

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Ульяновский государственный  
педагогический университет имени И.Н. Ульянова»



# **Управление качеством образования: проблемы и перспективы**

Материалы Всероссийской научно-практической конференции, посвященной 50-летию  
создания кафедры методики преподавания математики УлГПИ

Ульяновск, 2024

***Министерство просвещения РФ***

***Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Ульяновский государственный педагогический  
университет имени И.Н. Ульянова»***

***ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКОГО И  
ТЕХНОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ***

***Управление качеством образования:  
проблемы и перспективы***

*Материалы  
Всероссийской научно-практической конференции, посвященной 50-летию  
создания кафедры методики преподавания математики УлГПИ*

*7-8 декабря 2023 года*

*Ульяновск  
2024*

**УДК 37** Печатается по решению редакционно-издательского совета  
**ББК 74.00** **ФГБОУ ВО «УлГПУ имени И.Н. Ульянова»**  
**У 67**

**Ответственный редактор:**

**Н. В. Сидорова**, кандидат педагогических наук, доцент, заведующий кафедрой методик математического и информационно-технологического образования ФГБОУ ВО «УлГПУ им. И. Н. Ульянова».

**Редакторы:**

**Д. И. Беркутова**, кандидат педагогических наук, доцент, доцент кафедры методик математического и информационно-технологического образования ФГБОУ ВО «УлГПУ им. И. Н. Ульянова».

**М. С. Нуждина**, старший лаборант кафедры методик математического и информационно-технологического образования ФГБОУ ВО «УлГПУ им. И. Н. Ульянова».

**Рецензенты:**

**Н. О. Седова**, доктор физико-математических наук, профессор кафедры информационных технологий ФГБОУ ВО «УлГУ».

**В. Г. Шубович**, доктор педагогических наук, кандидат технических наук, профессор, заведующий кафедрой информатики ФГБОУ ВО «УлГПУ им. И. Н. Ульянова».

**У 67 Управление качеством образования: проблемы и перспективы:**  
Материалы Всеросс. научно-практ. конф., посвящ. 50-летию создания кафедры методики преподавания математики УлГПИ / Ответ. ред. Н. В. Сидорова. – Ульяновск: ФГБОУ ВО «УлГПУ им. И. Н. Ульянова», 2024. – 359 с.

В сборнике представлены материалы Всероссийской научно-практической конференции преподавателей и студентов образовательных учреждений, прошедшей на базе факультета физико-математического и технологического образования Ульяновского государственного педагогического университета им. И.Н. Ульянова 7-8 декабря 2023 года.

В числе проблем, касающихся повышения качества образования, освещенных в статьях, можно указать следующие: управленческие аспекты, влияющие на качество образовательных услуг, диагностика образовательных результатов, информатизация профессионального среднего и дополнительного образования, формирование цифровых компетенций обучающихся, приобщение учащихся к проектно-исследовательской деятельности, формирование функциональной грамотности школьников, методические особенности изучения отдельных учебных тем и ряд других.

Статьи публикуются в авторской редакции.

