

Мир науки. Педагогика и психология / World of Science. Pedagogy and psychology <https://mir-nauki.com>

2021, №6, Том 9 / 2021, No 6, Vol 9 <https://mir-nauki.com/issue-6-2021.html>

URL статьи: <https://mir-nauki.com/PDF/27PDMN621.pdf>

**Ссылка для цитирования этой статьи:**

Пирогов, В. Ю. О возможностях дистанционного преподавания программирования для студентов технических специальностей / В. Ю. Пирогов // Мир науки. Педагогика и психология. — 2021. — Т. 9. — № 6. — URL: <https://mir-nauki.com/PDF/27PDMN621.pdf>

**For citation:**

Pirogov V.Ju. On the possibilities of distance teaching of programming for students of technical specialties. *World of Science. Pedagogy and psychology*, 9(6): 27PDMN621. Available at: <https://mir-nauki.com/PDF/27PDMN621.pdf>. (In Russ., abstract in Eng.).

**Пирогов Владислав Юрьевич**

ФГБОУ ВО «Шадринский государственный педагогический университет», Шадринск Россия  
Профессор кафедры «Программирования и автоматизации бизнес процессов»  
Кандидат физико-математических наук, доцент  
E-mail: Vladislav-133@yandex.ru

## **О возможностях дистанционного преподавания программирования для студентов технических специальностей**

**Аннотация.** Переход на дистанционное обучения в связи с распространением коронавирусной пандемии высветил некоторые подходы в преподавании, которые могут обогатить и традиционные подходы образования. В статье рассматриваются некоторые частные аспекты, связанные с преподаванием дисциплин в учреждениях высшего образования для студентов технических специальностей, изучающих дисциплины, связанные с изучением программирования. Автор рассматривает два типа занятий: лекционные и практические. В статье показывается, что возможности онлайн-лекций с точки зрения подачи информации обогащают учебный процесс: возможность записи занятия и последующем ее использовании для подготовки студентов, использование дополнительных технических средств, чему способствует тот факт, что преподаватель сам выбирает аппаратные средства и программное обеспечение. Кроме того автор указывает на простоту показа процесса программирования, проектирования, отладки и т. д. Автор указывает на то, что практические занятия в онлайн-режиме можно сделать эффективными, если использовать такое средство, как автоматическую проверку заданий. Указывается, что в настоящее время имеется целый ряд инструментов, которые позволяют организовать подобный вид учебной работы. В статье отмечается, что эта возможность должна обязательно быть перенесена на занятия в обычном формате, поскольку это может значительно повысить эффективность выработки навыков практического программирования у студентов. В статье также рассматривается вопрос о работе с программными проектами. Автор указывает, что онлайн-режим здесь также не менее эффективен, чем аудиторный. Определенные элементы, несомненно, должны быть перенесены в обычный режим работы.

**Ключевые слова:** дистанционное обучение; программирование; онлайн-лекции; программные проекты; обучение программированию; программное обеспечение; отладка

## Введение

Пришедшая пандемия коронавируса изменила многие парадигмы окружающего мира, коснувшись самых разных сторон жизни. Серьезные изменения произошли и в системе высшего и среднего образования. Переход на дистанционную форму обучения происходил стремительно и не мог не принести определенные потери в качества образования. Вместе с тем преподавательское сообщество и вся система образования довольно гибко среагировала на вынужденные обстоятельства и спустя уже более полутора лет с начала вспышки заболевания демонстрирует, на наш взгляд, устойчивость к меняющимся обстоятельствам.

Автору также пришлось вместе со всей системой высшего образования участвовать в этом вынужденном эксперименте<sup>1</sup>, разрабатывать новые подходы, переосмысливать старые методики и менять свое собственное отношение к системе взаимодействия между преподавателем и студентами.

В рамках данной статьи мы рассмотрим особенности перехода с традиционного на дистанционное обучение [1–5] по дисциплинам, связанным с таким важным направлением подготовки, как программирование. Мы особо выделяем область нашего рассмотрения, поскольку другие предметные разделы требуют отдельного анализа.

