

DOI: 10.32517/0234-0453-2021-36-6-60-66

ТЕХНОЛОГИЯ ДИФФЕРЕНЦИРОВАННОГО ОБУЧЕНИЯ ОСНОВАМ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ НА ПЕРВОМ КУРСЕ ВУЗА НА БАЗЕ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ DL.GSU.BY

М. С. Долинский¹ ✉, М. А. Долинская¹

¹ Гомельский государственный университет имени Франциска Скорины, г. Гомель, Беларусь

✉ dolinsky@gsu.by

Аннотация

В статье описывается технология обучения первокурсников текстовому программированию на базе сайта DL.GSU.BY. К основным достоинствам технологии можно отнести: «нулевой порог входа», адаптирующееся к студенту обучение, многолетний опыт практического применения, результативность, масштабируемость. В статье последовательно рассмотрены следующие вопросы: идеализированная целевая установка; использование студентами языка программирования по выбору из множества современных языков программирования при выполнении практических заданий по дисциплине; ясная проверка достижения цели; смешанное обучение; эффективная персонализация учебного процесса; нестандартные организационные и технические решения; результативность обучения. Идеализированная целевая установка включает в себя необходимость научить студентов следующему: алгоритмическое переформулирование условия задачи; владение набором базовых конструкций языка, а также основными встроенными процедурами и функциями; умение использовать базовые алгоритмы на одномерных и двумерных массивах, множествах точек плоскости, строках, очереди; способность разрабатывать и отлаживать новые алгоритмы. Эффективная персонализация учебного процесса обеспечивается с помощью таких приемов: на каждом занятии студенту предлагается выбор видов деятельности, соответствующих текущему уровню его подготовки; обеспечивается автоматическая проверка решений с сервисом «Уступка тестов»; используется система автоматического дифференцированного обучения.

Ключевые слова: дистанционное обучение программированию, сайт DL.GSU.BY, Гомельский государственный университет имени Франциска Скорины.

Для цитирования:

Долинский М. С., Долинская М. А. Технология дифференцированного обучения основам алгоритмизации и программирования на первом курсе вуза на базе системы дистанционного обучения DL.GSU.BY. *Информатика и образование*. 2021;36(6):60–66. DOI: 10.32517/0234-0453-2021-36-6-60-66

TECHNOLOGY OF DIFFERENTIATED TRAINING BASES OF ALGORITHMIZATION AND PROGRAMMING ON THE FIRST COURSE OF THE UNIVERSITY ON THE BASIS OF THE DL.GSU.BY SYSTEM OF DISTANCE LEARNING

M. S. Dolinsky¹ ✉, M. A. Dolinskaya¹

¹ Francisk Skorina Gomel State University, Gomel, Belarus

✉ dolinsky@gsu.by

Abstract

The article describes the technology of teaching text-based programming on the basis of the DL.GSU.BY website. The main advantages of the technology include: “zero entry threshold”, training adapted to the student, many years of practical experience, efficiency, scalability. The following issues are consistently considered in the article: idealized goal setting; the use by students of a programming language of their choice from a variety of modern programming languages when performing practical tasks in the discipline; a clear verification of goal achievement; blended learning; effective personalization of the educational process; non-standard organizational and technical decisions; effectiveness of training. The idealized goal setting includes the need to teach students the following: algorithmic reformulation of the problem statement; knowledge of a set of basic language constructs, as well as basic built-in procedures and functions; the ability to use basic algorithms on one-dimensional and two-dimensional arrays, sets of plane points, lines, queues; the ability to develop and debug new algorithms. Effective personalization of the educational process is provided with

the help of such techniques: at each lesson, the student is offered a choice of activities that correspond to current level of his training; automatic verification of solutions is provided with the test assignment service; the system of automatic differentiated learning is used.

Keywords: distance learning for programming, site DL.GSU.BY, Francisk Skorina Gomel State University.

For citation:

Dolinsky M. S., Dolinskaya M. A. Technology of differentiated training bases of algorithmization and programming on the first course of the university on the basis of the DL.GSU.BY system of distance learning. *Informatics and Education*. 2021;36(6):60–66. DOI: 10.32517/0234-0453-2021-36-6-60-66 (In Russian.)

