

УДК 378.147:004.942
DOI 10.17513/snt.39712

ПОДГОТОВКА ПО ПРОГРАММИРОВАНИЮ БАКАЛАВРОВ НАПРАВЛЕНИЯ «ПРОГРАММНАЯ ИНЖЕНЕРИЯ» С УЧЕТОМ КОНЦЕПЦИИ КОМПЛЕКСНОГО ПОДХОДА К ИНЖЕНЕРНОМУ ОБРАЗОВАНИЮ

Лавина Т.А., Мытникова Е.А., Ярускина Е.Т.

ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова», Чебоксары,
e-mail: mylene0210@mail.ru

Готовность к профессиональной деятельности в условиях цифровизации всех сфер экономики достигается у IT-специалистов через формирование в процессе обучения профессиональной компетентности. В статье описан опыт использования концепции комплексного подхода к инженерному образованию – CDIO – при обучении программированию бакалавров направления подготовки «Программная инженерия». Выявлено противоречие между возможностью реализации концепции CDIO и неразработанностью проблемы формирования профессиональных компетенций у программных инженеров в процессе изучения программирования с учетом CDIO-цикла. Авторами проанализированы задачи, решаемые в рамках предметной области «программная инженерия», раскрыт потенциал концепции CDIO для обучения программированию бакалавров по направлению подготовки «Программная инженерия». Проанализированные тенденции в области обучения программированию IT-специалистов, исследование рынка труда позволили сделать вывод о необходимости изучения объектно-ориентированного программирования с учетом современных образовательных технологий. Также приведены примеры разработанных практико-ориентированных заданий по дисциплине «Объектно-ориентированное программирование», решаемых с использованием CDIO-цикла. Представлены результаты педагогического эксперимента, позволяющие сделать вывод об эффективности формирования деятельностного компонента профессиональной компетентности при использовании концепции CDIO в процессе обучения объектно-ориентированному программированию. Перспективами дальнейшего исследования является применение CDIO при обучении всем профессиональным дисциплинам программы подготовки бакалавров программной инженерии.

Ключевые слова: программная инженерия, CDIO, объектно-ориентированное программирование, профессиональная компетентность

BACHELOR'S DEGREE IN PROGRAMMING IN THE DIRECTION OF «SOFTWARE ENGINEERING», TAKING INTO ACCOUNT THE CONCEPT OF AN INTEGRATED APPROACH TO ENGINEERING EDUCATION

Lavina T.A., Mytnikova E.A., Yaruskina E.T.

Chuvash State University named after I.N. Ulyanov, Cheboksary, e-mail: mylene0210@mail.ru

Readiness for professional activity in the conditions of digitalization of all spheres of the economy is achieved by IT specialists through the formation of professional competence in the process of training. The article describes the experience of using the concept of an integrated approach to engineering education – CDIO – when teaching programming to bachelors in the field of training “Software Engineering”. The contradiction between the possibility of implementing the CDIO concept and the undevelopment of the problem of the formation of professional competencies among software engineers in the process of studying programming taking into account the CDIO cycle is revealed. The authors analyzed the tasks solved within the framework of the subject area “software engineering”, revealed the potential of the CDIO concept for teaching programming to bachelors in the field of training “Software Engineering”. The analyzed trends in the field of programming training for IT specialists, labor market research, allowed us to conclude that it is necessary to study object-oriented programming taking into account modern educational technologies. Examples of developed practice-oriented tasks in the discipline “Object-oriented programming”, solved using the CDIO cycle, are also given. The results of the pedagogical experiment are presented, which allow us to conclude about the effectiveness of the formation of the activity component of professional competence when using the CDIO concept in the process of teaching object-oriented programming. The prospects for further research are the use of CDIO in teaching all professional disciplines of the Bachelor's degree program in software engineering.