Перестройка подходов при переходе на дистанционную форму обучения высветила и некоторые аспекты, которые уже давно назревали в рамках традиционного образования [6–8]. Эти подходы, несомненно, могут обогатить очную форму обучения [9]. Более того, при подготовке специалистов в области программного обеспечения [9–12] они, скорее всего, станут обязательными. Мы имеем в виду прежде всего систему проверки и оценки практических навыков в области программирования и разработки программных продуктов<sup>2</sup>.

Далее мы будем говорить об онлайн-занятиях, онлайн-обучении, имея в виду, что данный тип взаимодействия, обучающего и обучаемого является частным случаем дистанционного подхода в обучении.

## Особенность онлайн-лекций

Онлайн-лекции, которые приходится читать при дистанционном обучении, потребовали внесения определенной корректировки в организации процесса. Укажем на некоторые особенности, имея в виду ту область преподавания, которая была обозначена во введении [13].

1. Вопрос присутствия студентов на занятиях, конечно, относится не столько к вопросу методики изложения материала, сколько к техническим моментам организации занятия: наличие у присутствующих веб-камер, периодическая переключка и т. п. Однако возможность записи лекции, о чем будет сказано ниже, до определенной степени смягчает данную проблемы.

2. На наш взгляд более важной чертой онлайн-лекции является возможности, которые имеются у преподавателя для представления лекционного материала. Поскольку для представления материала используются компьютер преподавателя, то для него открываются самые разные возможности: установка дополнительных камер, графических планшетов, показ работы на компьютере по принципу «делай как я». Преподаватель, таким образом, может расширить возможности изложения материала на основе своих собственных представлений о

---

<sup>1</sup> В 2020 году были внесены ряд поправок в федеральный закон Об образовании в Российской Федерации.

<sup>2</sup> Мы разделяем обучение программированию и процессу разработки программных продуктов. В первом случае упор делается на владение языком программирования и алгоритмические навыки. Во втором случае речь идет о компетенциях, связанных с получением готового программного продукта и всех этапах его разработки.

предмете, а не на основе представленных ему некоторых унифицированных технических средств. Мы оставляем в стороне вопрос о приобретении преподавателем тех или иных технических средств и считаем, что он может быть решен по договоренности с учебным заведением.

3. При обучении программированию, на наш взгляд, важным приемом является живой показ процесса программирования преподавателем [14]. С демонстрацией ошибок, их исправлениями, показам процесса отладки, элементов тестирования, параллельным объяснением того, что происходит, наконец, с демонстрацией работы самой программы. Наконец, очень важным является показ того, как работать с тем или иным инструментом программирования: средами, отладчиками, профайлеров, систем тестирования и т. д. Конечно, все это может присутствовать и в традиционном изложении лекционного материала. Но, во-первых, есть ограничение технического порядка, о котором было сказано выше, во-вторых, определенным сдерживающим фактором была и традиция чтения лекций. В образовании подобные традиции живут долго. При переходе на онлайн-обучение процесс ухода подобных традиций ускорится.

4. Важнейшим фактором, влияющим на усвоение материала, является возможность записи читаемой лекции. Записанная лекция или ее важный фрагмент по усмотрению преподавателя может быть предоставлен для последующего просмотра студентами, например, через облачные сервисы. В результате, во-первых, студент всегда может повторить не понятный фрагмент ли просмотреть лекцию, если он по какой-то причине отсутствовал на ней. В определенном смысле снимается проблема «надиктовки» материала, которая часто используется в традиционных лекциях и постепенно себя изживает. Такие записанные лекции становятся частью тех источников, которые могут рекомендоваться к самостоятельному ознакомлению студентам вместе с книгами, статьями, веб-сайтами и другими ресурсами.