1. Введение

Одной из глобальных тенденций изменения высшего образования является переход к *смешанному обучению*, когда существенная часть учебного процесса переносится из учебной аудитории в Сеть. С помощью интернета студенты получают задания и необходимую теоретическую информацию в самой разнообразной форме (текстовой, графической, аудио, видео), выполняют задания и отправляют их на проверку тьюторам или системам автоматической проверки решений.

В последние годы к теоретическим и практическим вопросам смешанного обучения приковано внимание исследователей многих стран. В литературе по этой теме отражены четыре основных направления:

- обзоры;
- предложения по переходу к смешанному обучению;
- оценка (в том числе сравнительная) систем, используемых при смешанном обучении;
- изложение опыта работы.

Среди многочисленных источников хотелось бы отметить несколько зарубежных работ, в которых отражены вопросы внедрения смешанного обучения *в вузах*, в частности *при обучении компьютерным наукам*.

К первому направлению — обзорные материалы — относятся, например, работы [1] и [2]. Работа [1] посвящена обзору стратегий и средств внедрения смешанного обучения. Работа [2] представляет собой обзор и анализ компьютерного образования в японских университетах. Авторы работы выделили следующие виды компьютерного образования:

- на факультетах, специализирующихся в компьютерных дисциплинах;
- на факультетах, не специализирующихся в компьютерных дисциплинах, но имеющих компьютерные дисциплины как важную часть обучения;
- общее компьютерное образование для студентов всех факультетов, обычно на первом-втором годах обучения;
- компьютерное образование для студентов, желающих получить лицензию на преподавание компьютерных дисциплин в школах.

Ко второму направлению — подходы к переходу к смешанному обучению — относятся работы [3, 4]. В работе [3] постулируется необходимость цифровой трансформации образования и подробно поясняется, что понимается под такой трансформацией. В работе [4] описаны направления развития смешанного обучения в вузах Азии.

Третье направление — разнообразные языки и системы, используемые при обучении. Среди предлагаемых систем упоминаются GeoGebra [5], Google Classroom [6], Moodle [7, 8].

Четвертое направление объединяет работы, представляющие реальный опыт обучения с помощью компьютерных технологий в вузах: программированию [9], компьютерному мышлению [10], химии [11], основам медицинских знаний [12], наукам о Земле [13], анатомии [14], веб-дизайну [15], нейроанатомии [16], английскому языку [17], новым образовательным технологиям [18].

Данная статья описывает опыт авторской реализации смешанного образования в Гомельском государственном университете имени Франциска Скорины при обучении программированию студентов первого курса в первом семестре.

Авторы много лет активно занимаются подготовкой школьников к олимпиадам по информатике и программированию [19]. Опыт этой работы лег в основу занятий со студентами [20].

Однако необходимо заметить, что работа *со школьниками* ведется в рамках кружков по интересам, и поэтому не требуется тратить существенные усилия на обучение тех, кто не хочет, — такие ребята просто перестают ходить на кружок. Принципиально другая ситуация *со студентами*. Работать необходимо со всеми. Особенно сложно стало примерно с 2005 года, когда существенно увеличился набор на платное обучение (до 50 % от набора), в результате чего резко упал средний уровень подготовки студентов первого курса, причем не только специальной подготовки (по информатике и программированию), но и общей (например, по математике, русскому языку и устной речи). Когда на совете факультета преподаватели начали жаловаться на такую ситуацию декану, он ответил: «Учите этих, других не будет». Несколько сглаживает такую ситуацию то, что у студентов есть экзамен по дисциплине.

Таким образом, *методология обучения должна развиваться по трем направлениям: процесс обучения должен быть **посильным, интересным, заставляющим*** (для тех, кому посильности и интересности недостаточно).

2. Идеализированная целевая установка

В рамках обучения в первом семестре студентов первого курса дисциплине «Основы алгоритмизации и программирования» **авторы ставят перед собой цели научить студентов следующему:**

1. Алгоритмическое переформулирование условия задачи.

Оригинальное условие задачи, как правило, содержит некоторую легенду или «пользовательское» объяснение работы, которую должна выполнить программа. Студент должен переформулировать условие задачи, сохранив ее суть, в терминах обработки чисел, символов, строк, точек (заданных координатами X и Y на плоскости) или их одномерных/двумерных массивов.

