

АЛГОРИТМИЧЕСКИЙ СТОРИТЕЛЛИНГ КАК РАЗВИТИЕ МЕТОДА АНИМАЦИИ АЛГОРИТМОВ

Осипов Валерий Николаевич¹, Добрынин Владимир Николаевич²

¹Аспирант;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: ovn.22@uni-dubna.ru.

²Кандидат технических наук, профессор;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: arbatsolo@yandex.ru.

В статье рассматривается современное состояние и ограничения метода анимации алгоритмов, применяемого в образовательных целях. Анализируются причины снижения интереса к этому методу и предлагается новое направление его развития – алгоритмический сторителлинг. Этот подход основан на интеграции принципов цифрового образовательного сторителлинга в структуру анимации алгоритма. Предлагается расширить рамки анимации алгоритмов, дополнив её кратким описанием задачи, интерпретацией входных и выходных данных, а также включением персонажей-рассказчиков, поясняющих логику работы алгоритма, его преимущества, недостатки и возможные области применения. Статья содержит примеры алгоритмических историй для простых и сложных алгоритмов. Показано, что предложенный метод может использоваться в широком спектре сценариев, где процессы могут быть представлены в виде алгоритма, например, для анимации технологического процесса. Алгоритмический сторителлинг способствует повышению выразительности визуализации, и позволяет активнее вовлекать студентов в осмысление и обсуждение алгоритмов.

Ключевые слова: алгоритм, анимация алгоритмов, цифровой сторителлинг, алгоритмический сторителлинг, алгоритмическая история.

Для цитирования:

Осипов В. Н., Добрынин В. Н. Алгоритмический сторителлинг как развитие метода анимации алгоритмов // Системный анализ в науке и образовании: сетевое научное издание. 2025. № 2. С. 83-90. EDN: SXHRFB. URL: <https://sanse.ru/index.php/sanse/article/view/672>.

ALGORITHMIC STORYTELLING AS A DEVELOPMENT OF THE ALGORITHM ANIMATION METHOD

Osipov Valery N.¹, Dobrynin Vladimir N.²

¹PhD student;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: ovn.22@uni-dubna.ru.

²PhD in Engineering sciences, professor;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: arbatsolo@yandex.ru.

The article examines the current state and limitations of the algorithm animation method used for educational purposes. It analyzes the reasons for the decline in interest in this method and proposes a new



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

direction for its development – algorithmic storytelling. This approach is based on the integration of digital educational storytelling principles into the structure of algorithm animation. It is proposed to expand the scope of algorithm animation by supplementing it with a brief description of the task, interpretation of input and output data, and the inclusion of storytelling characters explaining the logic of the algorithm, its advantages, disadvantages, and possible areas of application. The article contains examples of algorithmic stories for simple and complex algorithms. It is shown that the proposed method can be used in a wide range of scenarios where processes can be presented as an algorithm, for example, for animation of a technological process. Algorithmic storytelling helps to increase the expressiveness of visualization and allows for more active involvement of students in understanding and discussing algorithms.

Keywords: algorithm, algorithm animation, digital storytelling, algorithmic storytelling, algorithmic story.

For citation:

Osipov V. N., Dobrynin V. N. Algorithmic storytelling as a development of the algorithm animation method. *System analysis in science and education*, 2025;(2):83-90 (in Russ). EDN: SXHRFB. Available from: <https://sanse.ru/index.php/sanse/article/view/672>.

Введение

Анимация алгоритмов – это метод представления алгоритмов в виде динамических иллюстраций, которая сосредоточена на демонстрации изменений переменных в ходе исполнения алгоритма. В прошлом этот метод нашел широкое применение в обучении студентов алгоритмизации и программированию. В последние годы эта тема поднимается редко, а большинство из разработанных ранее решений, реализующих этот метод, больше не используются.

Учитывая существование развивающихся решений, используемых в обучении алгоритмам, а также существующий тренд на цифровизацию образования, можно заключить, что анимация алгоритмов все еще является перспективным методом в обучении алгоритмизации. Причину снижения его популярности следует искать в чрезмерной узконаправленности и нерешенных проблемах самого метода.