5. Важным аспектом онлайн-лекций является то, каким образом студент выходит на связь с лектором. Возможность выхода на видеоконференцию через ноутбук или стационарный компьютер (а не через телефон или планшет) позволяет наладить полноценную обратную связь между студентом и преподавателем. В результате можно строить лекцию с элементами практического занятия, что значительно повышает ее ценность. Например, на лекциях по языку управления реляционными базами данных SQL, преподаватель вместе с объяснением нового материала может давать студентам задания, которые они тут же могут выполнять. При этом они заранее получают учебную заполненную базу данных, которую устанавливают на имеющиеся на их компьютерах СУБД. Лекция при этом читается с демонстрацией на той же самой базе данных. В качестве заданий преподаватель предлагает написать ту или иную команду у себя на компьютере, проверить ее, а потом переслать лектору. При этом последний тут же может продемонстрировать ее работу у себя на компьютере и объяснить ошибки, если в этом есть необходимость. В обычной аудиторной лекции наладить вот такой тип взаимодействия довольно сложно.

### **Особенность проведения практических занятий**

Практическая деятельность в области программирования и разработки информационных систем можно разделить два класса: решение программных задач и разработка проектов (проектная деятельность). Остановимся в начале на первом виде деятельности.

1. Для овладения языком программирования и выработке алгоритмического мышления нужен большой объем практики. Программированию учатся только решением большого количества задач. Рассмотрим, например, язык программирования Pascal. Можно выделить

следующие темы для изучения: начала программирования, простейшие программы; переменные и типы переменных; консольный ввод-вывод; арифметика целых чисел; структура программ; условные конструкции, оператор выбора; цикл for; цикл while; цикл repeat; пользовательские типы; вложенные циклы; одномерные массивы; двумерные массивы; процедуры и функции; математические функции; основы рекурсивного программирования; основы управления файлами; основы динамических структур данных.

Нами перечислены восемнадцать тем по основам языка программирования Pascal. Список далеко не полон, поскольку можно выделить еще несколько тем, связанных с изучением алгоритмов, например поиск, сортировки, деревья и т. д. Далее идет большая область изучения, связанная с объектно-ориентированным программированием, которая также делится на большое количество тем. Для освоения той или иной темы требуется решить порядка десяти задач. Получается, что количество задач, которые должен выполнить, студент для освоения основ программирования на одном из алгоритмических языков исчисляется несколькими сотнями. Для того, чтобы зачесть все эти задачи, они должны проверены. При чем речь идет еще и о том, что студент должен решать свой вариант. Работа по проверке огромна, даже если преподаватель будет использовать какие-либо средства автоматизации тестирования на основе готовых тестовых заданий.

Онлайн-занятия в значительной степени подтолкнула к решению подобной проблемы, хотя во многих вузах к автоматизации проверок пришли уже гораздо раньше. Решением подобной проблемы является использование специализированных автоматизированных систем для проверки программных заданий. Конечно, при этом требуется, чтобы программа имела стандартный ввод и вывод данных. Но для изучения основ программирования этого вполне достаточно. Да, некоторые крупные университеты имеют свои сервера для автоматизации проверки программных заданий<sup>3</sup>. Но в настоящее время в открытом доступе имеются ресурсы, позволяющие в автоматическом режиме осуществлять проверку программных заданий на основе некоторого набора тестов. Такие ресурсы можно разделить на две категории:

- позволяющие самим формировать задания и набор тестов;
- ориентированные на некоторый язык программирования и содержащие уже готовый набор заданий и тестов для их проверки.

К первой категории можно отнести такой открытый ресурс как Яндекс.Контест<sup>4</sup>. Конечно, он изначально декларировался как сервер для организации олимпиад по программированию (также математики). Но преподаватель всегда может готовить материал для аудиторного занятия или для домашнего выполнения студентов.

В качестве примера ресурса второй категории можно указать сайт Информатикс<sup>5</sup>, где имеется набор типовых задач по программированию для начинающих, с возможностью проверить решение на наборе подобранных тестов. В последнее время стали появляться сайты по изучению определенных языков программирования, где также можно на основе существующих там тестов отработать навыки программирования<sup>6</sup>.