© ФГБОУ ВО «УлГПУ им. И.Н. Ульянова», 2024

© Авторский коллектив, 2024

<b>Иевлева Д. А., Шмакова А. П.</b> Применение визуального программирования при изучении английского языка	<b>280</b>
<b>Колединский Д. А.</b> Цифровая трансформация методов обучения иностранному языку	<b>283</b>
<b>Лубянина А. В., Горшкова Т. А.</b> Практические аспекты геймификации на уроках технологии	<b>287</b>
<b>Марунина Т. В., Шмакова А. П.</b> Использование визуальных технологий на уроках иностранного языка в средней школе	<b>291</b>
<b>Марченко Ю. В.</b> Технология веб-квест как средство повышения мотивации на уроке информатики	<b>294</b>
<b>Назарова Е. А.</b> Искусственный интеллект в образовании	<b>299</b>
<b>Некрасов В. А., Шубович М. В., Шубович В. Г.</b> Информационная культура студентов как предмет исследования	<b>302</b>
<b>Рамазанов Э. Р., Бутырева С. А.</b> Преодоление языкового барьера в FRONEND-разработке: необходимый стартовый набор английских слов	<b>308</b>
<b>Редькина О. Д., Горшкова Т. А.</b> Цифровые образовательные ресурсы: новые возможности для обучения технологии в школе	<b>312</b>
<b>Селезнева Е. К., Шмакова А. П.</b> Технология разработка тестовых заданий для младших школьников	<b>316</b>
<b>Серебрякова П. И.</b> Интерактивные формы обучения на уроках технологии	<b>319</b>
<b>Сибирев И. В.</b> Инструментарий в преподавании дисциплины «Основы алгоритмизации» для СПО	<b>321</b>
<b>Симонова Е. А., Солощенко М. Ю.</b> Технология создания электронного учебного пособия на тему «Обыкновенные дроби» для 5-6 классов с помощью программы TURBOSITE	<b>328</b>
<b>Степанова А. Ю.</b> Интернет сайты как средство повышения качества подготовки к ОГЭ по математике	<b>334</b>
<b>Талицких А. А.</b> Аксиология цифровой образовательной среды	<b>336</b>
<b>Токарев Я. Е., Рамазанов Э. Р.</b> Формирование инженерных навыков в дополнительном образовании по 3D-печати	<b>339</b>
<b>Утлова К. А., Сидорова Н. В.</b> Средства организации профессионального самоопределения школьников	<b>346</b>
<b>Федорова А. В., Федоров В. Н.</b> Нейросеть как средство актуализации познавательной деятельности школьников на уроках географии	<b>348</b>
<b>Сведения об авторах</b>	<b>352</b>

Ссылка на QR-код:



Ссылка на диск:

<https://disk.yandex.ru/d/Xs0rwTT9vY1TDw>

Подводя итог, можно отметить то, что WPS Office обладает всеми необходимыми для работы функциями и вполне может помочь разбавить уроки не только технологии, но и любой другой дисциплины.

#### **Список литературы**

1. Веселовская Ю.А. Online-сервисы совместной работы как средство формирования метапредметных образовательных результатов обучающихся в процессе внеурочной деятельности по математике / Ю.А. Веселовская, А.П. Шмакова, Н.В. Сидорова // Педагогическое образование / Pedagogical Education. – 2022. – Том 3. – № 7. – С. 67–70.
2. Наумова А.Н., Сидорова Н.В. Использование инфографики в организации проектной деятельности школьников на уроках информатики // Управление качеством образования: проблемы и перспективы: Материалы Всеросс. научно-практ. конф., посвящ. 100-летию С.Г.Первухиной. – Ульяновск: ФГБОУ ВО «УлГПУ им. И.Н. Ульянова», 2023. – С. 351–356.
3. Шмакова А.П. Интерактивные технологии в современной школе: методические рекомендации для zoom-педагога. – Ульяновск: ФГБОУ ВО «УлГПУ им. И.Н. Ульянова», 2021. – 28 с.

### **ИНСТРУМЕНТАРИЙ В ПРЕПОДАВАНИИ ДИСЦИПЛИНЫ «ОСНОВЫ АЛГОРИТМИЗАЦИИ» ДЛЯ СПО**

**И. В. Сибирев**, преподаватель информатики

Колледж информатики и программирования ФГОБУВО «Финансовый университет при Правительстве Российской Федерации», Москва

При обучении программистов важно использовать современные способы хранения, передачи программных кодов, комментариев к ним, соответствующей теории, задач и образцов их решения. В данной статье описан инструментарий для организации хранения и передачи информации, программных кодов в процессе преподавания ИТ дисциплин.

**Ключевые слова:** несколько языков программирования в блокноте Jupiter, технология магических слов.

### **TOOLKIT FOR TEACHING THE DISCIPLINE “FUNDAMENTALS OF ALGORITHMIZATION” FOR SECONDARY VOCATIONAL EDUCATION**

**I. Sibirev, Moscow**

Training programmers requires modern methods of storing, transmitting and using program codes, comments on them, theories, problems and examples of their solutions. This article describes the tools for organizing the storage and transmission of information and program codes in the process of teaching IT disciplines.

