

Балабекова М.Ж., Сапиева Г.Е.

*Казахский национальный аграрный университет*

## ОСНОВНЫЕ ПОДХОДЫ К ОБУЧЕНИЮ СТУДЕНТОВ ОБЪЕКТНО-ОРИЕНТИРОВАННОМУ ПРОГРАММИРОВАНИЮ И ПРОЕКТИРОВАНИЮ

### **Аннотация**

Информатизация общества, являясь доминирующей тенденцией развития цивилизации, диктует новые требования к образованию. Информатизация системы образования Республики Казахстан направлена на повышение качества содержания учебного процесса, также внедрение и развитие новых информационных технологий во всех видах деятельности, в том числе инженерного образования и современных технических специальностей. Главной целью образовательных реформ в Казахстане является адаптация системы образования к новой социально-экономической среде. Президентом Казахстана была также поставлена задача овладения республикой в число пятидесяти наиболее конкурентоспособных стран мира. Совершенствование системы образования играет важную роль в достижении этой цели.

**Ключевые слова:** программирование, образование, информатика.

### **Введение**

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.

## Материалы и методы

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 много семестровые объединяющие проекты, которые обеспечат "лучшее обучение, разработку новых курсов, факультетские исследования" [2].

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 сложно-го 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.

## **Результат исследований**

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, как универсального средства моделирования проектных решений, 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 до минимума 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.

## **Вывод**

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.

## Литература

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», 2010г. - 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.

Балабекова М.Ж., Сапиева Г.Е.

## ОБЪЕКТИЛІ-БАҒЫТТАЛҒАН ПРОГРАММАЛАУ ЖӘНЕ ЖОБАЛАУДЫ ОҚЫТУДА БАСТЫ ЖОЛДЫ ҚОЛДАНУ

*Түйін* Бұл мақалада объектілі-бағытталған программалау туралы түсініктемелері қарастырылған. Қазіргі уақытта информатикада ең басты әдіснама объектілі-бағытталған программалау болып табылады.

*Кілт сөздер:* бағдарламалау, білім, информатика.

MAIN APPROACHES TO TEACHING STUDENTS OBJECT-ORIENTED  
PROGRAMMING AND DESIGN

*Resume* The article deals with the formation of the concept of object-oriented programming. At present, the methodology of object-oriented programming is the leader in computer science.

*Keywords:* programming, education, science.

ӘОЖ 004.422

**Бегасилова А.Е, Тенгаева А.А.**

*Қазақ ұлттық аграрлық университеті*

RATIONAL ROSE ҚҰРАЛЫНДА АҚПАРАТТЫҚ  
ЖҮЙЕЛЕРДІ ЖОБАЛАУ

**Аңдатпа**

Бұл мақалада ақпараттық жүйелерді жобалаудың құралдар жинағы CASE-технологиялардың бірі Rational Rose құралында университет курстарын есепке алу жүйесі жобаланды.

**Кілт сөздер:** CASE-технология, CASE-құралы, ақпараттық жүйе, модель, Rational Rose жүйесі, Unified Modeling Language (UML) тілі.

**Кіріспе**

CASE-құралдарына байланысты ақпараттар біздің ойымызша кәдімгі бір программалаумен тығыз байланысты құрал сияқты. АҚШ-да бәсекелестердің күшті болуына байланысты, CASE-құралдар программалық қамтамасыз етуді өңдеуші фирмаларды басып отыру үшін қолданылады. CASE-құралдардың дамуы объектілі-бағытталған ПҚ өңдеу технологияларына байланысты [1].

Бүгінде алдыңғы қатардағы CASE-жүйе болып Rational Rose корпорациясымен шығарылған Rational Software алынады. Rational Rose жүйесі Unified Modeling Language (UML) тілін қолдана отырып модельдер құру үшін пайдаланылады. Айтайық, UML объектілі-бағытталған стандартты тілі болып қалыптасуы Rational Software-нің арқасында деп айтсақ та артық емес, ол UML-ді қолданылатын программалық өнімдерді шығарып, қана қоймай, сонымен қатар UML тілінің спецификациясын құратын және жаңартатын CORBA таратылған есептеулермен байланысты технологиялар Object Management Group (OMG)-пен де жұмыс жасайды және Rational компаниясында UML тілін және объектілі-бағытталған өндеуді құрушылар жұмыс жасайды [2].

**Негізгі бөлім**

UML – бұл көрнекілеу, спецификациялау, құрастыру және бағдарламалық жүйелердің артефактілерін құжаттау тілі. UML тілінің сөздігі мен ережесі жақсы анықталған модельді қалай құрып және қалай оқуды түсіндіреді, бірақ та қандай жағдайда қандай модельдерді құру керектігі туралы хабарламайды. Жүйенің кейбір ерекшеліктерін бәрінен бұрын мәтін түрінде модельдеуге болады, екінші біреулерін – графикалық түрінде модельдеуге болады. Шын мәнісінде барлық қызықты жүйелерде бір ғана бағдарламалау тілінің көмегімен беру мүмкін емес құрылымдар бар болады. Мұндайда, UML– екінші белгіленген мәселелерді шешуге мүмкіндік беретін графикалық тіл [3].