Описанный подход можно использовать и на занятии, и для выполнения домашних заданий. Особо следует отметить именно онлайн-занятия. В этом случае преподаватель может

---

<sup>3</sup> Примером сервера для автоматизации проверки, который может быть свободно установлен любым учебным заведением на своих ресурсах, служит Ejudge (см. <https://ejudge.ru/>).

<sup>4</sup> Регистрация на ресурсе совершенно свободна по адресу <https://contest.yandex.ru/edu>.

<sup>5</sup> Расположен по адресу <https://informatics.msk.ru/>.

<sup>6</sup> Например, сайт по изучению языка Python <http://pythontutor.ru/>.

контролировать выполнение заданий студентов непосредственно в проверяющей системе. Это гораздо удобнее, чем просить студента показать ему его текущую работу. Конечно, выполняемым заданиям, могут предъявляться дополнительные требования, кроме необходимости пройти заданные тесты. Например, это может быть требование использовать в программе конкретный набор средств. Но вот именно это уже может сделать визуальная проверка преподавателя.

Описанный выше подход, несомненно, не только останется при переходе к обычной форме обучения, но и, видимо, станет основным при проверке программных заданий как в учреждениях высшего образования, так и в школах. В следующем разделе статьи мы подробнее остановимся на некоторых вопросах подбора задач и тестировании их решений.

2. Автор данных строк довольно давно занимается проектным подходом в преподавании дисциплин, ориентированных на разработку информационных систем. Проверка результатов работы здесь не сводится только к тестированию конечного продукта. Оцениваются разные стороны разработки: структура базы данных, интерфейс пользователя, наличие документации, соответствие функциональным и техническим требованиям. Не в последнюю очередь на оценку влияет и доклад разработчиков. Заметим, что взаимодействие разработчиков и преподавателя в онлайн-режиме дает полный спектр возможностей проверки результатов работы. Для проверки работы проекта преподавателю достаточно иметь себя настроенную виртуальную машину: СУБД, web-сервер, языки программирования и библиотеки.

При дистанционной работе на основе проектного подхода взаимодействие студента и преподавателя может быть основано на принципе непрерывной доставки<sup>7</sup> [15]. При этом средством доставки могут и электронная почта, и общий облачный каталог и каталог git. Есть и другой подход к решению проблемы проектов. Каждый студент получает свою виртуальную на машину на ресурсах Вуза. В его задачу, таким образом, будет входить не только разработка проекта, но и полная настройка виртуальной машины для такой разработки. Для проверки проекта преподавателю достаточно получить доступ либо к самой виртуальной машине студента, либо ссылке на сайт, размещенный на данной машине. Наконец сам доклад о разработанном проекте может быть сделан в одной из видеоконференций, при чем для прослушивания этого доклада можно привлечь и других преподавателей кафедры.

### Использование автоматической проверки задач

Выше мы говорили об обучении языкам программирования и алгоритмизации путем автоматической проверки решений. Это большой и серьезный вопрос для преподавателя в первую очередь потому, что разработка заданий и тестов к ним — это большой объем работы. Остановимся на некоторых ее аспектах.

1. **Тексты заданий.** Разработка текстов заданий это большая работа. Предполагается, что все задания рассчитаны на разработку консольных программ, имеющих на выходе некоторый набор данных и передающих на стандартное устройство вывода результаты — также некоторый набор данных. Проверка решений таких задач предполагает, что автоматизированная система сверяет результаты работы программы при данном наборе входных данных с некоторым тестовым эталоном или набором эталонов. Если сверка дала положительный результат, то считается, что тест пройден. Строгость подхода к проверке задания предполагает строгость формулировок текста задач. Содержание задачи должно

---

<sup>7</sup> Данный принцип основывается на требовании, согласно которому программный продукт должен всегда находиться в рабочем состоянии (см. «Зачем нужны непрерывная доставка и непрерывное развертывание?» <https://habr.com/ru/company/custis/blog/557540/>).