В современном образовании все чаще можно услышать о сторителлинге – методе обучения, основанном на создании запоминающихся историй. Этот подход зарекомендовал себя как эффективный образовательный инструмент [1]. Особенный интерес в рамках рассматриваемой темы представляет цифровой сторителлинг – метод рассказывания истории с применением технологий мультимедиа [1]. Типология цифрового образовательного сторителлинга, помимо прочих видов, включает визуальный и аналитический сторителлинг [1]. Это означает, что приемы сторителлинга могут быть применимы и для анимации алгоритмов.

Целью работы является анализ недостатком метода анимации алгоритмов и разработка метода алгоритмического сторителлинга, направленного на улучшение метода анимации алгоритмов за счёт включения элементов образовательного сторителлинга. Предлагаемый метод не заменяет существующие визуализации, а дополняет их сюжетной структурой и персонажами. Такое решение позволяет повысить выразительность анимации и адаптировать её к различным образовательным сценариям и уровням подготовки.

Актуальность работы состоит в том, что сторителлинг представляет собой перспективный способ обновить метод анимации алгоритмов, расширив его выразительные возможности и область применения. Это позволяет придать методу новое развитие и вернуть к нему интерес в образовательной практике.

1. Анализ метода анимации алгоритмов

Метод анимации алгоритмов существует довольно давно. Вероятно, первая реализации концепции в виде информационной системы – это проект *Balsa* Брауновского университета, США, разработанный в 1980-х гг. [2].

В основном, подход был популярен в 1990-х и 2000-х гг. В частности, в этой статье несколько раз даются ссылки на опыт разработчиков систем *Jeliot* и *JHAVÉ*, которые разрабатывались в 2000-е гг. Большую коллекцию визуализаций алгоритмов собрали *S. Halim* и др. при сборе списка алгоритмов, доступных в виде визуализации на просторах Интернета [3].

Современной реализацией подхода можно назвать веб-сайт *VisuAlgo*, который является продолжением работы, описанной в статье *S. Halim* и др. в работе авторы в основном уделяли внимание анимации алгоритмов, которые редко встречаются, или вовсе не встречаются в таком виде в Интернете, а также на обеспечении доступности анимаций на различных устройствах.

В этом разделе рассмотрены некоторые особенности метода и его реализаций. Акцент в большей степени сделан на недостатках самого метода и проблемах, с которыми приходится сталкиваться разработчикам при его практической реализации.

1.1. Область применения метода

Современный метод анимации алгоритмов является очень узкоспециализированным. Метод применяется только для обучения студентов алгоритмизации. Авторы первых реализаций метода рассматривали возможность его применения в других областях, а не только в образовании. В частности, *M.H. Brown* и *R. Sedgewick* отмечали, что система *Balsa* может быть использована как инструмент в системном программировании [2].

Как пишет *T.L. Naps*, анимация алгоритмов прошла путь, который часто ждет новые популярные технологии: от резкого роста интереса до разочарования и осознания, что технология имеет как преимущества, так и недостатки [4]. Если раньше разработчики систем для анимации алгоритмов искали ее применения и в разработке, и в отладке программ, то сейчас рассматриваются только вопросы обучения студентов алгоритмизации.

Например, современное решение, веб-сайт *VisuAlgo*, используется именно в таком ключе. *S. Halim* и др. описали в статье разработку и применение визуализаций алгоритмов, которые затем привели к созданию сайта *VisuAlgo*. Из статьи можно заключить, что разработчики уделили основное внимание разработке анимаций для новых алгоритмов и вопросам доступности анимаций на большем количестве устройств, но не искали новых способов применения метода [3].

Сама реализация анимации алгоритмов, хоть и проблематична, но выполнялась уже множество раз. Однако гораздо сложнее добиться от нее нужного эффекта в образовании. В частности, необходимо добиться вовлеченности студентов в визуализацию [4]. Узконаправленность метода может сыграть дурную шутку: хотя метод действительно имеет свои преимущества, есть множество других направлений, требующих приложить схожее количество усилий, но способных принести гораздо больше пользы. Поэтому при разработке нового метода следует подумать и о новых способах его применения.