**Keywords:** several programming languages in Jupiter notebook, magic word technology.

Качество образования, в частности, будущих программистов в СПО, во многом зависит от способа хранения и передачи информации, программных кодов. Существующие способы не обладают должной эффективностью хранения

знаний в плане скорости поиска и доступа, поддержания в актуальном состоянии.

1. В школе и в университете, скорее всего, мы писали лекции в тетради. Если это лекции, связанные с ИТ дисциплинами, то такой способ хранения, доступа, и обработки информации не выдерживает никакой критики в условиях современного темпа жизни. Потому что, к примеру, материалы дисциплины «Основы алгоритмизации», изложенные в виде лекций в тетради, будут иметь крайне низкую скорость доступа к ним и обработки. К тому же, в таком виде они не несут должного объема знаний для поддержания профессиональной актуальности и конкурентоспособности на рынке труда. Хранение знаний в таком формате требует многократного обращения к ним, непрерывной актуализации и модификации, что затруднительно. Частота обращений превышает разумные пределы, предусмотренные способом хранения и носителем информации. Как быть?

**Проблема** – требуются актуальные способы хранения и обработки информации при профессиональной подготовке программиста, не требующие больших вложений ресурсов для обработки и поддержания.

Применимы ли некоторые известные системы контроля версий для решения поставленной проблемы?

2005 г., Git [3] – система контроля версий, содержит папки проекта программных кодов. Использует центральный репозиторий и локальный репозиторий для редактирования сайтов. Система полностью децентрализована и может работать без Интернета. Программное обеспечение свободно распространяемое. Позволяет вести поочередное командное редактирование файлов и хранить историю версий. Сам по себе Git не является решением проблемы, отчасти потому что Git консольный, работать с ним без подготовки трудно. На начальных этапах можно воспользоваться графическими интерфейсами для упрощения работы, например, Git Extensions [4].

2008 г., GitHub [6] – сайт-сервер для централизованного хранения репозитория программных кодов и файлов свободно распространяемого программного обеспечения. Из минусов: базируется в Сан-Франциско; в вопросах безопасности проект оставляет желать лучшего; при выкладывании важной информации – есть риск кражи аккаунта, поэтому не подходит для коммерческих проектов; не работает скачивание архивом; заниженная скорость для России. Плюсы: он есть, он работает, он позволяет просматривать исходные файлы и скачивать файлы по-отдельности, позволяет выкладывать HTML файлы и генерировать ссылку для их отображения у конечного пользователя (поисковиком не индексируется).

Отечественный аналог – GitFlic [5]. Коммерческая защищенная версия – GitLab [7]. Не являются решением проблемы, так как это специализированные сервисы, не рассчитанные на начальный уровень освоения.

2011-2015 г. – Jupiter Notebook (Anaconda) [1]. Это блокнот для редактирования программных кодов в браузере, поддерживает HTML, CSS, JavaScript и прочие «блага» веб-разработки, очень хорошая визуализация, не

зависящая от производителя программного обеспечения. Есть возможность создать файл, скачать, сохранить его в формате \*.ipynb, \*.HTML, \*.PDF, что позволяет выкладывать его на сайт, например, через GitHub. Есть возможность запускать программные коды, полная независимость от окружения. При некоторой настройке можно страницу «вынуть» из локальной сети и разместить на сайте организации, в Интернете, как пример запускаемых программных кодов или лекцию. Jupiter Notebook по праву считается идейным потомком и актуальной заменой, в том числе, и для Mathcad.

Но есть и минусы. Jupiter Notebook написан на Python, большая часть обновлений и настроек – только через консоль. Основной язык программирования логики – Python, что требует времени, трудозатрат, объемных программных кодов, конечный результат не всегда компактен. Требует навыков поиска информации и программирования. Сам по себе не является готовым решением проблемы в полной мере. Тем не менее – это удобно для конечного потребителя лекционного материала.

2. В данной статье рассказывается об опыте использования описанных технологий в преподавании «Основ алгоритмизации» и других предметов, связанных с ИТ, в рамках среднего профессионального образования.