обладать полнотой и не иметь каких-либо двойных толкований. Например, если речь идет о числах, то в тексте задачи обязательно указывается диапазон их значений. Для вещественных чисел обязательно оговаривается их точность и точность результата. Для массивов чисел и текстовой информации также указываются диапазоны: максимальное и минимальное количество элементов массива, количество строк и их длин. Возможно также указание признаков окончания ввода данных (количество вводимых данных в начале или признак того, что ввод закончился). Только при четкой формулировке задачи можно говорить о возможности ее тестовой проверки.

Мы выделяем два вида текстов задач. К первому типу относятся тексты, не содержащие лишней информации, кроме четких описаний входной и выходной информации с учетом частных случаев и диапазонов значений. Следует иметь в виду, что и сам текст задания также является учебным инструментом, демонстрирующим студенту то, как нужно формулировать задания. Ко второму типу задач мы относим задачи, имеющие некоторый сюжет. Часто авторы пытаются сделать сюжет занятным и даже проводить сюжетную линию через несколько заданий. Чем мотивируется такой подход к формулировкам задач? Конечно не только и не столько желанием сделать их не скучными для студента. Скорее наоборот, усложнить для обучаемого анализ текста. На наш взгляд это очень значимый инструмент подготовки будущего специалиста в области разработки программного обеспечения. Частью любой подобной разработки является этап анализа предметной области, в котором исследование текстов является пожалуй самой важной составляющей [16].

**2. Тестирование решений.** Для автоматизированной проверки заданий необходим набор тестов. Выделим важные аспекты этого процесса.

а) При разработке тестовой проверки есть важное требование. Это ограничение по используемым ресурсам: объем памяти, время выполнения теста, использование файлов, использование Интернет и др. Регулирование этих ограничений может побуждать студента более глубоко и творчески понять проблему, которая заключена в решении конкретной задачи, в алгоритме решения. Современные системы проверки задач позволяют настраивать подобные ограничения не только для конкретной задачи, но и для конкретного теста, хотя это и дополнительная нагрузка для того, кто эти задания готовит.

б) Набор тестов может быть разделен на несколько частей. Каждая из частей дает свой вклад в оценку выполнения задания. Чаще всего речь идет об выделении группы тестов с большим объемом данных — большим количеством чисел, строк, символов, большими числами. Очень часто стандартный алгоритм решения задачи для таких тестов дает слишком большое время выполнения или требует слишком большой памяти. Если решение работает и для такой группы тестов, то за задачу начисляются дополнительные баллы.

в) Важной особенностью автоматических систем проверки задач является наличие в них инструмента предварительной проверки решения текста программы. Обычно система поддерживает некоторый язык программирования, позволяющий проводить анализ текстов. Это еще один способ регулирования, тонкой настройки системы под конкретную изучаемую тему. Такая подстройка, например, не пропустит решение, с оператором цикла `while`, если изучаемая тема циклы `for`. Конечно, в некоторых случаях такая подстройка может оказаться достаточно сложной, например, обязательное требование решить задачу рекурсивно. Но может избавить преподавателя от проверок большого количества текстов программ, только для того, чтобы отметить, была ли использована в программе данная технология или нет.

В заключение раздела приведем пример авторского текста задачи. Чтобы не перегружать текст статьи, мы опускаем набор тестов для данной задачи.

## На первый — второй рассчитайся

### Фабула

Тот, кто служил в армии, знает команду «на первый, второй — рассчитайся!». По этой команде военнослужащие в шеренге слева направо по очереди говорят «первый» или «второй». Команда дается, чтобы потом разделить всю группу военнослужащих на две подгруппы. Иногда требуется разделить группу на три подгруппы и тогда дается команда «на первый, второй, третий — рассчитайся!». Команду можно обобщить: «на первый, второй, ...  $n$ -й — рассчитайся!».

### Условие задачи

Пусть в последовательности (строю) находится  $t$  человек, дается «команда»: «на первый, второй,  $n$ -й — рассчитайся!». Вывести получившиеся списки групп.

### Входной файл input.txt.

В первой строке содержится число:  $n$ ,  $n > 1$ . В следующих строках идут фамилии, стоящих в шеренге, по одной на каждую строку. Количество стоящих в шеренге ( $t$ ) больше  $1$  и не превосходит 100000. Количество фамилий  $t$  и число  $n$  никак друг с другом не коррелируют.