2. Владение набором базовых конструкций языка, к которым мы относим объявления переменных и массивов, операторы присваивания, условия, циклы FOR и WHILE, а также встроенные процедуры и функции:

- обработки чисел: abs, sqr, sqrt, odd;
- обработки строк: length, delete, insert, copy, pos;
- обработки символов: chr, ord;
- преобразования типов: val, str;
- ввода-вывода: readln, writeln, assign, reset, rewrite, close.

3. Владение набором базовых алгоритмов, к которым мы относим:

- *базовые алгоритмы обработки одномерных массивов*: суммирование; подсчет элементов, обладающих заданными свойствами; максимальный/минимальный элемент; поиск элемента, обладающего указанными свойствами; сортировка: обменом, пузырьком, подсчетом; сортировка с номерами;
- *перенос базовых алгоритмов на двумерные массивы и их компоненты* (строки, столбцы, диагонали);
- *алгоритмы поиска расстояний на плоскости*: между двумя точками; от одной точки до нескольких; между соседними точками множества точек; от каждой точки до каждой в одном множестве точек; между всеми парами точек двух множеств;
- *базовые строковые алгоритмы*: разбиение предложения на слова; формирование всех различных символов строки; линейный подсчет количества символов в строке, переворачивание строки;
- *очередь*: задача о коне (минимальном количестве ходов коня из одной клетки шахматной доски в другую) и задача о кусках (количество кусков, на которое распадется листок бумаги после вырезания некоторых клеток).

4. Умение применять при решении задач базовые алгоритмы, их модификации, композиции и суперпозиции.

5. Способность разрабатывать новые алгоритмы и программы, выполнять ручную прокрутку алгоритмов и отладку программ в среде разработки с использованием команд отладчика: поставить переменные и массивы в окно просмотра; удобно расположить на экране окна исходного текста и просмотра данных; выполнить строку; выполнить до курсора; войти в процедуру; выполнить до конца.

3. Замечание о мультиязычности при изучении дисциплины

Студент имеет право сдавать решения предлагаемых задач на любом из языков программирования, установленных в систему проверки, в том числе: Pascal (Turbo, Free, ABC), C++ (множество версий), Java, Python, C#. При необходимости в систему устанавливаются новые версии языков и новые языки.

Обучение для начинающих ведется на Pascal в связи с наличием мощной системы автоматического обучения, разрабатываемой авторами данной статьи, эксплуатируемой и развивающейся более 20 лет. Однако имеются конспекты лекционных материалов для других перечисленных выше языков программирования, и эти материалы тоже постоянно совершенствуются.

4. Ясная проверка достижения цели

Экзаменационная оценка за семестр студенту фактически определяется количеством задач, которые он решает за академическую пару (1,5 часа) — от 0 до 10.

На экзамене предлагаются задачи на следующие темы:

- 1) Введение в программирование.
- 2) Одномерный массив.
- 3) Двумерный массив.
- 4) Геометрия.
- 5) Строки.
- 6) Одномерный массив.
- 7) Двумерный массив.
- 8) Геометрия.
- 9) Строки.
- 10) Очередь.

Как можно заметить, темы задач 2–5 в точности совпадают с темами задач 6–9. Отличие — в уровне сложности.

Задачи в темах 2–5 подобраны таким образом, чтобы с ними справился любой студент, который более или менее добросовестно занимался в течение семестра, даже при нулевом уровне предварительной готовности.

Задачи 6–9 фактически проверяют качество сформированности у студента навыков разработки алгоритмов и отладки программ.

Многолетняя практика и адаптация использования подхода показала свою эффективность в оценивании (показывая адекватный умениям студентов разброс от 6 до 9).

Способность решить задачу 10 показывает готовность студента к изучению более сложных алгоритмов.

5. Смешанное обучение

Учебный процесс ведется с помощью соответствующего курса в **системе дистанционного обучения DL.GSU.BY**, который с 2005 года имеет историческое

название «ЭВМ, программирование, методы алгоритмизации — 20nn».

Курс обеспечивает:

- доступ к теории;
- выдачу заданий;
- автоматическую проверку решений;
- дифференцированное обучение, реализующее адаптирующуюся к студенту индивидуальную образовательную траекторию;
- мотивирующее отображение результатов обучения в разнообразных разрезах.

Все это гарантирует возможность эффективной самостоятельной работы студентов в учебных аудиториях и вне их.

Задачи преподавателя — доходчиво и лаконично излагать новую теорию на лекционных занятиях, организовывать и поддерживать рабочую атмосферу на учебных занятиях.