Keywords: software engineering, CDIO, object-oriented programming, professional competence

Программная инженерия является инженерной областью знаний, которая охватывает все аспекты создания программного обеспечения: от определения системных требований на начальном этапе разработки до развертывания и использования программного обеспечения. В рамках программной инженерии реализуется научный

подход, основанный на опыте, для поиска эффективных и экономичных решений при разработке программных продуктов.

В контексте развития инженерного образования в России вузы – флагманы подготовки кадров для высокотехнологичного производства (Московский физико-технический институт, Томский политехнический

университет, Санкт-Петербургский политехнический университет Петра Великого и др.) применяют лучшие международные образовательные технологии, такие как CDIO и STEM, разрабатывая собственный подход к их реализации. В китайских университетах и колледжах адаптированная концепция комплексного подхода к инженерному образованию успешно реализуется для обучения студентов IT-направлений.

Цель исследования состоит в теоретическом обосновании и практической реализации концепции CDIO в процессе подготовки по программированию бакалавров программной инженерии.

Гипотеза исследования основана на предположении о том, что применение концепции CDIO в процессе обучения программированию позволит эффективно формировать деятельностный компонент профессиональной компетентности бакалавров программной инженерии.

Задачи исследования:

- проанализировать задачи, решаемые в рамках предметной области «программная инженерия»;
- проанализировать потенциал концепции CDIO для обучения программированию бакалавров по направлению подготовки «Программная инженерия»;
- разработать практико-ориентированные задания по дисциплине «Объектно-ориентированное программирование», решаемые с использованием CDIO-цикла;
- экспериментально проверить эффективность формирования деятельностного компонента профессиональной компетентности при использовании концепции CDIO в процессе обучения объектно-ориентированному программированию.

Материалы и методы исследования

В исследовании принимали участие студенты второго курса бакалавриата по направлению подготовки 09.03.03 «Программная инженерия» (профиль «Управление разработкой программных проектов») Чувашского государственного университета имени И.Н. Ульянова» (г. Чебоксары). Эксперимент длился с 2019 по 2023 г., всего в нем приняло участие 78 чел.

Выбор бакалавров второго курса был обусловлен тем, что на первом курсе они уже изучили основы программирования в рамках дисциплины «Программирование» и получили представление об областях знаний, входящих в программную инженерию, в рамках дисциплины «Основы программной инженерии», что способствует пониманию студентами применения за-

дач данной дисциплины в курсе объектно-ориентированного программирования.

Методы исследования: анализ научных трудов в области CDIO, метод экспертных оценок, статистический анализ данных.

Результаты исследования и их обсуждение

Программная инженерия является быстро развивающейся областью в связи с постоянным ростом потребностей в новых и усовершенствованных программных решениях в различных сферах, включая бизнес, образование, медицину, науку, игровую индустрию и др.

Приведем примеры задач, решаемых в рамках предметной области «Программная инженерия»:

- разработка программного обеспечения для решения конкретных задач, таких как системы управления базами данных, мобильные приложения, веб-приложения, игры и т.д.;
 - проектирование архитектуры программных продуктов, оптимизация их производительности и масштабируемости;
 - тестирование программного обеспечения, включая разработку тестовых сценариев, автоматизацию тестирования и анализ результатов тестирования;
 - управление проектами разработки программного обеспечения, включая планирование, контроль бюджета и сроков, организацию команды разработчиков и взаимодействие с заказчиками;
 - работа с современными технологиями и инструментами, такими как машинное обучение, блокчейн, облачные вычисления, DevOps и т.д.;
 - разработка систем безопасности и защиты данных, включая анализ угроз, проектирование и внедрение механизмов защиты и обеспечение безопасности пользователей;
 - разработка программных продуктов, соответствующих международным стандартам и регулятивным требованиям.
- Вышеизложенное позволяет сделать вывод, что IT-специалисты, выпускники по направлению подготовки «Программная инженерия», могут заниматься анализом требований, проектированием и конструированием программного обеспечения, тестированием и отладкой приложений, а также их внедрением. Они могут работать в различных сферах разработки программного обеспечения, включая мобильные приложения, веб-приложения, системы управления базами данных, а также область научных исследований и разработок. Программные инженеры, помимо этого,

должны обладать навыками коммуникации и управления проектами, так как разработка программного обеспечения обычно осуществляется в команде и требует тесного взаимодействия с другими специалистами.