### Выходной файл output.txt.

Содержит группы фамилий. В начале идут фамилии, получившие номер 1, потом получившие номер 2 и т. д. до  $n$ . По одной на каждую строку. Причем должен соблюдаться их порядок следования в основном строю. Между группами фамилий — пустая строка.

### Ограничение

Время выполнения программы не более 2 с.

### Примеры

| Входной файл input.txt | Выходной файл output.txt |
|------------------------|--------------------------|
| 3                      | Petrov                   |
| Petrov                 | Rostkov                  |
| Ivanov                 | Vasilev                  |
| Sidorov                |                          |
| Rostkov                | Ivanov                   |
| Potehin                | Potehin                  |
| Dozmorov               | John Silver              |
| Vasilev                |                          |
| John Silver            | Sidorov                  |
|                        | Dozmorov                 |
| 4                      | abc                      |
| abc                    |                          |
| bac                    | bac                      |
| cba                    |                          |
|                        | cba                      |

## Выводы

Мы рассмотрели некоторые аспекты, связанные с переходом к дистанционному обучению в учреждениях высшего образования по техническим дисциплинам, связанным с программированием. Дистанционное обучение выявило ряд подходов при проведении занятий, обогатив тем самым сам процесс обучения, в том числе и его очную форму.

1. Возможности онлайн-лекций с точки зрения подачи информации обогащают учебный процесс: возможность записи занятия, и в последствие использование их в учебном процессе, использование дополнительных технических средств, чему способствует возможность преподавателя самому выбирать аппаратуру и программное обеспечение, простота показа процесса программирования, проектирования, отладки и т. д. Подходы отработанные в процессе онлайн-обучения могут теперь вполне быть реализованы при традиционном образовании.

2. Практические занятия в онлайн-режиме можно сделать эффективными, если использовать автоматическую проверку заданий и это возможность должна обязательно перенесена на занятия в обычном формате. В настоящее время такие возможности есть у всех Вузов. При работе с проектами онлайн режим не менее эффективен, чем аудиторный, поскольку процесс получения преподавателем результатов работы и их обновление легко реализуем с помощью различных инструментов. Определенные элементы, несомненно, должны быть перенесены в обычный режим работы.

## ЛИТЕРАТУРА

1. Андреев, А.А. Дистанционное обучение: сущность, технология, организация / А.А. Андреев, В.И. Солдаткин — Москва, 1999. — 196 с. — Текст: непосредственный.
2. Положенцева, И.В. Содержание, базовые понятия и научное обоснование дистанционного образования / И.В. Положенцева. — Текст: непосредственный // Вестник московского государственного областного университета. — 2016. — Серия: Педагогика. — № 3. — С. 24–32.
3. Логинова, Н.С. Дистанционное обучение: проблемы и варианты их решения (на примере обобщения опыта дистанционного обучения в АГМУ) / Н.С. Логинова, А.Ю. Бендрикова, С.И. Дегтярёв. — Текст: непосредственный // Межкультурная коммуникация в образовании и медицине. — 2021. — № 3, С. 16–19.
4. Домрачев, В.Г. Дистанционное обучение: возможности и перспективы / В.Г. Домрачев. — Текст: непосредственный // Высшее образование в России. — 1994. — № 3. — С. 11.
5. Тихомиров, В.П. Дистанционное обучение: к виртуальным средам знаний / В.П. Тихомиров, В.И. Солдаткин, С.Л. Лобачев, О.Г. Ковальчук. — Текст: непосредственный // Дистанционное образование. — 1999. — № 2. — с. 3–4.
6. Пирогов, В.Ю. Некоторые вопросы преподавания программирования / В.Ю. Пирогов, Е.И. Попова. — Текст: непосредственный // Мир науки. Педагогика и психология. — 2019. — Т. 7. — № 6.
7. Пирогов, В.Ю. Некоторые особенности преподавания языка управления базами данных / В.Ю. Пирогов. — Текст: непосредственный // Мир науки. Педагогика и психология. — 2018, — Т. 6 — № 6.
8. Баландина, И.В. Организация дистанционного обучения в вузе в современных условиях / И.В. Баландина, Е.И. Попова. — Текст: непосредственный // Вопросы педагогики. — 2020, — № 6–2. — С. 34–38.



