



ОБУЧЕНИЕ ИНФОРМАТИКЕ TEACHING COMPUTER SCIENCE

DOI 10.22363/2312-8631-2023-20-1-52-62

EDN: CCZGJZ

УДК 378

Научная статья / Research article

Совместное проектное обучение студентов ИТ-направлений подготовки на основе разделения задач в проекте и применения средств контроля версий

К.В. Рочев  , А.В. Семяшкина *Ухтинский государственный технический университет, Ухта, Российская Федерация* k@rochev.ru

Аннотация. *Постановка проблемы.* Актуальность работы обусловлена растущей необходимостью подготовки специалистов, обладающих компетенциям коллегиальной разработки программного обеспечения, диктуемой требованиями современной ИТ-индустрии. Цель исследования – сформировать во время учебного процесса условия, повторяющие частые ситуации при промышленной разработке программного обеспечения и позволяющие получить навыки коллективной разработки, близкие к реальному участию в производстве программного обеспечения. *Методология.* В процессе обучения студенты разрабатывают отдельные части общего проекта, встраивая свое решение в общую логику системы на основе реализации заданного преподавателем интерфейса класса. При этом описываются и применяются часто используемые шаблоны проектирования, такие как model-view-controller, стратегия и др. Дополненный проект синхронизируется между участниками с помощью системы контроля версий. В результате у студентов получается полноценное приложение, части которого написаны отдельными участниками. *Результаты.* Реализация подхода рассмотрена на примере игры «Шахматы», в которой каждый студент может написать код отдельной фигуры, указаны возможности применения подхода в рамках одного или нескольких занятий и потенциальные риски его использования. *Заключение.* Предложенный подход позволяет студентам получить опыт работы, приближенной к промышленному программированию, и освоить такие компетенции, как совместная разработка, работа в системах контроля версий и построение модульных систем.

Ключевые слова: программирование, проектный метод, шаблоны проектирования, командная разработка, шахматы, Git, Unity

История статьи: поступила в редакцию 8 июня 2022 г.; доработана после рецензирования 15 сентября 2022 г.; принята к публикации 17 октября 2022 г.

© Рочев К.В., Семяшкина А.В., 2023

This work is licensed under a Creative Commons Attribution 4.0 International License
<https://creativecommons.org/licenses/by-nc/4.0/legalcode>

Для цитирования: Рочев К.В., Семьяшкина А.В. Совместное проектное обучение студентов ИТ-направлений подготовки на основе разделения задач в проекте и применения средств контроля версий // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2023. Т. 20. № 1. С. 52–62. <http://doi.org/10.22363/2312-8631-2023-20-1-52-62>

Joint project-based training of IT students based on the use of version control tools and task separation in the project

Konstantin V. Rochev  , Anastasia V. Semyashkina 

Ukhta State Technical University, Ukhta, Russian Federation

 k@rochev.ru

Abstract. *Problem statement.* The relevance of the work lies in the proposed combination of methods and tools for application development during the educational process, which repeats frequent situations in industrial software development. The goal is to create conditions that allow to gain collective development skills close to real participation in software production. *Methodology.* During the training, students will write separate parts of the overall project, embedding their solution into the general logic of the system based on the implementation of the class interface set by the teacher. At the same time, frequently used design patterns are described and applied, such as model-view-controller, strategy, etc. The augmented project is synchronized between the participants using a version control system. As a result, students have a full-fledged application, parts of which are written by individual participants. *Results.* The implementation of the approach is considered on the example of the game “Chess”, in which each of the students can write the code of a separate figure; the possibilities of applying the approach within one or more classes and the potential risks of its use are indicated. *Conclusion.* The proposed approach allows students to gain work experience close to industrial programming and master such competencies as joint development, work in version control systems and building modular systems.

Keywords: programming, project method, design patterns, team development, Chess, Git, Unity

Article history: received 8 June 2022; revised 15 September 2022; accepted 17 October 2022.