В настоящее время в России не сложилось единого мнения по вопросу содержания подготовки студентов по направлению «Программная инженерия». Обучение специалистов в области программной инженерии началось в нашей стране с 2011 г., а в настоящее время более 100 вузов России осуществляют подготовку по этому направлению. ФГОС ВО 3++ по направлению бакалавриата 09.03.04 содержит требования к подготовке в части указания универсальных и общепрофессиональных компетенций, профессиональные компетенции разрабатываются вузом самостоятельно на основе профессиональных стандартов. Так же самостоятельно вузом определяются индикаторы достижения профессиональных компетенций.

Поскольку обучение программированию – одна из основных задач подготовки бакалавров по направлению «Программная инженерия», проанализируем возможности реализации концепции CDIO в аспекте изучения программирования. Аббревиатура концепции состоит из названий начальных букв слов *Conceive* – задумать, *Design* – спроектировать, *Implement* – реализовать, *Operate* – эксплуатировать.

Концепция CDIO представляет собой интегрированный подход к обучению инженеров [1, 2] и охватывает весь жизненный цикл инженерных систем, в том числе информационных. Мы согласны с выводами автора статьи об актуальности дальнейших изысканий в области перспектив развития инженерного образования в отечественных технических вузах [3]. В настоящее время комплексный подход к инженерному образованию находит все большее применение в области программной инженерии.

CDIO, как и программная инженерия, предлагает системный подход, что способствует развитию у студентов не только технических навыков, но и личностных качеств, необходимых для успешной работы в ИТ-индустрии. Обучение на основе реальных проблем и проектов помогает развить навыки и умения, необходимые для решения практических задач в будущей профессиональной деятельности. Интегрированный подход к обучению, связывающий знания и умения из разных областей, помогает студентам получить более глубокое понимание тем, с которыми они работают. Работа в командах позволяет студентам развить навыки коммуникации, лидерство,

умение работать в коллективе. Ориентация на результаты и оценка не только знаний, но и умения применять их на практике позволяют оценить подготовленность студента к решению профессиональных задач. При этом непрерывное совершенствование учебно-методического обеспечения позволяет сохранять актуальность образовательной программы и соответствовать требованиям рынка труда [4, 5].

Наглядно продемонстрируем связь между программной инженерией и CDIO, применяя ее в контексте разработки программного обеспечения, предназначенного для автоматизации производственных процессов. Основной задачей является оптимизация производственного процесса с целью повышения его эффективности и снижения вероятности возникновения ошибок, связанных с влиянием человеческого фактора. В этом контексте может быть применен подход к разработке программного обеспечения, базирующийся на концепции объектов, взаимодействующих друг с другом и образующих сложные структуры, что соответствует методологии объектно-ориентированного программирования.

1. *Conceive*. На этой стадии происходит выявление и анализ требований к системе, выделение отдельных групп классов и взаимодействия их объектов между собой. Результаты этого процесса будут отражены в спецификации требований.

2. *Design*. На этой стадии происходит построение архитектуры системы. Для проектирования компонентов подсистем, представляющих различные составляющие производственного процесса (конвейеры, роботы и т.д.), можно использовать объектно-ориентированное проектирование с использованием нотации графических языков моделирования. Таким образом, абстракция, инкапсуляция, наследование и полиморфизм помогут в создании гибкой и модульной системы.

3. *Implement*. На этой стадии происходит реализация системы. Использование объектно-ориентированных языков программирования позволяет написать код, создающий классы и объекты, спроектированные на предыдущей стадии.

