

УДК 004.55+378.147.31

## ЦИФРОВАЯ ТРАНСФОРМАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОСТРАНСТВА ЛЕКЦИИ ДЛЯ ОБУЧАЮЩИХСЯ ПРОГРАММИРОВАНИЮ

Беляков А.Ю.

ФГБОУ ВО Пермский ГАТУ, г. Пермь, Россия

Email: [belyakov@pgatu.ru](mailto:belyakov@pgatu.ru)

### АННОТАЦИЯ

Процесс разработки программного обеспечения усложняется с каждым годом. Появляются новые парадигмы программирования, растёт количество отдельных и самостоятельных понятий, абстракций и концепций проектирования. Всё более актуальным становится противоречие между повышением требований, предъявляемых к уровню подготовленности выпускника вуза и возможностями образовательного процесса. В данной работе рассматривается подход к организации образовательного пространства лекции для обучающихся программированию, основанный на использовании облачных технологий. Проводится сравнение функциональных возможностей существующих интернет-сервисов. Выделяется самая значимая функция – совмещение поясняющего текста лектора с программным кодом в едином пространстве, с возможностью его модификации, запуска на исполнение и получения результатов непосредственно в рамках документа. Совмещение теоретического материала с его практическим применением в течение лекции позволяет повысить эффективность образовательного процесса.

*Ключевые слова: облачные технологии, интерактивная лекция, компьютерная программа, прочность научения.*

### ВВЕДЕНИЕ

В процессе размышлений человек со средними возможностями может удерживать в своей кратковременной памяти от 5-ти до 9-ти мыслительных объектов. Эту закономерность, так называемый «Кошелёк Миллера», обнаружил и опубликовал ещё в 1956 году американский учёный-психолог Джордж Миллер [1]. В большинстве случаев в процессе обучения в вузе, в рамках обсуждения предметных областей учебных дисциплин, такого оперативного пространства хватает для осмысления и принятия закономерностей, логических связей и аналитических выводов.

Однако анализ образовательной программы и требований к выпускникам вузов по направлениям обучения, связанным с изучением современных технологий разработки программных продуктов, выявляет тенденцию к заметному расширению количественного состава элементарных единиц усвоения материала и интегрирующих их концепций и подходов. Конечно это связано, прежде всего, с технологической революцией и цифровым бумом в современном обществе. Развитие и

совершенствование микропроцессорной базы, проникновение цифровых технологий во все сферы деятельности человека обуславливает перманентно возрастающее многообразие подходов к разработке программного обеспечения и внушительный рост отдельных и самостоятельных понятий, элементов и концепций проектирования.

Таким образом, всё более актуальным становится противоречие между потребностями современного промышленного производства программных продуктов, в части касающейся требований к умениям и навыкам начинающих программистов, и возможностями образовательного процесса в вузе.

## МЕТОДИКА

Алгоритмы и программы, которые должен понимать, осваивать или разрабатывать современный студент соответствующего направления обучения, могут содержать десятки и сотни объектов разной степени сложности. Однако, компьютерные программы, написанные на современных языках программирования, подразумевают агрегирование сложности программного кода в отдельные самостоятельные сущности: фреймворки и библиотеки, классы и объекты, функции и методы, коллекции и переменные. Такой подход позволяет современному программисту читать, понимать и модифицировать программу в сотни и тысячи строк кода, при условии, что он в состоянии осознавать и манипулировать агрегированными сущностями семантики языка программирования. Названные агрегированные сущности позволяют работать программисту с большими проектами, но не снижают напряжённости в вопросе несоответствия объёма часов подготовки в рамках образовательных программ и выдвигаемых требований к выпускникам.

Одним из возможных вариантов повышения эффективности образовательного процесса обучающихся программированию и, в каком-то смысле, уплотнения подачи учебного материала может служить подход вовлечения студентов в рамках лекционных занятий в активную деятельность по выработке умений и навыков. Это не соответствует классическому восприятию лекции как аудиторного занятия, посвящённого исключительно обсуждению понятий, концепций и теоретических оснований изучаемой науки. Тут следует отметить, что программирование основано на предметных и практических умениях и навыках, довольно конкретных по своей реализации. Обсуждение же той или иной парадигмы программирования во время лекции даёт довольно поверхностное представление о способах программной реализации соответствующих алгоритмов. А, если ещё будет временной разрыв в неделю или месяц между текущим лекционным объяснением и применением полученных знаний на практике, то у студента разрушаются нити размышления и присвоенные ранее базовые понятия нужно будет вновь формировать.