Особое внимание требуется уделить развитию навыков **самостоятельной работы первокурсников**. Очень многие из них на первых занятиях пытаются беспрерывно и без реальной надобности требовать помощи преподавателя, вместо того чтобы подумать и решить проблему самостоятельно или с помощью обучающей системы.

6. Эффективная персонализация учебного процесса

Наша цель при организации учебных занятий формулируется следующим образом: на каждом занятии каждую минуту работает каждый студент:

- желательно потому, что ему интересно;
- как минимум потому, что его заставляет система и/или учитель и/или «общественное мнение».

Как и за счет чего эта цель в значительной степени достигается в реальной практике?

1. Возможность выбора видов деятельности для каждого студента на каждом занятии соответственно текущему уровню его подготовки.

На лекциях (после прочтения новой теории, как правило, в течение 20–30 минут) предлагаются задания *разных* уровней сложности по теме лекции. Кроме того, на лекциях студент может решать задачи как на собственном ноутбуке в одиночку, так и в команде из двух человек, близких по текущему уровню знаний (этот уровень определяется текущей оценкой каждого).

На практике и при самостоятельной работе дома у студента есть выбор как темы (из ранее изученных) для закрепления, так и уровня сложности решаемых задач.

2. Автоматическая проверка решений (программ, написанных студентами на лекции и практике), выполняемая в режиме реального времени в течение нескольких секунд.

3. Сервис «Уступка тестов», который позволяет студенту узнать тест (входные данные и правильный ответ для них), на котором его программа работает неправильно.

4. Система автоматического дифференцированного обучения по каждой из изучаемых тем:

- 1) Введение в программирование.
- 2) Одномерный массив.
- 3) Двумерный массив.
- 4) Геометрия.
- 5) Строки.
- 6) Очередь.

Первые пять тем имеют два уровня сложности — базовый и упрощенный. Так что каждый студент может обучаться самостоятельно как на занятиях, так и дома.

5. Интегрированные в тематическое обучение пакеты заданий на развитие мышления, которые студент может пропускать при невозможности или нежелании выполнять такие задания.

6. Доступность теории, которая была изложена на лекционных и практических занятиях.

7. Индивидуальные задания. В соответствующую папку собираются задачи как по темам, так и по олимпиадам, для которых в курсе нет ни решений, ни объяснений. Для каждой задачи отображается уровень ее сложности, описываемый двумя числами: 1) сколько человек сдали эту задачу и 2) среднее количество дней на одну сдачу (т. е. количество дней с момента установки задачи до текущего дня, деленное на количество человек, сдавших эту задачу). Студент может выбрать себе любую задачу, решить ее (на любом языке программирования), а затем описать в форуме свое решение и выложить читабельный самодокументированный исходный текст решения. После проверки преподавателя такая задача переносится в другое место курса — «Индивидуальные задания с описанием решений в форуме». Сама работа высоко оценивается в текущей ведомости успеваемости. Кроме того, ссылки на все такие описания решений собираются по темам — как безотносительно языка программирования, так и отдельно по каждому из языков программирования. Таким образом, студенты получают *возможность дополнительного саморазвивающегося вида обучения*: решать задачи при возможности, в конце концов, прочитать решение на форуме.

8. Соревновательные элементы, внедренные в учебный процесс:

- После каждого учебного занятия в специальную тему «Кто как работает» на форуме курса заносится автоматически сформированный результат работы, выполненной на этом занятии каждым студентом: в какой теме работал, сколько заданий выполнил, где находится в теме. При этом выше в таблице находятся студенты, которые прошли в обучении дальше.
- Еженедельно проводится контрольная работа, которая содержит порядка 30 задач на разработку программ различного уровня сложности — от самых простых до самых сложных (задачи как на пройденные, так и на еще не изученные на лекциях темы). В конце контрольной работы таблица результатов (кто

сколько и каких задач сдал, сделал попыток) предъявляется всем студентам.

- Ежедневно на лекции после прочтения теории предлагается к решению порядка 30 заданий различной сложности по изученной на текущей лекции теме. В конце занятия таблица результатов предъявляется студентам.
- Постоянно в течение семестра студентам доступна текущая ведомость оценок, которая интегрированно оценивает результаты обучающихся в семестре, включая работу на лекции, практике, дома, показывая также текущую экзаменационную оценку каждого.