4. *Operate*. На этой стадии происходит внедрение и эксплуатация системы. Предварительно требуется провести процессы, связанные с верификацией и валидацией, на соответствие спецификации требований. Для облегчения процессов разработки системы необходимо использовать case-средства на всех этапах ее создания, а для управления ее развитием – применять современные бизнес-процессы.

Таким образом, CDIO-цикл может быть использован в рамках объектно-ориентированного программирования для разработки программного обеспечения, управляющего инженерными системами. Студенты должны получить навыки реализации каждой стадии жизненного цикла [6, 7] процесса разработки программных систем. Например, на стадии концептуализации обучающиеся должны уметь определять требования к программному обеспечению. Они учатся понимать и анализировать реальные проблемы и потребности в конкретной области, что способствует формированию у них инновационного мышления и креативности. Студенты должны научиться генерировать идеи, выделять ключевые факторы, планировать реализацию, оценивать риски и принимать обоснованные решения. На стадии проектирования приобретаются навыки построения моделей системы, студенты учатся разрабатывать архитектуру проекта, строить UML-диаграммы. На стадии реализации у обучающихся вырабатываются навыки написания кода, они используют свои знания языков программирования для того, чтобы научиться разрабатывать информационные системы, тестировать их на соответствие требованиям, идентифицировать и устранять ошибки. На стадии внедрения и эксплуатации улучшается понимание всех процессов создания программного продукта, а также его сопровождения. Студенты должны научиться управлять системами, проводить их мониторинг и обновление.

Практико-ориентированное обучение представляет собой методический подход, в основе которого лежит активное вовлечение студентов в решение реальных практических задач, максимально соответствующих специфике их будущей профессиональной деятельности. В рамках данной методики обучения студенты получают возможность не только углубить свои теоретические знания в соответствующей области, но и на практике осуществить их в условиях, наиболее близких к реальным. Данный подход основывается на относительном преобладании практической составляющей обучения, за счет чего создается близкий к реальному опыт работы и готовятся специалисты, действительно готовые к решению задач в рамках своей области. Кроме того, практико-ориентированное обучение не только предоставляет студентам возможности для практического применения знаний, но и значительно повышает их мотивацию. Большинство студентов стремится увидеть результаты своей работы в реальной жизни и убедиться в важности и полезности получаемых знаний. Существуют

различные формы практико-ориентированного обучения: кейс-стади, лабораторные работы, практические занятия, моделирование реальных ситуаций и т.д.

Основной целью практико-ориентированного обучения является формирование у студентов профессиональной компетентности и подготовка их к решению реальных профессиональных задач в будущей карьере [8]. CDIO – это концепция, ориентированная именно на практику [9, 10], что помогает студентам развивать навыки и умения, необходимые для решения реальных профессиональных задач.

Требования, вытекающие из концепции CDIO, существенно влияют на содержание обучения программированию в рамках направления подготовки «Программная инженерия». Поэтому нами был разработан комплекс практико-ориентированных заданий по дисциплине «Объектно-ориентированное программирование», решаемых с использованием CDIO-цикла. Приведем примеры таких заданий:

Тема лабораторного занятия: Создание простого класса. Конструкторы и деструктор.

Задание 1. Журнал оценок

Создать класс *Студент*, который будет представлять студента и содержать информацию о его имени, фамилии и оценках по различным предметам. Задание включает в себя создание конструкторов и деструктора класса, методов для добавления и получения оценок, вычисления среднего балла и определения успеваемости студента. Создать объекты класса *Студент* для каждого студента и вывести средние результаты их успеваемости.

Задание 2. Банковский счет

Создать класс *Банковский счет*, который будет представлять банковский счет и содержать информацию о его номере, владельце и балансе. Задание включает в себя создание конструкторов и деструктора класса, методов класса для пополнения и снятия денег со счета, а также метода для вывода текущего баланса. Создать объекты класса *Банковский счет* для каждого пользователя, продемонстрировать функции пополнения, снятия денег и проверки баланса.