Основной проблемой Jupiter Notebook при решении поставленной проблемы является ядро страницы \*.ipynb, то есть язык, на котором программируется логика. В электронном источнике [10] приведены часто используемые здесь языки, это Python и др. Среди «других»: язык Julia, пакет IJulia [8], язык R, пакеты zmq и irkernel [17], язык Ruby, пакет IRuby [18]. Для них требуется установка и настройка специальной интерактивной оболочки.

С другими языками еще сложнее. Для языка C# есть инструкция [14], требующая .NET Core 3.0 SDK, который больше не поддерживается, но все еще доступен для установки. Язык C++ требует, чтобы все это было развернуто на операционной системе Linux. Есть WCL – “Windows Subsystem for Linux”, который позволяет развернуть консоль Bash/Linux под Windows, поставить туда ядро GNU Linux, затем G++, потом интерфейс xeus-cling [2]. Желание связываться с обильными и ненадежными инструкциями невелико. К тому же, большинство часто используемых средств визуализации и механизмов разработано именно под Python. Использование в качестве ядра блокнота Jupiter другого языка зачастую отрицает «все прошлые достижения человечества».

Есть несколько альтернатив. Первая – писать свой интерфейс IProgramLanguage на все случаи жизни. Это создаст еще больше проблем, а именно, лавинообразный рост трудозатрат на переписывание всего, что было ранее, по шаблону, с поддержкой старого функционала под каждый конкретный случай. Многие именно это и делают. Если судить по многообразию и сложности инструкций настроек и установок, то можно сделать вывод, что это не очень получается.

Автором статьи предлагается довольно простая альтернатива – использовать имеющийся инструментарий языка Python с сохранением многообразия визуализации для решения этой проблемы.

В интерфейсе IPython есть технология «магических слов» [13]. Магические слова могут быть однострочными и многострочными. Интересны многострочные магические слова, они могут ячейку исполняемого кода Python превратить в ячейку кода, например, HTML. Доступные магические слова: “%%bash”, “%%html”, “%%javascript”, “%%latex”, “%%perl”, “%%python2”, “%%python3”, “%%ruby”, “%%cmd”. Как видите, доступно много языков, способных уживаться на обычном ядре интерфейса IPython. При этом при переписывании функционала под другие языковые интерфейсы, это многообразие теряется. У магических слов есть и минус. Ячейки становятся отдельно существующими, то есть к ним крайне затруднительно, но не невозможно, программно передать параметр из других ячеек.

3. Перед автором стояла цель настроить магическое слово для языка C++/G++ под Windows. Приведем инструкцию.

Шаг первый. Скачать C-Free 32 бита под Windows 10 [<https://developerinsider.co/download-and-install-borland-c-compiler-on-windows-10/>]. Да, это старенький компилятор Borland, и, как сказали на одном американском форуме: «Это технологии 90 годов, и компилятор ныне не рекомендуем ни для чего». Но есть несомненные плюсы: он просто ставится (без консольных настроек, установок и обновлений, смс и регистраций), он работает и позволяет четко разделить C и C++. Опыт нашего общения с ним изложен в материале [16].

Шаг второй. Сразу начинаем работать. В качестве ядра блокнота Jupiter используем Python. По сравнению с другими инструкциями – почти ничего не нужно делать. Первой ячейкой блокнота исполняемых программных кодов Python вписываем следующее.

#Config Cell Python 3
<pre> <b>from</b> IPython.core.magic <b>import</b> (register_line_magic,register_cell_magic) <b>def</b> Debag_Lang_Cpp(code):     <b>with</b> open('main.c', mode='w') <b>as</b> file:file.write(code)     !"C:\Program Files (x86)\C-Free 5\mingw\bin\c++.exe" main.c -o Start.exe     !Start.exe&lt;inPut.txt     @register_line_magic     <b>def</b> Cpp_line(line):         Debag_Lang_Cpp(line)     @register_cell_magic     <b>def</b> Cpp(line,cell):         Debag_Lang_Cpp(cell)     @register_cell_magic     <b>def</b> Input(line,cell):         <b>with</b> open('inPut.txt', mode='w') <b>as</b> file:file.write(cell)         Input(None," \n") </pre>

Получаем новое магическое слово “%% Cpp”.