Между тем, совмещение обсуждения теоретических концепций с исследованием и испытанием соответствующей программной реализации позволяет в большей мере осознать сущность изучаемого вопроса, принять и перевести её в

долговременную память. Здесь речь идёт не о предъявлении статической программы, выведенной на слайде презентации, а именно о собственноручно исправленном и запущенном на исполнение программном коде.

Современные облачные технологии позволяют не только получать доступ к просмотру удалённо хранимой информации вне зависимости от операционной системы локального устройства и установленного программного обеспечения на нём, но и модифицировать данные, работать с ними в интерактивном режиме, запускать на исполнение программный код. В настоящее время функционирует множество интернет-ресурсов, которые позволяют разрабатывать, хранить и исполнять программный код на разных языках программирования (<https://www.jdoodle.com/>, <https://replit.com/>, <https://jsfiddle.net/>, <https://sqliteonline.com/>, <https://codesandbox.io/>, <https://app.code2flow.com/>), но в рамках образовательного пространства лекции этого недостаточно. Хорошим решением будет наличие возможности совмещать в одном документе разные формы контента – текст, графики функций, рисунки, видеофрагменты и собственно программный код. Довольно близко к такой систематизации учебного материала подошли ресурсы подобные <https://stepik.org/>, однако функционал такого рода систем распределён на отдельных самостоятельных страницах и предназначен, скорее, для самостоятельного изучения материала с последующим прохождением контроля обученности.

Некоторое время назад стали появляться проекты, совмещающие контент разного формата и исполняемый программный код в рамках одного документа. Среди научных работников, специалистов по статистике, Big Data и Machine Learning набрала популярность среда разработки Jupyter Notebook - это веб-приложение, позволяющее создавать документы с текстом, рисунками, уравнениями, визуализациями и исполняемым программным кодом. С таким редактором можно работать локально (требуется установка) и делиться разработанными интерактивными документами как обычными файлами, но в рамках образовательного процесса удобнее пользоваться именно онлайн сервисами (<https://cloud.yandex.ru/>, <https://jupyter.org/try>, <https://colab.research.google.com/>).

Ориентируясь на количество пользовательских функций, необходимых для комфортной организации образовательной среды, следует особо выделить сервис от корпорации Google. Обозначим в данной работе некоторые, наиболее значимые возможности данного сервиса:

- большое количество предустановленных библиотек и простота их подключения,
- наличие возможности интеграции сторонних библиотек,
- наличие авторизации на сервисе с сохранением разработок,
- хранилище файлов любых форматов и возможность доступа к файлам, хранимых в облачном хранилище, непосредственно из разрабатываемой программы,
- возможность выбора процессора для исполнения кода (обычный, графический, тензорный),

- возможность разграничения уровней доступа к документу для других пользователей,
- возможность формировать содержание документа для удобства навигации,
- интеграция с облачным хранилищем и с системой управления версиями программного обеспечения GitHub.

Воспользовавшись предоставляемым функционалом [2], преподаватель программирования может для каждой темы учебной дисциплины или отдельной лекции подготовить интерактивный блокнот (рис. 1) и предоставить доступ к нему по ссылке для студентов.

```

1 lst = ['A', 'B', 'C', 'D'] # список объектов
2 count = len(lst)
3
4 for num in range(2**count):
5     cur_subset = []
6     for i in range(count):
7         if (1 << i) & num:
8             cur_subset.append(lst[i])
9     print(cur_subset)

```

```

[]
['A']
['B']
['A', 'B']
['C']
['A', 'C']
['B', 'C']
['A', 'B', 'C']
['D']
['A', 'D']
['B', 'D']
['A', 'B', 'D']
['C', 'D']
['A', 'C', 'D']
['B', 'C', 'D']
['A', 'B', 'C', 'D']

```

Рис. 1. Интерфейс интерактивного блокнота сервиса Google Colab.

Непосредственно во время проведения лекции, как в аудитории, так и дистанционно, студенты получают возможность не просто знакомиться с конспектом лекции, но и самостоятельно и оперативно, прямо во время лекционного обсуждения, исследовать программный код, модифицировать его и запускать на исполнение, получать ошибки и исправлять их, оставлять комментарии и обращаться к преподавателю, если он предоставил такую возможность. При наличии личного аккаунта студент получает возможность сохранять все произведённые с кодом модификации к себе в облачное хранилище и, в последствии, после лекции продолжить проводить эксперименты и изучать предложенный преподавателем материал.