Тема лабораторного занятия: Простое наследование.

Задание 1. Сотрудники

Создать базовый класс *Сотрудник*, который будет представлять общие свойства и методы для различных видов сотрудников. Затем создать производные от него классы: *Менеджер*, *Инженер* и *Учитель*. Каждый из производных классов должен иметь свои собственные уникальные свойства и методы, связанные с конкретной должностью.

Создать объекты этих классов и продемонстрировать выполнение методов, связанных с их должностями.

Задание 2. Игровые персонажи

Создать базовый класс *Игровой_персонаж*, который будет представлять общие свойства и методы для различных видов игровых персонажей. Затем создать производные от него классы: *Игрок*, *Враг* и *Бот*. Каждый из производных классов должен иметь свои собственные уникальные свойства и методы, связанные с конкретным типом персонажа. Создать объекты этих классов и продемонстрировать выполнение методов, связанных с этими персонажами.

Тема лабораторного занятия: Множественное наследование.

Задание 1. Товары

Создать базовый класс *Товар*, который будет представлять общие свойства и методы для различных товаров. Например, свойствами могут быть название и цена товара. Создать производные от *Товар* классы *Товар_со_скидкой* и *Товар_с_доставкой*, которые будут представлять функциональность скидки и доставки соответственно. Создать класс *Товар_со_скидкой_и_доставкой*, наследующий от классов *Товар_со_скидкой* и *Товар_с_доставкой*, чтобы объединить функциональность скидки и доставки. Создать объекты класса *Товар_со_скидкой_и_доставкой* и применить методы для расчета скидки и оформления доставки товара.

Задание 2. Аккаунты социальных сетей

Создать базовый класс *Аккаунт*, который будет представлять общие свойства и методы для различных аккаунтов в социальных сетях. Создать производные от *Аккаунт* классы *VK_Аккаунт* и *OK_Аккаунт*, которые будут представлять специфичные методы и свойства для каждой социальной сети. Создать класс *Union_Аккаунт*, наследующий от классов *VK_Аккаунт* и *OK_Аккаунт*, чтобы объединить функциональность обоих аккаунтов. Создать объекты класса *Union_Аккаунт* и применить методы, связанные с работой в обеих социальных сетях.

Исследование проводилось в период с 2019 по 2023 г. В эксперименте приняло участие 25, 24 и 29 бакалавров очной формы обучения по направлению подготовки «Программная инженерия» соответственно в 2020–2021 уч. году, 2021–2022 уч. году, 2022–2023 уч. году. На формирующем этапе проведенного нами педагогического эксперимента осуществлялась проверка эффективности применения концепции CDIO для формирования деятельностного компонента профессиональной компетентности у обучающихся по направлению подготовки

бакалавриата 09.03.04 «Программная инженерия» в процессе обучения объектно-ориентированному программированию.

Оценка уровня сформированности деятельностного компонента профессиональной компетентности была проведена путем экспертного оценивания практических заданий по дисциплине «Объектно-ориентированное программирование». Для оценки деятельностного компонента на основе индикаторов достижения профессиональных компетенций, разработанных нами ранее на констатирующем этапе эксперимента, мы подготовили «Листы экспертного оценивания».

Максимальное значение, которое могло быть получено за выполнение задания, составляло 32 балла. Для объективности работа каждого студента была оценена пятью экспертами. Итоговый получаемый балл рассчитывался как среднее арифметическое всех экспертов. При определении уровня сформированности (низкий, средний, высокий) мы использовали шкалу перевода баллов по В.П. Беспалько [11]: показатель от 70 до 100% от максимального значения является высоким уровнем (у нас это 22,4 баллов и выше), от 50 до 69% от максимального значения является средним уровнем (у нас это интервал от 16 до 22,3 баллов), ниже 50% от максимального значения является низким уровнем (у нас это менее 16 баллов).