Cell Python 3
%% Input



17
W
Cell Python 3
<pre>%%C++ #include &lt;stdio.h&gt; int main(int argc, char *argv[]) {     printf("Hello C-Free!\n");     int age=0;     scanf("%d\n", &amp;age);     printf("age = %d\n", age);     int res = 0;res=getchar();     printf("res = &lt;%c&gt;\n", res);     return 0; }</pre>
<pre>Hello C-Free! age = 17 res = &lt;w&gt;</pre>

В результате получена ячейка C++/G++ с минимальным числом шагов настроек и установок. Также реализован ввод значений через консоль.

А теперь о том, как это работает. Магическое слово “%%Input” создает заранее заготавливаемый файл с данными для ввода с консоли в папке блокнота Jpiter. Магическое слово “%%C++”, используя функцию Python `Debag_Lang_Cpp(Code)`, создает временный файл, посылает его на компиляцию, получает скомпилированный бинарник “Start.exe”, запускает его с перенаправлением на ввод из заранее подготовленного файла `inPut.txt`. Обратите внимание, что в этом контексте “!” – это аналог однострочного магического слова “%cmd”.

4. Эта инструкция настолько относительно проста, что ее смогли повторить группа учащихся второго курса колледжа (по меркам университета – первокурсники) за 30 минут с предварительно установленной Jupiter Notebook Anaconda.

Они получили новое магическое слово “%%Java”. Вкратце опишу их действия. Первое, установка средств разработчика JDK [<https://www.oracle.com/java/technologies/downloads/>]. Второе, замена в функции `Debag_Lang_Cpp(code)` строки полного пути к компилятору на `java`, так как предыдущий шаг создает соответствующую глобальную переменную.

Они получили в блокноте Jupiter сразу 3 исполняемых языка: Python, C++/G++, Java. Это «доселе невиданное зрелище».

5. Формат лекций в блокноте Jupiter удобен для конечного потребителя, но все еще трудозатратен для разработчика или преподавателя. Как быть? Постараемся предложить ответ и на этот вопрос, на примере языка программирования CMD/DOS/BAT при разработке программных кодов.

Шаг первый. Ставим Notepad++ [11]. При установке указываем локализацию «русская».

Шаг второй. Создаем скрипт. Создаем файл блокнот. Сохраняем его с расширением “\*.bat”. Открываем его с помощью Notepad++. Выбираем нужную кодировку. Для CMD/DOS/BAT кодировка «Кодировки/Кодировки/Кириллица/OEM 866».

```

*.BAT
.....
@echo off
if "%cd%" EQU "%~dp0" ((start notepad++ %0)&&(exit /b))
cd %~dp0
.....
set qwe=dsa;asda;
echo %qwe%
.....
(TIMEOUT /T 1)&&(pause)&&(exit /b)
.....

```

Сохраняем, закрываем редактор. Запускаем. Строка с ветвлением сравнивает директорию (папку), в которой лежит скрипт и точку запуска (целевую директорию приложения). Если они совпадают, то скрипт открывает свой исходный код в Notepad++ и завершает исполнение. Далее мы нажимаем “Ctrl+S” (горячая клавиша «сохранить»), “F5” (запуск скрипта) (см. рис.1.)

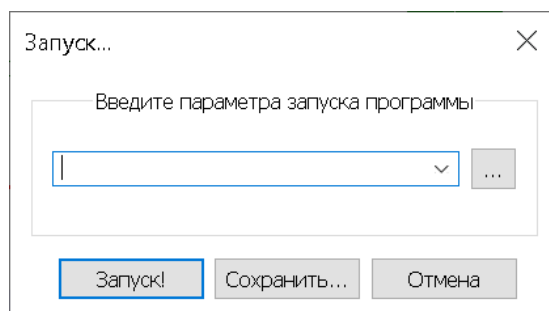


Рисунок 1. “F5” – запуск скрипта

В появившемся меню выбираем «сохранить». Открывается меню «Назначение клавиш быстрого доступа» (см. рис.2.).