For citation: Rochev KV, Semyashkina AV. Joint project-based training of IT students based on the use of version control tools and task separation in the project. *RUDN Journal of Informatization in Education*. 2023;20(1):52–62. (In Russ.) <http://doi.org/10.22363/2312-8631-2023-20-1-52-62>

Постановка проблемы. Большая потребность в специалистах, которые призваны обеспечить цифровизацию экономики, вызывает необходимость в новых формах, методах и средствах их подготовки, сближающих учебную и профессиональную деятельность и позволяющих сформировать их профессиональную компетентность на момент окончания вуза [1–3]. В современных условиях, диктующих необходимость цифровизации образования, некоторые преподаватели и вузы уже начинают использовать средства профессиональной

ИТ-среды, такие как системы контроля версий, в рамках реализации учебного процесса [4; 5]. Особенную актуальность средства командной разработки получают при реализации совместных проектов при дистанционном обучении, как и при удаленной работе, востребованность которых существенно возросла в последние годы [6–9].

Применение методов снижения связности кода, шаблонов проектирования, направленных на модульность программных проектов и систем контроля версий в рамках дисциплин компьютерного цикла, позволяет сместить акценты в процессе обучения с чисто учебных действий на решение профессиональных задач средствами, применяемыми в профессиональных сообществах, в командах разработчиков программного продукта [1; 2].

Цель исследования – изучить вопросы применения средств контроля версий в полноценном контексте их использования в подготовке ИТ-специалистов при работе над совместным проектом, что позволит обеспечить наиболее эффективный способ получения востребованных компетенций реализации коллективной разработки ПО [1; 10].

Методология. Обучение вопросам коллегиальной разработки проектов можно провести в рамках одного или нескольких занятий, желательно совмещенных в относительно небольшом временном периоде. В рамках подобного «хакатона» преподаватель выступает в роли ведущего программиста, который формирует архитектуру проекта и рассказывает о ней студентам. Пишет (а лучше показывает заранее написанные) основные части приложения, выделяет области приложения, реализацию которых можно легко делегировать, и располагает там интерфейсы для подключения будущих реализаций. После этого студентам предлагается произвести совместную доработку проекта, реализовав эти интерфейсы (под интерфейсом здесь и далее подразумевается чисто абстрактный класс, указывающий какие методы и свойства должны иметь его наследники).

В качестве проекта для совместной реализации может быть выбрано любое приложение, имеющее несколько сходных элементов с несколько отличающейся логикой. Например, хорошо подойдет игра «Шахматы», где такими элементами могут быть фигуры. В этом случае преподаватель демонстрирует, как устроена общая часть игры, а реализация логики и представления отдельных фигур предлагается студентам для индивидуального или группового исполнения, в зависимости от количества участников. Реализация отдельной фигуры является здесь достаточно понятной и компактной задачей.

Код проекта предоставляется в Git-репозитории, например в GitHub или Bitbucket [11]. Подключение к проекту предполагается с помощью любого Git-клиента, например Sourcetree. Первая версия проекта, реализованного в рамках занятий, проведенных по описываемой методике, представлена в открытом репозитории Bitbucket¹.

Структура проекта реализуется на основе MVC шаблона проектирования. Выделяются такие классы модели, как Игрок, Доска, Матч, Ход, Фигу-

¹ Репозиторий с проектом «Шахматы» для совместной реализации. URL: <https://itue@bitbucket.org/itue/mrchess.git> (дата обращения: 25.05.2022).

ра. При этом Фигура реализуется в виде абстрактного класса или интерфейса, от которого студенты будут наследовать конкретные фигуры, определяющие разрешение на ход.

Модель можно реализовать в отдельном проекте, подключив его к Unity в виде библиотеки классов (например, указав путь выхода сборки в папку проекта Unity: Assets/Dll). Это продемонстрирует разделение проекта на модули и обеспечит низкую связность кода модели и представления. А в случае реализации серверной части для сетевой игры, позволит подключить эту же библиотеку в серверный проект (например, на Asp.Net или Photon Server). С другой стороны, для облегчения работы по синхронизации этот шаг можно опустить.