Нами была сформулирована следующая нулевая гипотеза: большинством студентов будет достигнут высокий уровень сформированности деятельностного компонента профессиональной компетентности.

При сопоставлении эмпирического распределения с теоретическим мы использовали такой критерий, как критерий хи-квадрат. Чтобы принять или отвергнуть нулевую гипотезу, мы воспользовались таблицей критических значений хи-квадрата для уровней статистической значимости.

Приведем результаты оценивания работ экспертами (рис. 1–3).

Хи-квадрат эмпирическое во всех случаях попадало у нас в зону незначимости. Следовательно, можно было принять нулевую гипотезу.

По результатам экспериментальной работы (рис. 4) можно сделать вывод: низкий уровень сформированности деятельностного компонента не был зафиксирован, у части студентов (23 чел.) деятельностный компонент профессиональной компетентности был сформирован на среднем уровне, а большинство (55 чел.) показали высокий уровень сформированности деятельностного компонента профессиональной компетентности.

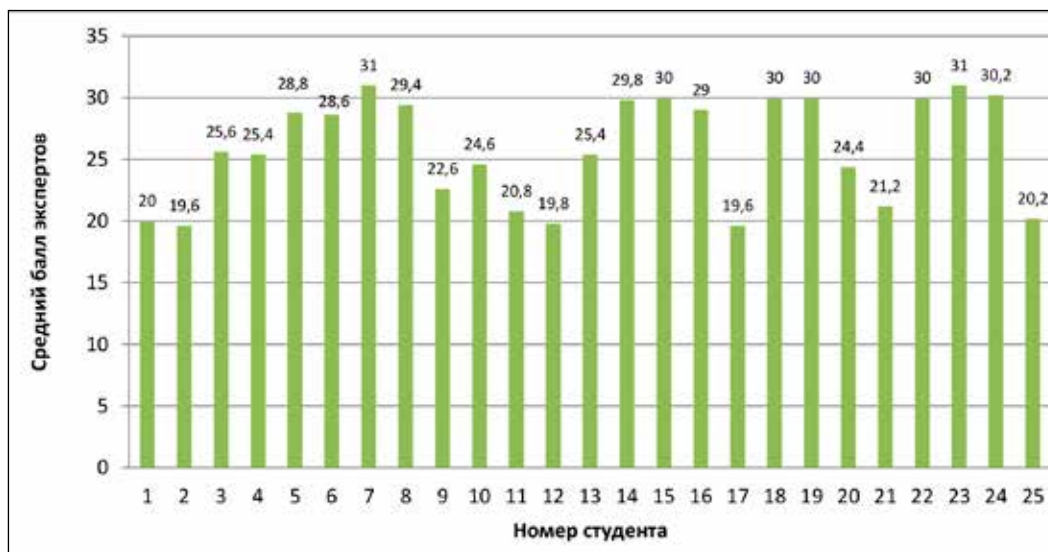


Рис. 1. Результаты проверки сформированности деятельностного компонента в 2020–2021 уч. году