9. Дедюхин, Д.Д. Дистанционное обучение в системе высшего образования: проблемы и перспективы / Д.Д. Дедюхин, А.А. Баландин, Е.И. Попова. — Текст: непосредственный // Мир науки. Педагогика и психология. — 2020. — Т. 8. — № 5.
10. Иванова, Л.В. Методы и формы обучения программированию в вузе / Л.В. Иванова, В.Е. Чекушина. — Текст: непосредственный // Сборник научных трудов SWorld. — 2013. — Т. 17. — № 3. — С. 18–22.
11. Иванова, Л.В. Методические аспекты обучения программированию будущих учителей информатики / Л.В. Иванова. — Текст: непосредственный // Информатика: проблемы, технологии: материалы XII Международной научно-методической конференции (9–12 февраля 2012). — Воронеж, 2012. — Т. 2. — С. 92–94.
12. Кулигина, Н.О. Методика преподавания и оценки знаний студентов по программированию на кафедре АИС ДПИ НГТУ / Н.О. Кулигина. — Текст: непосредственный // Сборник трудов конференции Нижегородского государственного технического университета им. Р.Е. Алексеева. — 2014. — С. 198–202.
13. Рогович, В.И. Организация лекций и самостоятельной работы при дистанционном обучении с учетом болонской модели образования и модели зрелости электронного обучения / В.И. Рогович, К.А. Данилов. — Текст: непосредственный // Информационные системы и технологии. — 2009. — № 6(56). — с. 79–87.
14. Пирогов В.Ю. Основы методики обучения рекурсивному программированию / В.Ю. Пирогов, И.В. Баландина. — Текст: непосредственный // Интернет-журнал «Мир науки». — 2018. — Т. 6. — № 4.
15. Пономарев, В.А. Опыт дистанционного преподавания дисциплины «системное программирование» / В.А. Пономарев, Н.А. Баженов, Е.И. Рыбин. — Текст: непосредственный // Цифровые технологии в образовании, науке, обществе: материалы XIV всероссийской науч.-практ. конф. (01–04 декабря 2020 года). — Петрозаводск, 2020. — С. 128–130.
16. Пирогов, В.Ю. Информационные системы и базы данных [Текст] / В.Ю. Пирогов. — СПб.: БХВ, 2009. — 528 с.

**Pirogov Vladislav Jurievich**

Shadrinsk Pedagogical University, Shadrinsk, Russia  
E-mail: Vladislav-133@yandex.ru

## **On the possibilities of distance teaching of programming for students of technical specialties**

**Abstract.** The transition to distance learning in connection with the spread of the coronavirus pandemic has highlighted some approaches in teaching that can enrich traditional approaches to education. The article discusses some particular aspects related to the teaching of disciplines in institutions of higher education for students of technical specialties studying disciplines related to the study of programming. The author considers two types of classes: lecture and practical. The article shows that the possibilities of online lectures from the point of view of providing information enrich the educational process: the ability to record a lesson and then use it to prepare students, the use of additional technical means, which is facilitated by the fact that the teacher chooses the hardware and software himself. In addition, the author points out the simplicity of showing the programming process, design, debugging, etc. The author points out that practical classes in online mode can be made effective if you use such a tool as automatic task verification. It is indicated that currently there are a number of tools that allow you to organize this type of educational work. The article notes that this opportunity must necessarily be transferred to classes in the usual format, since this can significantly increase the effectiveness of developing practical programming skills among students. The article also discusses the issue of working with software projects. The author points out that the online mode is also no less effective here than the classroom one. Certain elements should undoubtedly be moved to normal operation.

**Keywords:** distance learning; programming; online lectures; software projects; programming training; software; debugging