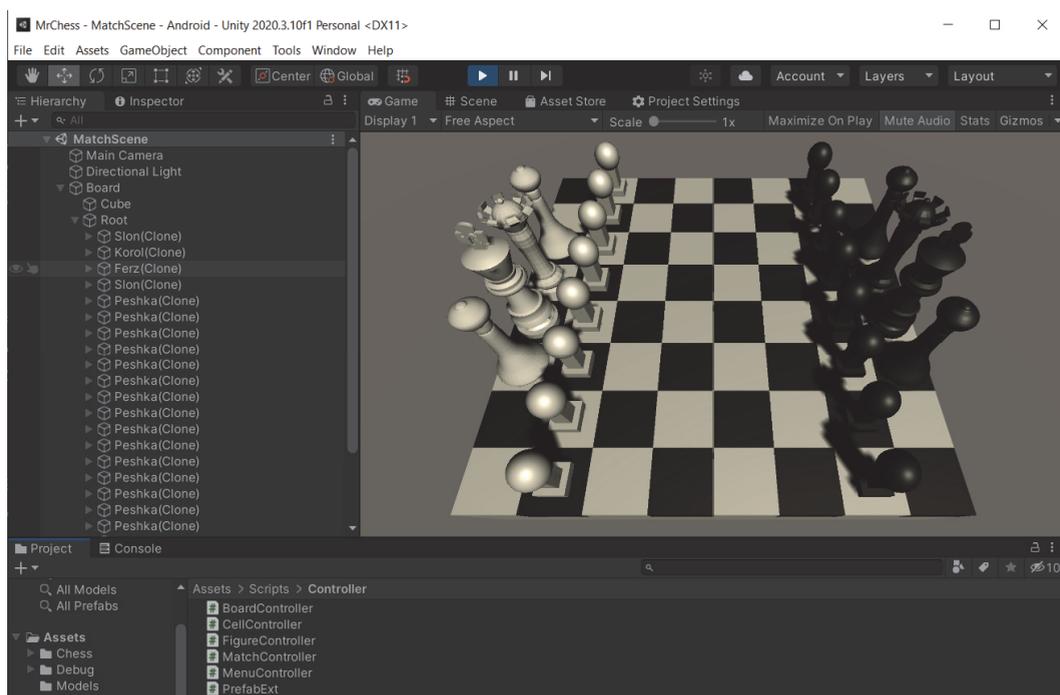


Рис. 1. Заготовка проекта – отдельные фигуры постепенно реализуются и добавляются студентами в общий проект с помощью Git

Figure 1. Project preparation – individual figures are gradually implemented and added by students to the overall project using Git

В качестве представления будут выступать префабы. Префаб – особый тип ассетов, позволяющий хранить GameObject со всеми компонентами и значениями свойств. Префаб выступает в роли шаблона для создания экземпляров хранимого объекта в сцене. Любые изменения в префабе немедленно отражаются и на всех его экземплярах, при этом можно переопределять компоненты и настройки для каждого экземпляра в отдельности².

По аналогии с моделью потребуются такие префабы, как Доска и Фигура. Префабы Фигур будут реализовываться отдельными участниками проекта либо из имеющихся примитивов Unity (куб, цилиндр, сфера), либо на основе им-

² Руководство Unity. URL: <https://docs.unity3d.com/ru/530/Manual/Prefabs.html> (дата обращения: 22.05.2022).

портируемых моделей, подготовленных в средствах 3D-моделирования (рис. 1). Префабы Игрок, Матч, Ход тоже могут быть реализованы, например, как части интерфейса игры.

Контроллерами в нашем случае будут скрипты Unity-проекта, обеспечивающие связку между классами модели и префабами. Реализуются как наследники MonoBehaviour, содержащие в себе ссылки на классы модели и подключаемые к соответствующим префабам. Основная задача контроллеров – обеспечить связку между представлением (префабами) и логикой. Можно отметить, что нам будет достаточно одного контроллера для всех имеющихся видов фигур, благодаря полиморфизму в реализации шаблона проектирования «Стратегия» (рис. 2).