7. Нестандартные организационные и технические решения

1. Автоматизированный учет пропусков и их отработки.

Одна из проблем современного обучения в вузах — пропуск занятий студентами. Авторы понимают, что для этого могут быть как реальные объективные и субъективные причины, так и «исследование студентами границ дозволенного». Поэтому пропуски введены отдельной колонкой в ведомость успеваемости и отрицательно влияют на оценку работы в семестре. Практически после семи пропущенных занятий оценка становится отрицательной — у тех, кто лучше всех работает на лекциях и практике. А у тех, кто работает хуже, оценка может стать отрицательной уже после пяти пропусков. Поэтому студентам предоставлена возможность *отрабатывать пропуски*. В любое удобное для себя время студент может зайти на сайт, отработать полтора часа (от первой до последней отсылки по протоколу работы), выбрать дату пропуска, который отрабатывался, и система сформирует сообщение в специальной теме форума «Отработка пропусков — 20пп» такого вида: такой-то студент из такой-то группы работал такого-то числа с такого-то по такое-то время, решил столько-то задач, общее время работы такое-то. В это сообщение также включается прямая ссылка на протокол этой отработки.

Преподаватель со своего аккаунта может кликнуть по этой ссылке, проверить, была ли это действительная работа или ее симуляция, и принять отработку (т. е. пропуска не станет) или отказать в отработке (пропуск останется).

В целях обеспечения непрерывной работы по дисциплине в семестре отработка без преподавателя (дома) «пара за пару» разрешается только в течение недели после пропуска. Иначе отработка по желанию студента совершается или при преподавателе «пара за пару», или дома в увеличенном размере (на пару за каждую лишнюю неделю между пропуском и отработкой).

2. Борьба с недисциплинированностью студентов во время занятий.

К проявлениям такой недисциплинированности можно отнести опоздания на занятия, «удлиненные»

перерывы, отсутствие тетради и ручки на занятиях, отвлечение на мобильные телефоны во время занятий, использование ноутбука не в целях занятий и т. д. Для большинства студентов достаточно объяснения правил, в крайнем случае, — однократных замечаний-напоминаний. Однако есть и «активные исследователи границ дозволенного», не реагирующие на объяснения для всех и даже на персональные замечания. Таким студентам выставляется пропуск текущего занятия с правом студента «сбросить» его, если следующую такую же пару (ровно через неделю) студент проведет без замечаний и напомнит преподавателю о желании сбросить пропуск, выставленный за замечание.

3. Использование форума дисциплины для поддержки самостоятельного обучения.

Авторы стараются максимально обеспечить студентам возможность самостоятельной работы как на занятиях, так и вне их.

В частности, на форуме есть *тема «Помогите, пожалуйста, — 20пп»*, в которой студент может задать интересующий его вопрос (в том числе по решаемым задачам). Если оперативно никто из студентов не отвечает, тогда это делает преподаватель. Очень важно, что в начале этой темы в систематическом порядке происходит накопление ссылок на вопросы и ответы к ним. Таким образом, зачастую студенту может и не потребоваться задавать свой вопрос, если такой вопрос уже задавался и ответ на него зафиксирован.

Кроме того, по инициативе одного из студентов (2009 года обучения) была создана *тема «Учим друг друга, учимся друг у друга — 20пп»*, в которой можно написать, как решил трудную/индивидуальную задачу, или почитать, как это удалось сделать другому. Оглавление темы вручную постоянно пополняется преподавателем.

Тема «Мнение студентов — 20пп» предназначена для получения обратной связи от студентов как в процессе обучения, так и после его завершения.

Тема «ЭВМ, программирование, методы алгоритмизации — 20пп» в начале содержит все организационные сообщения для новых студентов, а также ссылки на материалы по всем изучаемым темам. Последнее сообщение в теме содержит ссылки на материалы по последней прочитанной теме.

Наконец, *тема «Бонусы — 20пп»* содержит оценки каждого студента в «бонусах» по последнему лекционному занятию. Кроме того, там появляются по мере необходимости сообщения о бонусировании той или иной познавательной активности студентов на лекции, практике или в самостоятельной работе.

4. Многократные повторные пересдачи экзамена во время семестра.

