том 17. – С. 655-661.