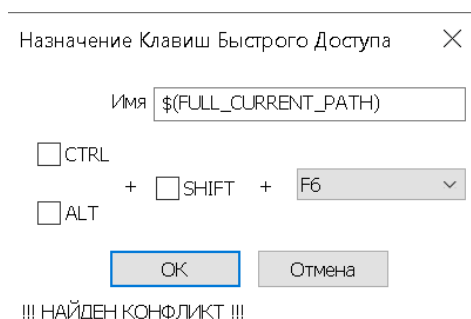
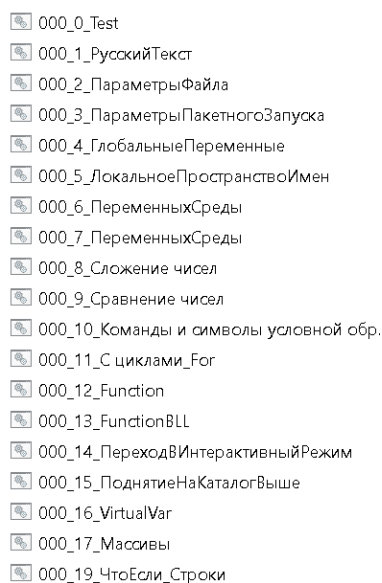


Рисунок 2. Меню «Назначение клавиш быстрого доступа»

В меню «Назначение клавиш быстрого доступа» в поле «Имя» вписываем параметр макроса “\$(FULL\_CURRENT\_PATH)” [12]. Назначаем горячую клавишу “F6”. Подтверждаем. Теперь “F6” позволяет запускать именно вкладку

с тем скриптом, «на который Вы смотрите». “F6” ни с чем не конфликтует. У нас это сообщение выводится по причине прошлой настройки. Убедиться в настройке можно «Запуск/Управление командами/...». Одна из настроек – только что назначенная горячая клавиша. Теперь нажимаем “Alt+F4” – горячая клавиша «закрыть текущее окно» (осторожно – при отсутствии окон выключает компьютер).



000\_0\_Test  
000\_1\_РусскийТекст  
000\_2\_ПараметрыФайла  
000\_3\_ПараметрыПакетногоЗапуска  
000\_4\_ГлобальныеПеременные  
000\_5\_ЛокальноеПространствоИмен  
000\_6\_ПеременныхСреды  
000\_7\_ПеременныхСреды  
000\_8\_Сложение чисел  
000\_9\_Сравнение чисел  
000\_10\_Команды и символы условной обр.  
000\_11\_С циклами\_For  
000\_12\_Function  
000\_13\_FunctionBLL  
000\_14\_ПереходВИнтерактивныйРежим  
000\_15\_ПоднятиеНаКаталогВыше  
000\_16\_VirtualVar  
000\_17\_Массивы  
000\_19\_ЧтоЕсли\_Строки

*Рисунок 3. Папка с примерами программных кодов*

Представим, что у нас есть папка, используемая в учебном процессе, с примерами программных кодов, в правильной кодировке, с комментариями к программным кодам (см. рис. 3). Откладываем мышь в сторону. Пользуемся только клавиатурой и вышеописанными горячими клавишами. Скорость работы и написания новых программных кодов, в том числе, с учебными примерами возрастает в десятки раз.

Подобным образом можно организовать свою работу на любом другом языке программирования. Строка с открытием в блокноте реализуема и на Python. С компилируемыми языками программирования дело обстоит чуть сложнее. В этом случае придется из скрипта делать временный файл, слать его на компиляцию, затем запускать результат, примерно, как с магическим словом в Jupiter. Принцип «один пример – один файл» будет действовать и в этом случае.

Предлагаемый инструментарий для организации хранения и передачи информации, программных кодов в процессе преподавания ИТ дисциплин апробирован в Колледже информатики и программирования Финансового университета при Правительстве Российской Федерации и показал свою эффективность.

#### ***Список литературы***

1. Anaconda – операционная система для искусственного интеллекта. [Электронный ресурс] – URL: <https://www.anaconda.com/> (дата обращения 9.12.23).
2. C++ для Jupyter Notebook [Электронный ресурс] – URL: <https://www.pranav.ai/cplusplus-for-jupyter> (дата обращения 9.12.23).
3. Git – система управления версиями с открытым исходным кодом [Электронный ресурс] – URL: <https://git-scm.com/> (дата обращения 9.12.23).