Если студент регулярно занимается, то фактически его оценка на экзамене определяется оценкой по контрольному срезу. Примерно с середины семестра студенту разрешается заявляться на контрольный срез и писать его раз в неделю. При этом оценка фиксируется только в сторону увеличения.

8. Результативность обучения

На первом практическом занятии проводится входной контрольный срез, который фиксирует начальную подготовку студентов до поступления в вуз, с небольшой поправкой на то, что до первого практического занятия бывают одна или даже две лекционных недели и многие студенты, мотивированные проведенными лекционными занятиями и возможностью работать дома, неплохо продвигаются в знаниях и за этот стартовый отрезок времени. Тем не менее в среднем оценка каждого студента в конце семестра повышается на значение от 4 до 9.

Авторам представляются важными и такие субъективные показатели результативности обучения:

- большинство студентов начинают заниматься сразу по приходе на занятие, не дожидаясь звонка о начале пары, и «не замечают» перерыв между часами пары, продолжая работать;
- во время занятий стоит практически идеальная тишина, все активно и увлеченно работают;
- большинство студентов приветливо здороваются с преподавателем и после завершения семестра и сдачи экзаменов.

9. Заключение

В данной работе представлены многолетний опыт и методология обучения основам алгоритмизации и программирования в первом семестре первого курса студентов математического факультета Гомельского государственного университета имени Франциска Скорины. Они могут быть использованы любым вузом и преподавателем как непосредственно — преподаватель может подписать своих студентов на соответствующий курс системы дистанционного обучения DL.GSU.BY, так и позаимствовав методологию обучения, но используя другие технические и программные решения.

К важным компонентам методологии относятся:

- использование лектором персонального компьютера и проектора;
- использование ноутбуков студентами на лекции;
- прямое закрепление теоретического материала практикой;
- оперативная проверка программ — решений задач;
- таблица результатов в реальном времени на лекциях и практиках;
- решение задач в командах;
- «оценочное» поощрение активной работы на лекциях и практиках;
- адаптивное персонализированное обучение;
- автоматизированная система учета и отработки пропусков;
- автоматизированное объективное оценивание знаний и умений студентов.

Список источников / References

1. Aznam N., Perdana R., Jumadi J., Nurcahyo H., Wiyatmo Y. The implementation of Blended Learning and Peer Tutor Strategies in Pandemic Era: A systematic review. *Advances in Social Science, Education and Humanities Research*. 2021;(541):906–914. DOI: 10.2991/assehr.k.210326.130
2. Kakeshita T., Ohtsuki M. Survey and analysis of computing education at Japanese universities: Non-IT departments and courses. *Olympiads in Informatics*. 2019;(13):57–79. Available at: https://ioinformatics.org/journal/v13_2019_57_80.pdf
3. Jones K. A., Sharma R. S. Higher education 4.0. The digital transformation of classroom lectures to Blended Learning. Singapore: Springer; 2021. 279 p. DOI: 10.1007/978-981-33-6683-1
4. Zaugg H., Graham C. R., Lim C. P., Wang T. Current and future directions of Blended Learning and teaching in Asia. *Blended Learning for Inclusive and Quality Higher Education in Asia*. Singapore: Springer; 2021. P. 301–327. DOI: 10.1007/978-981-33-4106-7
5. Stahl G. Redesigning mathematical curriculum for Blended Learning. *Education Sciences*. 2021;11(4):165–177. DOI: 10.3390/educsci11040165
6. Astarilla L., Warman D. The effect of Google classroom in Blended Learning on university students' English ability. *J-SHMIC: Journal of English for Academic*. 2021;8(1):12–23. DOI: 10.25299/jshmic.2021.vol8(1).6216
7. Antwi-Boampong A. Blended Learning adoption in higher education: Presenting the lived experiences of students in a public university from a developing country. *The Turkish Online Journal of Educational Technology*. 2021;20(2):14–22. Available at: <http://www.tojjet.net/articles/v20i2/2022.pdf>
8. Oktaria S. D., Sasongko R. N., Kristiawan M. Development of Blended Learning designs using Moodle to support academics of the curriculum in university of Bengkulu. *Jurnal Studi Guru dan Pembelajaran*. 2021;4(1):118–126.
9. Lodi M., Malchiodi D., Monga M., Morpurgo A., Spieker B. Constructionist attempts at supporting the learning of computer programming: A survey. *Olympiads in Informatics*. 2019;(13):99–121. Available at: https://ioinformatics.org/journal/v13_2019_99_122.pdf
10. Weigend M., Vaníček J., Pluhár Z., Pesek I. Computational thinking education through creative unplugged activities. *Olympiads in Informatics*. 2019;(13):171–192. Available at: https://ioinformatics.org/journal/v13_2019_171_192.pdf
11. Chamberlain S., Elford D., Lancaster S. J., Silve F. Tailored Blended Learning for foundation year chemistry students. 2021;75(1-2):18–26. DOI: 10.2533/chimia.2021.18
12. Lovey T., O'Keeffe P., Petignat I. Basic medical training for refugees via collaborative Blended Learning: Quasi-experimental design. *Journal of Medical Internet Research*. 2021;23(3). DOI: 10.2196/22345
13. Bond C. E., Cawood A. J. A role for virtual outcrop models in Blended Learning — improved 3D thinking and positive perceptions of learning. *Geoscience Communication*. 2021;4(2):233–244. DOI: 10.5194/gc-4-233-2021
14. Sarkar S., Sharma S., Raheja S. Implementation of Blended Learning approach for improving anatomy lectures of Phase I MBBS students — learner satisfaction survey. *Advances in Medical Education and Practice*. 2021;(12):413–420. DOI: 10.2147/AMEP.S301634
15. Sulistiyarini D., Sabirin F., Ramadhani D. Effect of project-based learning through Blended Learning on website design skills. *Journal of Educational Science and Technology*. 2021;7(1):58–66. Available at: <http://103.76.50.195/JEST/article/view/17789>
16. Border S., Woodward C., Kurn O., Birchall C., Laurayne H., Anbu D., Taylor C., Hall S. Working in creative partnership with students to co-produce neuroanatomy