## РЕЗУЛЬТАТЫ

В течение 2019-2021 гг. автором были проведены исследования возможностей применения интерактивных блокнотов для организации образовательного пространства лекции с обучающимися разных уровней обученности (школьников, студентов бакалавриата и магистратуры) в рамках различных учебных дисциплин: «Информатика и программирование», «Алгоритмы и структуры данных», «Анализ и разработка алгоритмов».

По итогам педагогических наблюдений можно выделить и отметить ряд положительных моментов, проявляющихся при использовании описанного в данной работе подхода.

1. Оптимизация расходования времени. Наличие готового конспекта, предоставляемого студентам, высвобождает время на лекции для обсуждения наиболее сложных вопросов и дополнительного материала.

2. Систематизация учебных материалов. Готовый конспект является систематизирующим началом, справочником по текущей теме и остаётся со студентом на лекциях, лабораторных занятиях и при подготовке к экзамену.

3. Прочность научения. Использование интерактивной онлайн среды непосредственно в процессе обсуждения материала лектором инициирует эффект закрепления на практике, когда частично усвоенный материал апробируется непосредственно в рамках контекста программного кода. Ряд исследователей отмечают [3], что «мы научаемся тому, что мы делаем» и «студент не выучивает то, что было в книгах или на лекции. Он научается тому, что книга или лекция побуждает его делать». Когда принятие предмета научения происходит через многократный личный опыт применения на практике, в том числе непосредственно в рамках лекции, то приобретённые умения и навыки надёжнее усваиваются.

4. Мотивация. Для ряда студентов оказывается значимым позитивное подкрепление, получение поощрительного стимула в виде того, что он действительно понял и разобрался, что он не хуже других, что понимает о чём говорят в аудитории, что может сам реализовать аналогичное решение. Инсайт происходит в том момент, когда обучающийся ещё не принял знания, не до конца понял логику работы обсуждаемого программного кода, но предоставленное контекстное и интерактивное поле для экспериментов даёт возможность испытать программный код, вникнуть в суть его работы «на лету».

5. Эффективность процесса научения. Традиционный уклад образовательного процесса в вузе подразумевает значительный по времени разрыв между проведением лекций и лабораторных занятий. Очевидно, что частично осознанные и принятые студентом образовательные единицы учебного материала «выветриваются» к моменту их практического применения, ослабевает их значимость и понимание. Организованное закрепление на практике способов применения алгоритмов и логики работы операторов языка программирования сразу после их предъявления лектором даёт синергетический эффект. В этом случае следующий этап научения начинается с большего уровня освоения материала и потенциально приводит к лучшим результатам.

## **ВЫВОДЫ**

Проведённые исследования особенностей применения интерактивных блокнотов в рамках лекционных занятий подтвердили влияние на эффективность образовательного процесса. Однако эти эффекты могут быть вызваны новизной пред-

лагаемых условий проведения занятий, поэтому для уверенной интерпретации полученных результатов следует провести ряд дополнительных педагогических экспериментов по сбору статистических данных в контрольных и экспериментальных группах.

#### Литература

1. George A. Miller. The Magical Number Seven, Plus or Minus Two. // The Psychological Review, 1956, vol. 63, pp. 81—97. – Текст: электронный. – Режим доступа: <http://psychclassics.yorku.ca/Miller/> (дата обращения: 12.09.2021).
2. Что такое Google Colaboratory? – Текст: электронный. – Режим доступа: <https://colab.research.google.com/notebooks/intro.ipynb> (дата обращения: 12.09.2021).
3. Хегенхан Б., Олсон М. Теория научения. – 6-е изд. СПб.: 2004. – 474 с.

### **DIGITAL TRANSFORMATION OF THE EDUCATIONAL SPACE LECTURES FOR STUDENTS OF PROGRAMMING**

**Belyakov A. Yu.**

Perm State Agro-Technological University, Perm, Russia

Email: [belyakov@pgatu.ru](mailto:belyakov@pgatu.ru)

#### **ABSTRACT**

The process of software development becomes more complicated with each step. New programming paradigms are emerging, the number of separate and independent concepts, abstractions and design concepts is growing. The contradiction between the increasing requirements for the level of readiness of a university graduate and the possibilities of the educational process is becoming more and more relevant. This paper considers an approach to the organization of the educational space of a lecture for students of programming, based on the use of cloud technologies. The functional capabilities of existing Internet services are compared. The most significant function is highlighted-the combination of the lecturer's explanatory text with the program code in a single space, with the possibility of modifying it, launching it for execution and obtaining results directly within the document. Combining theoretical material with its practical application during the lecture makes it possible to increase the efficiency of the educational process.

*Keywords: cloud technologies, interactive lecture, computer program, strength of learning.*