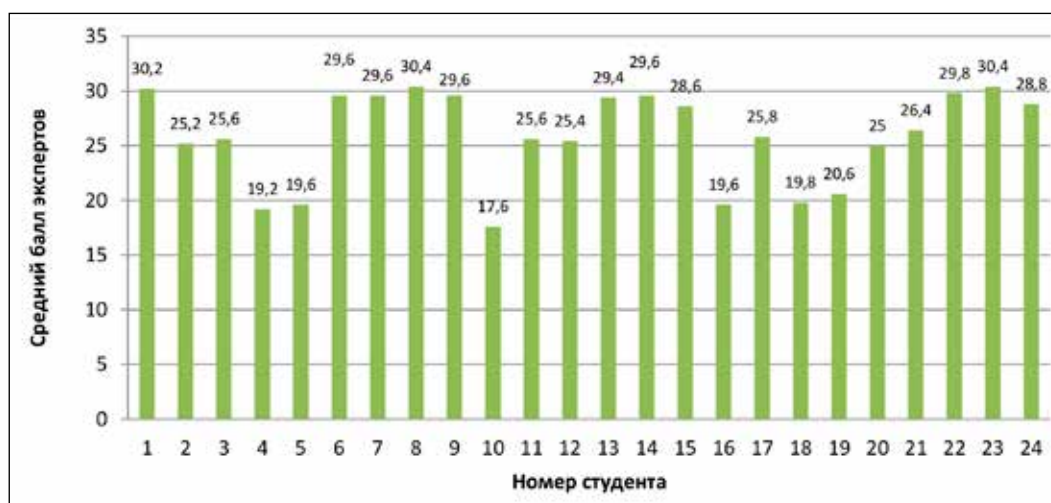


Рис. 2. Результаты проверки сформированности деятельностного компонента в 2021–2022 уч. году

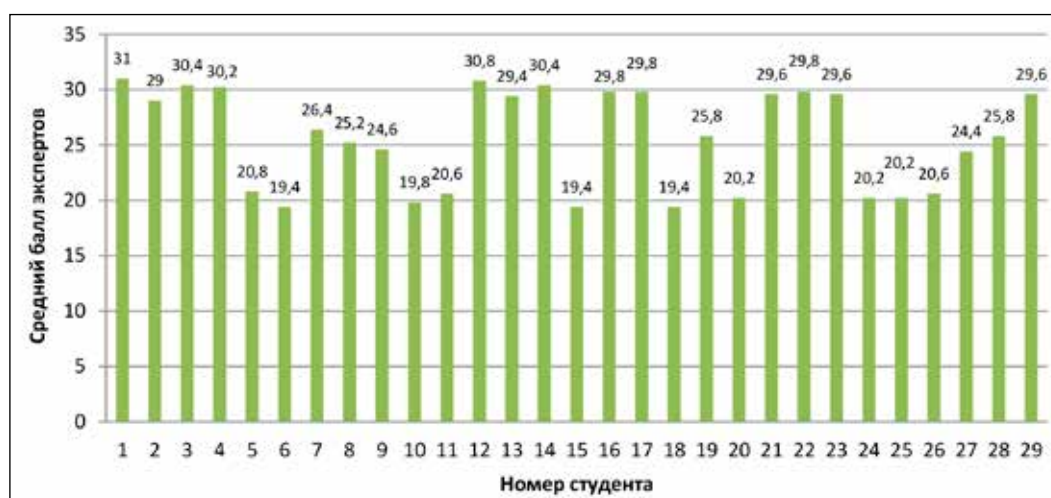


Рис. 3. Результаты проверки сформированности деятельностного компонента в 2022–2023 уч. году

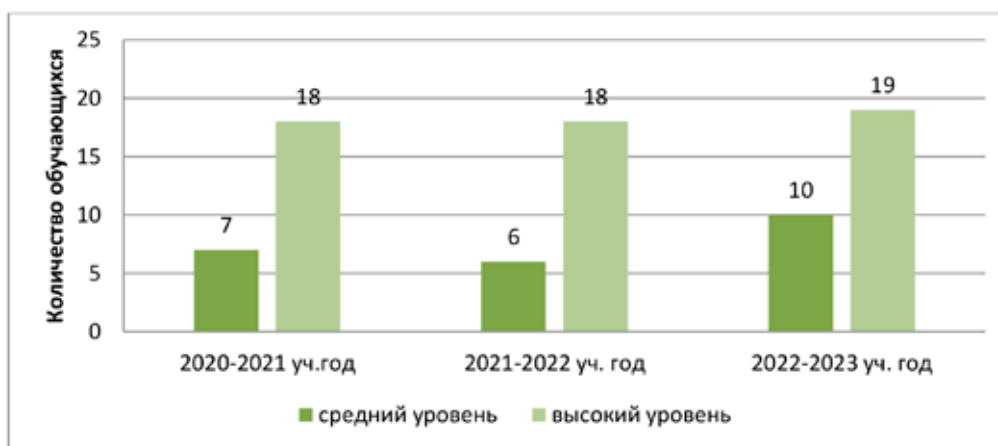


Рис. 4. Итоговые результаты проверки сформированности деятельностного компонента в 2020–2021, 2021–2022, 2022–2023 уч. годах