e-Learning resources in a new era of Blended Learning. *Anatomical Sciences Education*. 2021;14(4):417–425. DOI: 10.1002/ase.2090

17. Jerry M., Yunu M. Blended Learning in rural primary ESL classroom: Do or Don't. *International Journal of Learning, Teaching and Educational Research*. 2021;20(2):152–173. DOI: 10.26803/ijlter.20.2.9

18. Al Musawi A. S., Ammar M. E. The effect of different blending levels of traditional and e-Learning delivery on aca-

demic achievement and students' attitudes towards Blended Learning at Sultan Qaboos University. *The Turkish Online Journal of Educational Technology*. 2021;20(2):127–139. Available at: <http://www.tojet.net/articles/v20i2/20213.pdf>

19. Dolinsky M. Gomel training school for olympiads in informatics. *Olympiads in Informatics*. 2016;(10):237–247.

20. Dolinsky M. An approach to teach introductory-level computer programming. *Olympiads in Informatics*. 2013;(7):14–22.

Информация об авторах

Долинский Михаил Семёнович, канд. тех. наук, доцент, доцент кафедры математических проблем управления и информатики, факультет математики и технологии программирования, Гомельский государственный университет имени Франциска Скорины, г. Гомель, Беларусь; *ORCID*: <http://orcid.org/0000-0002-3057-4051>; *e-mail*: dolinsky@gsu.by

Долинская Мария Александровна, ассистент кафедры математических проблем управления и информатики, факультет математики и технологии программирования, Гомельский государственный университет имени Франциска Скорины, г. Гомель, Беларусь; *ORCID*: <http://orcid.org/0000-0003-1325-5822>; *e-mail*: mkugejko@gsu.by

Information about the authors

Mikhail S. Dolinsky, Candidate of Sciences (Engineering), Docent, Associate Professor at the Department of Mathematical Problems of Management and Informatics, Faculty of Mathematics and Technologies of Programming, Francisk Skorina Gomel State University, Gomel, Belarus; *ORCID*: <http://orcid.org/0000-0002-3057-4051>; *e-mail*: dolinsky@gsu.by

Maria A. Dolinskaya, Assistant Professor at the Department of Mathematical Problems of Management and Informatics, Faculty of Mathematics and Technologies of Programming, Francisk Skorina Gomel State University, Gomel, Belarus; *ORCID*: <http://orcid.org/0000-0003-1325-5822>; *e-mail*: mkugejko@gsu.by

Поступила в редакцию / Received: 28.05.2021.

Принята к печати / Accepted: 10.08.2021.