4. Git Extensions – автономный инструмент пользовательского интерфейса для управления репозиториями [Электронный ресурс] – URL: <https://gitextensions.github.io/> (дата обращения 9.12.23).
5. GitFlic – российский сервис для хранения исходного кода и работы с ним. [Электронный ресурс] – URL: <https://docs.gitflic.space/> (дата обращения 9.12.23).
6. GitHub – платформа для разработчиков на основе искусственного интеллекта. [Электронный ресурс] – URL: <https://github.com/> (дата обращения 9.12.23).
7. GitLab веб-инструмент жизненного цикла DevOps с открытым исходным кодом [Электронный ресурс] – URL: <https://about.gitlab.com/> (дата обращения 9.12.23).
8. Julia – работать с Julia в Jupyter Notebook [Электронный ресурс] – URL: <https://www.geeksforgeeks.org/how-to-work-with-julia-on-jupyter-notebook/> (дата обращения 9.12.23).
9. JDK [Электронный ресурс] – URL: <https://www.oracle.com/java/technologies/downloads/> (дата обращения 9.12.23).
10. Jupyter Notebook – интерактивная среда разработки. [Электронный ресурс]. Режим доступа: <https://jupyter.org/try> (дата обращения 9.12.23).
11. Notepad++ [Электронный ресурс] – URL: <https://notepad-plus-plus.org/downloads/> (дата обращения 9.12.23).
12. Notepad++: запуск файла из программы, плагины, горячие клавиши. [Электронный ресурс] – URL: <http://blog.kislenko.net/show.php?id=2618> (дата обращения 9.12.23).
13. Документация по работе с IPython. [Электронный ресурс] – URL: <https://ipython.readthedocs.io/en/stable/interactive/magics.html> (дата обращения 9.12.23).
14. Инструкция установки C# на Jupiter Notebook [Электронный ресурс] – URL: <https://habr.com/ru/companies/microsoft/articles/475110/> (дата обращения 9.12.23).
15. Материалы по C-Free 32 бита под Windows 10 [Электронный ресурс] – URL: <https://developerinsider.co/download-and-install-borland-c-compiler-on-windows-10/> (дата обращения 9.12.23).
16. Низкий старт [Электронный ресурс] – URL: [https://htmlpreview.github.io/?https://github.com/IvanSibirevV2/HowTo\\_CPP/blob/main/Тверь\\_2017/001\\_НизкийСтарт.html#/source\\_](https://htmlpreview.github.io/?https://github.com/IvanSibirevV2/HowTo_CPP/blob/main/Тверь_2017/001_НизкийСтарт.html#/source_) (дата обращения 9.12.23).
17. Установка Ruby на Jupiter Notebook [Электронный ресурс] – URL: <https://www.storybench.org/install-r-jupyter-notebook/> (дата обращения 9.12.23).
18. Язык Ruby, пакет IRuby [Электронный ресурс] – URL: <https://nithinbekal.com/posts/ruby-jupyter-notebooks/> (дата обращения 9.12.23).

## **ТЕХНОЛОГИЯ СОЗДАНИЯ ЭЛЕКТРОННОГО УЧЕБНОГО ПОСОБИЯ НА ТЕМУ «ОБЫКНОВЕННЫЕ ДРОБИ» ДЛЯ 5-6 КЛАССОВ С ПОМОЩЬЮ ПРОГРАММЫ TURBOSITE**

**Е. А. Симонова**, студент,

**М. Ю. Солощенко**, канд. пед. наук, доцент

Стерлитамакский филиал ФГБОУ ВО «Уфимский Университет Науки и Технологий»,  
Стерлитамак

В работе раскрыта технология создания электронного учебного пособия с помощью программы Turbosite.

**Ключевые слова:** электронное учебное пособие.

## **THE TECHNOLOGY OF CREATING AN ELECTRONIC TEXTBOOK ON THE TOPIC "ORDINARY FRACTIONS" FOR GRADES 5-6" USING THE TURBOSITE PROGRAM**

**E. Simonova, M. Soloshchenko, Sterlitamak**

The paper reveals the technology of creating an electronic textbook using the Turbosite program.