Заключение

Разработанные нами практико-ориентированные задания по дисциплине «Объектно-ориентированное программирование», решаемые с использованием CDIO-цикла, позволили развить навыки студентов в области программирования, что обусловлено применением их для реальных проектов.

Проведенный педагогический эксперимент показал эффективность использования концепции CDIO в процессе обучения объектно-ориентированному программированию для формирования деятельностного компонента профессиональной компетентности. Перспективы дальнейшего исследования заключаются в применении CDIO при обучении другим профессиональным дисциплинам основной образовательной программы подготовки бакалавров программной инженерии.

Список литературы

1. Кроули Э.Ф., Малмквист Й., Остлунд С., Бродер Д., Эдстрем К. Переосмысление инженерного образования. Подход CDIO / Пер. с англ., под науч. ред. А. Чучалина. Нац. исслед. ун-т «Высшая школа экономики». 2-е изд. 2019. [Электронный ресурс]. URL: <https://znanium.com/catalog/product/1040824> (дата обращения: 31.03.2023).

2. Чучалин А.И. Модернизация трехуровневого инженерного образования на основе ФГОС 3++ и CDIO++ // Высшее образование в России. 2018. № 4. С. 22–32.

3. Долженко Р.А. Концепция CDIO как основа инженерного образования: промежуточные итоги и направления дальнейшего использования в России // Известия УГТУ. 2017. Вып. 2 (46). С. 104–108. DOI: 10.21440/2307-2091-2017-2-104-108.

4. Чучалин А.И. Развитие кадрового потенциала российских вузов в области математики, информатики и цифровых технологий // Высшее образование в России. 2021. Т. 30. № 2. С. 9–20. DOI: 10.31992/0869-3617-2021-30-5-9-20.

5. Лавина Т.А., Мытникова Е.А., Давыдова О.В. Профессиональные компетенции бакалавров по направлению «Программная инженерия» // Вестник Череповецкого государственного университета. 2022. № 5 (110). С. 218–226. DOI: 10.23859/1994-0637-2022-5-110-18.

6. CDIO: An international initiative for reforming engineering education / K.F. Berggren, D. Brodeur, E.F. Crawley [et al] // World Transactions on Engineering and Technology Education. 2003. Vol. 2, Is. 1. P. 49–52.

7. Bankel J., Berggren K.F., Blom K., Crawley E.F., Wiklund I., Östlund S. The CDIO syllabus: a comparative study of expected student proficiency // European Journal of Engineering Education. 2003. Vol. 28, Is. 3. P. 297–315.

8. Лавина Т.А., Яруськина Е.Т. Подготовка преподавателей специальных дисциплин к использованию возможностей дистанционных технологий в процессе обучения менеджеров по персоналу (на примере LMS «MOODLE») // Современные проблемы науки и образования. 2014. № 3. URL: <https://science-education.ru/ru/article/view?id=13376> (дата обращения: 10.03.2023).

9. Malmqvist J., Edström K., Rosén A. CDIO Standards 3.0 – Updates to the Core CDIO Standards Proceedings of the International CDIO Conference. 2020. Vol. 16, Is. 1. P. 60–76.

10. Чучалин А.И. Адаптация the Core CDIO Standards 3.0 к высшему STEM-образованию // Высшее образование в России. 2021. № 2. С. 9–21. DOI: 10.31992/0869-3617-202130-2-9-21.

11. Беспалько В.П. Слагаемые педагогической технологии. М.: Педагогика, 1989. 192 с.