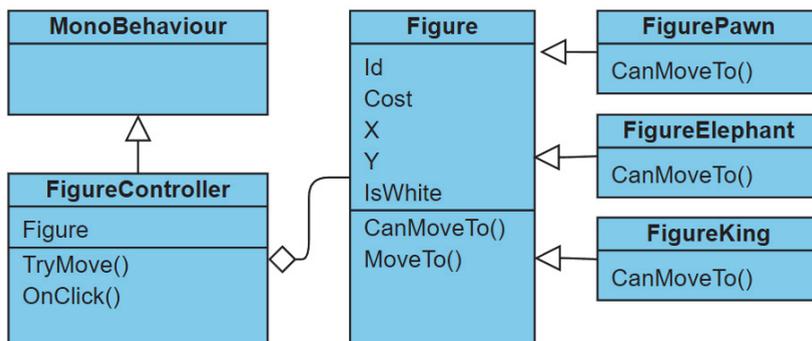


Рис. 2. Часть диаграммы классов проекта, иллюстрирующая взаимодействие модели и контроллера фигур
Figure 2. Part of the class diagram of the project illustrating the interaction between the model and the shape controller

Минимальная версия проекта реализуется с мультиплеером в формате попеременного хода участников, находящихся за одним устройством. В зависимости от времени, отводимого на данную тему, можно расширить проект дополнительными возможностями (рис. 3), такими как:

1. Реализация ИИ – можно, например, сделать набор ботов, предложив каждому из участников написать своего. При этом боты будут наследоваться от общего интерфейса, описывающего управление матчем.

2. Сетевое взаимодействие:

- а) регистрация аккаунта;
- б) матчмейкинг – запуск матчей;
- в) синхронизация ходов;
- г) построение рейтинга.

3. Монетизация приложений:

а) внутриигровые покупки, например: скины фигур, дополнительные режимы игры (12×12 ячеек и т. п.) – можно разделить реализацию отдельных покупок между участниками;

б) реклама: от Unity, с подключением агрегаторов – можно разделить подключение отдельных рекламных SDK к агрегатору между участниками.

4. Сборка и размещение в магазинах приложений.

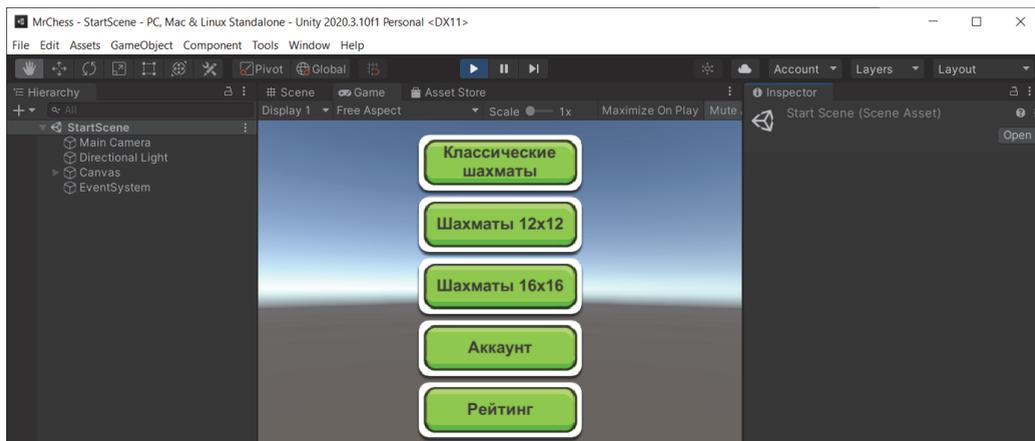


Рис. 3. Заготовка главного меню, предоставляющего доступ к различным функциям приложения

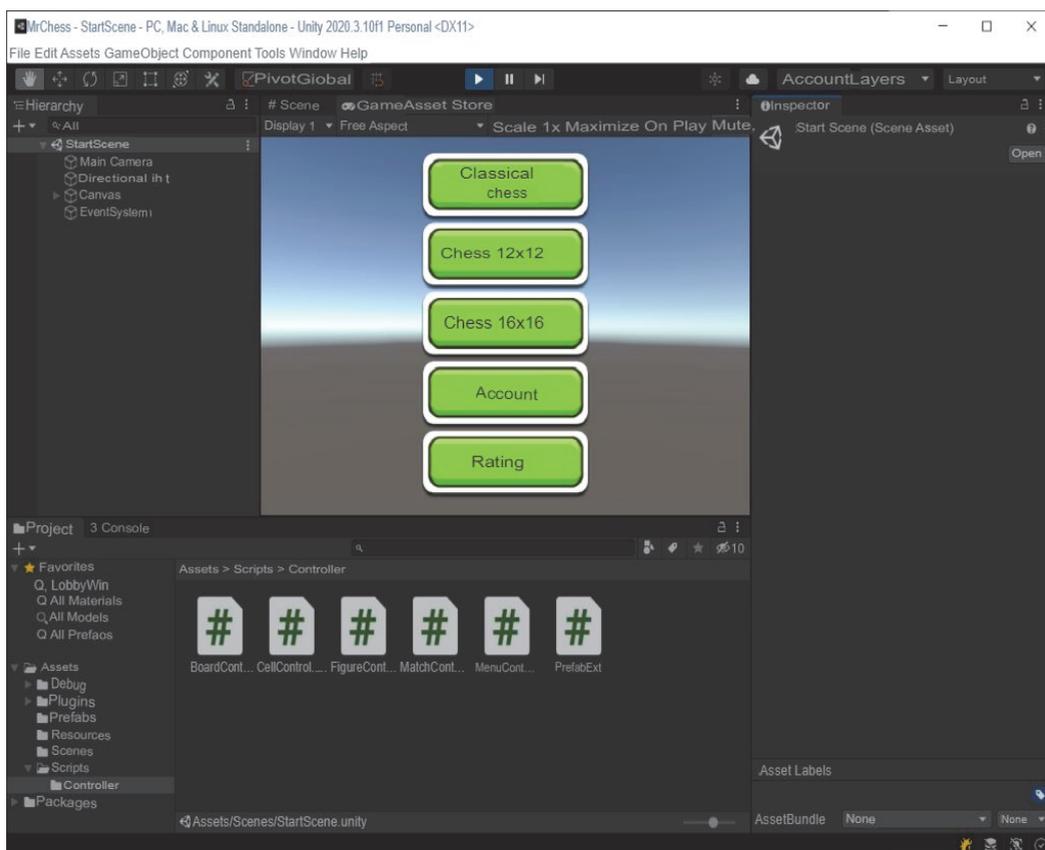


Figure 3. Draft main menu providing access to various functions of the application

Для улучшения контроля за выполнением проекта множеством студентов можно воспользоваться, например, google-таблицей, с отметками по шагам реализации проекта и имеющимися вопросами (рис. 4). При этом имеет смысл сделать таблицу доступной для редактирования всем участникам.

Кроме того, общий Git-репозиторий позволяет удобно просматривать и оценивать внесенные каждым студентом изменения и осуществлять взаимодействие по возникающим вопросам, что особенно полезно при дистанционном формате обучения.

D	E	F	G	H	I	J	K	L
Имеющиеся вопросы	Установлен гит	Установлен юнити	Скачан проект	Выбрана фигура	Реализована 3д модель фигуры	Реализована логика фигуры	Фигура добавлена в Git	Оценка
				Король				
	✓	✓	✓	Ладья				
	✓	✓	✓	Слон				
	✓	✓	✓	Ферзь				
	✓	✓	✓					

Рис. 4. Google-таблица с прогрессом участников проекта

D	E	F	G	H	I	J	K	L
Имеющиеся вопросы	Installed git	Installed unity	unity	The piece has been selected	Implemented a 3D model of the figure	of the figure implemented on the logic of the figure	Figure added in Git	Evaluation
				King				
	✓	✓	✓	Rook				
	✓	✓	✓	Bishop				
	✓	✓	✓	Queen				
	✓	✓	✓					

Figure 4. Google-table with the progress of the project participants

Результаты и обсуждение. Первоначальный уровень мотивации студентов легко создать, рассказав им о востребованности методов и технологий коллективной работы с кодом. Тем не менее существует риск затягивания проекта, который может привести к потере мотивации и вовлеченности студентов в процесс разработки, в результате чего часть студентов не пройдут своевременно весь путь, а это, в свою очередь может помешать собрать итоговый проект и продемонстрировать плодотворность коллективной работы. Во избежание этого целесообразно заранее проверить основные требования при подготовке к подобного рода занятиям. Во-первых, у студентов должны быть базовые знания в области программирования на выбранном языке. Во-вторых, программное и техническое обеспечение учебной аудитории должно поддерживать весь цикл разработки и синхронизации проекта: необходимы средства разработки одной версии на машинах преподавателя и студентов, Git-клиент, стабильный доступ в сеть, аккаунт в Git-репозитории. Можно, конечно, включить установку ПО в список задач, особенно если часть студентов использует свои ноутбуки. Например, Git-клиент установить можно, но процесс установки сред разработки легко может затянуться, а в случае с разными версиями и усложнить синхронизацию проекта.

Если брать в качестве примера именно шахматы, фигур может оказаться меньше, чем студентов. В этом случае можно либо объединить студентов в группы, применив, например, подход экстремального программирования [12], либо расширить список фигур дополнительными, например из «Омега-шахмат»³.

³ Омега-шахматы. URL: <https://omegachess.com/rules.htm> (дата обращения: 22.05.2022).

Основные сложности, возникшие при выполнении проекта

Проблема	Причина	Возможное решение
Долгая установка средств разработки	Реализация проекта на собственных ноутбуках	Предварительная задача на установку ПО (не помогло в 60 % случаев). Работа на аудиторном оборудовании с предустановленным ПО
Сложности реализации серверного API	Небольшие отличия реализации API в разных версиях фреймворка. Сложность быстрого освоения особенностей клиент-серверного взаимодействия	Применение знакомых сред разработки серверной части проекта. Предварительное изучение реализации API. Реализация проекта без серверной части
Проблемы подключения библиотек	Реализация общей логики клиента и сервера в отдельной библиотеке. Библиотеки, подключаемые к Unity иногда вызывают проблемы при синхронизации через Git, особенно при разных версиях среды разработки	Не отделять логику в отдельную библиотеку, что понизит сложность реализации, восприятия и синхронизации через Git, но потребует дублирование кода
Сложность восприятия материала	Недостаточно структурированная и последовательная подача материала	Предварительная реализация проекта преподавателем для устранения подводных камней и сокращения процесса демонстрации разработки за счет использования готовых частей кода
Потеря общего контекста проекта	Разделение реализации проекта на несколько занятий, ввиду чего проект становится тяжело воспринимаемым, особенно при наличии пропусков части занятий	Рассмотрение проекта в заранее подготовленном виде в рамках одного занятия. Краткий пересказ ранее рассмотренного материала, в случае разделения на несколько пар. Поэтапная реализация – сначала в минимальной готовой версии, потом дополнения (библиотеки, API и др.)

The main difficulties encountered in the implementation of the project

Problem	Reason	Possible solution
Long installation of development tools	Implementation of the project on your own laptops	A preliminary task for installing software (did not help in 60% of cases). Work on classroom equipment with pre-installed software
Difficulties in implementing the server API	Small differences in the implementation of the API in different versions of the framework. The difficulty of quickly mastering the features of client-server interaction	The use of familiar development environments of the server part of the project. Preliminary study of the API implementation. Implementation of the project without the server part
Problems of connecting libraries	Implementation of common client and server logic in a separate library. Libraries connected to Unity sometimes cause problems when syncing via Git, especially with different versions of the development environment	Do not separate the logic into a separate library, which will reduce the complexity of implementation, perception and synchronization via Git, but will require code duplication
The complexity of the perception of the material	Insufficiently structured and consistent presentation of the material	Preliminary implementation of the project by the teacher to eliminate pitfalls and reduce the development demonstration process by using ready-made parts of the code. Consideration of the project in a pre-prepared form within the framework of one lesson
Loss of the overall project context	The division of the project implementation into several classes, which makes the project difficult to perceive, especially if there are absences of part of the classes	A brief retelling of the previously considered material, in case of separation into several pairs. Step-by-step implementation – first in a minimal ready-made version, then an addition (libraries, APIs, etc.)

Апробация предложенной методики проведена в группе студентов 3-го курса Ухтинского государственного технического университета в рамках нескольких занятий дисциплины «Мобильные разработки». По итогам первой

апробации с поставленной задачей успешно справились 6 из 10 принявших участие студентов. При выполнении проекта выявлен ряд сложностей (таблица), которые можно предупредить предварительной подготовкой и поэтапной выдачей задачи.

Через полгода после реализации проекта был проведен краткий опрос студентов о полезности проделанной работы, по итогам которого активные члены группы оценили полученный опыт на 8,5 из 10.

Заключение. Предложенный формат проведения занятий по программированию может быть полезен тем, что совокупно дает ряд компетенций, широко востребованных в современном мире разработки программного обеспечения, таких как:

- 1) совместная разработка;
- 2) разделение проекта на модули;
- 3) работа в системах контроля версий;
- 4) применение шаблонов проектирования.

Список литературы

- [1] *Маклецов С.В., Старишинова Т.А., Заринов Р.Н.* Интеграция учебной и профессиональной деятельности в подготовке студентов ИТ-направлений на основе сервиса Github // Управление устойчивым развитием. 2020. № 5 (30). С. 100–104.
- [2] *Маклецов С.В., Старишинова Т.А.* Электронное обучение – новое средство реализации интегративного и дифференцированного подходов (на примере бакалавров, обучающихся по направлению подготовки «Математика и компьютерные науки»): монография. Казань: Изд-во КНИТУ, 2015. 145 с.
- [3] *Жильев А.Н., Олейник А.И.* Актуальные проблемы подготовки и развития кадрового потенциала в ИТ-отрасли // Качество. Инновации. Образование. 2015. № 12 (127). С. 9–22.
- [4] *Старишинова Т.А., Маклецов С.В.* Применение средств электронного обучения для формирования информационной компетентности // Вестник Казанского технологического университета. 2012. Т. 15. № 17. С. 330–333.
- [5] *Аблаева Л.Н., Абдулаева У.Б.* Зарубежный и отечественный опыт использования Github в вузах // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2019. № 3 (25). С. 65–72.
- [6] *Андропов В.В.* Организация совместной практической деятельности на дистанционных ресурсах // Continuum. Математика. Информатика. Образование. 2016. № 2. С. 46–51.
- [7] *Протасевич Ю.А., Змеев О.А., Соколов Д.А.* Инструменты для организации взаимодействия преподавателей и студентов с использованием систем контроля версий // Информатика и образование. 2021. № 4 (323). С. 36–46. <http://doi.org/10.32517/0234-0453-2021-36-4-36-46>
- [8] *Еремеев М.А., Трубиенко О.В., Захарчук И.И.* Применение подхода воспроизводимых исследований в процессе дистанционного обучения // Russian Technological Journal. 2022. Т. 10. № 4. С. 86–92. <http://doi.org/10.32362/2500-316X-2022-10-4-86-92>
- [9] *Федотов А.М., Байдавлетов А.Т., Жижимов О.Л., Самбетбаева М.А., Федотова О.А.* Цифровой репозиторий в научно-образовательной информационной системе // Вестник НГУ. Серия: Информационные технологии. 2015. Т. 13. Вып. 3. С. 68–86.
- [10] *Алексеевский П.И.* Применение средств управления версиями для коллективной работы студентов над проектом компьютерной игры // Педагогическое образование в России. 2012. № 6. С. 51–54.

- [11] Чакон С., Штрауб Б. *Git для профессионального программиста*. СПб.: Питер, 2016. 496 с.
- [12] Козликіна Ю.А. Экстремальное программирование как инструмент по снижению рисков при разработке программного обеспечения // Информационные технологии в науке, управлении, социальной сфере и медицине: сборник научных трудов международной научной конференции. Томск: Национальный исследовательский Томский политехнический университет, 2014. С. 88–90.

References

- [1] Makletsov SV, Starshinova TA, Zaripov RN. Integration of educational and professional activities in the preparation of IT students based on the Github service. *Management of Sustainable Development*. 2020;(5):100–104. (In Russ.)
- [2] Makletsov SV, Starshinova TA. *E-learning is a new means of implementing integrative and differentiated approaches (on the example of bachelors studying in the field of mathematics and computer science)*. Kazan: KNITU Publ.; 2015. (In Russ.)
- [3] Zhilyaev AN, Oleinik AI. Actual problems of training and development of human resources in the IT industry. *Quality. Innovation. Education*. 2015;(12):9–22. (In Russ.)
- [4] Starshinova TA, Makletsov SV. The use of e-learning tools for the formation of information competence. *Bulletin of Kazan Technological University*. 2012;15(17):330–333. (In Russ.)
- [5] Ablava LN, Abdulaeva UB. Foreign and domestic experience of using Github in universities. *Information and Computer Technologies in Economics, Education and Social Sphere*. 2019;(3):65–72. (In Russ.)
- [6] Andropov VV. Organization of joint practical activity on remote resources. *Continuum. Mathematics. Computer Science. Education*. 2016;(2):46–51. (In Russ.)
- [7] Protasevich YuA, Zmeev OA, Sokolov DA. Tools for organizing interaction between teachers and students using version control systems. *Computer Science and Education*. 2021;(4):36–46. (In Russ.) <http://doi.org/10.32517/0234-0453-2021-36-4-36-46>
- [8] Ereemeev MA, Trubienko OV, Zakharchuk II. Application of the reproducible research approach in the process of distance learning. *Russian Technological Journal*. 2022;10(4):86–92. (In Russ.) <http://doi.org/10.32362/2500-316X-2022-10-4-86-92>
- [9] Fedotov AM, Baidavletov AT, Zhizhimov OL, Sambetbaeva MA, Fedotova OA. Digital repository in the scientific and educational information system. *Bulletin of the NSU. Series: Information Technology*. 2015;13(3):68–86. (In Russ.)
- [10] Alekseevsky PI. The use of version control tools for the collective work of students on a computer game project. *Pedagogical Education in Russia*. 2012;(6):51–54. (In Russ.)
- [11] Chacon S, Straub B. *Git for a professional programmer*. St. Petersburg: Peter Publ.; 2016. (In Russ.)
- [12] Kozlikina YuA. Extreme programming as a risk reduction tool in software development. *Information Technologies in Science, Management, Social Sphere and Medicine: Collection of Scientific Papers of the International Scientific Conference*. Tomsk: National Research Tomsk Polytechnic University; 2014. p. 88–90. (In Russ.)

Сведения об авторах:

Рочев Константин Васильевич, кандидат экономических наук, доцент кафедры вычислительной техники, информационных систем и технологий, факультет экономики, управления и информационных технологий, Ухтинский государственный технический университет, Российская Федерация, 169300, Ухта, ул. Первомайская, д. 13. ORCID: 0000-0002-2720-3209, Scopus Author ID: 55611356300, ResearcherID L-9473-2013, eLIBRARY SPIN-код: 3614-8511, AuthorID: 574874. E-mail: k@rochev.ru

Семьяшкіна Анастасія Вікторівна, старший преподаватель, кафедра вычислительной техники, информационных систем и технологий, факультет экономики, управления и информационных технологий, Ухтинский государственный технический университет, Российская Федерация, 169300, Ухта, ул. Первомайская, д. 13. ORCID: 0000-0002-6781-0396. E-mail: asemyashkina@ugtu.net

Bio notes:

Konstantin V. Rochev, Candidate of Economics, Associate Professor, Department of Computer Engineering, Information Systems and Technologies, Faculty of Economics, Management and Information Technologies, Ukhta State Technical University, 13 Pervomayskaya St, Ukhta, 169300, Russian Federation. ORCID: 0000-0002-2720-3209, Scopus Author ID: 55611356300, ResearcherID L-9473-2013, eLIBRARY SPIN-code: 3614-8511, AuthorID: 574874. E-mail: k@rochev.ru

Anastasia V. Semyashkina, senior lecturer, Department of Computer Engineering, Information Systems and Technologies, Faculty of Economics, Management and Information Technologies, Ukhta State Technical University, 13 Pervomayskaya St, Ukhta, 169300, Russian Federation. ORCID: 0000-0002-6781-0396, eLIBRARY SPIN-code: 7492-7509. E-mail: asemyashkina@ugtu.net