1.2. Общие проблемы метода

Некоторые проблемы метода можно обнаружить практически невооруженным взглядом, основываясь лишь на общих представлениях о методе.

Во-первых, по анимации алгоритма сложно судить, какую задачу он решает. Анимация алгоритмов позволяет показать отдельные шаги алгоритма в порядке их выполнения, но не дает обобщения. Понять, какую задачу решает алгоритм, можно только после получения первых результатов его работы. Распространенный способ решения этой проблемы – дать краткую справку об алгоритме перед началом его анимации.

Во-вторых, в анимации алгоритмов сложно выделить моменты, в которых принимаются решения. Говоря другими словами, в анимации сложно увидеть ветвления и циклы. Обусловлено это тем, что анимация алгоритмов нацелена на отображение изменений переменных. Но процесс принятия решения о передаче управления в одну из ветвей алгоритма не изменяет ни одну из переменных.

Отчасти, эта проблема решается визуальным выделением действия, которое предшествует ветвлению. Например, если выполняется сравнение двух значений, они могут выделяться цветом. Однако, ветвление не обязано следовать за сравнением элементов: сравнение может быть частью

сложной логической функции. Более того, даже если проверка условия будет, она не обязательно соответствует конструкции ветвления, а может быть частью циклической конструкции. В этом случае выбор происходит не между альтернативными вариантами, а между продолжением или прерыванием цикла. Тем не менее, точно указать момент принятия решения и результат можно только с использованием программного кода или псевдокода.

В-третьих, при анимации алгоритмов не дается интерпретация ни входных, ни выходных данных. Метод анимации алгоритмов сосредоточен на исполнении самого алгоритма, но не охватывает этапы до и после его исполнения.

1.3. Проблемы при практической реализации метода

Создание анимация алгоритмов с помощью современных технологий графического интерфейса – не такая уж и сложная задача. Основные трудности начинаются, когда возникает необходимость применить анимацию в образовательных целях. Применение ее в обучении студентов накладывает новые, зачастую противоречивые, требования.

Первая проблема – необходимость соблюдения баланса между полнотой анимации алгоритма и краткостью изложения. Если изображать алгоритм с доскональной точностью, то анимация может быть перегруженной несущественными деталями, или быть слишком длинной [5]. С другой стороны, если удалить слишком много деталей, представленной информации может оказаться недостаточно для понимания алгоритма. Изучая способы разрешения этого противоречия, разработчики *Jeliot* пришли к выводу, что следует изучить возможность реализации контекстного масштабирования.

Еще одна проблема – поиск баланса между простотой создания визуализации и ее выразительностью. Например, разработчики системы *Jeliot* выбрали вариант, при котором программа, заданная пользователем, анимируется автоматически: от пользователя не нужно никаких действий, кроме ввода программы на языке Java. Однако при этом они столкнулись со сложностями при добавлении новых образов [5]. С другой стороны, создание анимации без вспомогательных средств слишком трудоемко, хоть и позволяет добиться наиболее гибкой настройки образов под каждый алгоритм.

Следующая проблема заключается в неоднозначности образов. С одной стороны, необходимо обеспечить различимость образов, чтобы зритель мог точно распознать каждое отдельное действие алгоритма. С другой стороны, образы должны быть узнаваемыми для зрителя. Некоторые понятия, такие как структуры данных, уже имеют устоявшиеся обозначения. Создавать новые ради уникальности каждого обозначения бессмысленно, так как они не будут понятны для зрителя.

Следует также отметить вопросы подбора исходных данных. С одной стороны, исходные данные должны наилучшим образом отражать как сильные, так и слабые стороны алгоритмов. Для этого необходимо подбирать несколько наборов исходных данных. С другой стороны, исходные данные не должны быть лишь набором случайных чисел. Необходимо отмечать принцип, по которому эти исходные данные выбраны, иначе это может усложнить понимание сильных и слабых сторон алгоритма.

2. Применение сторителлинга

Основная идея работы – применить технику сторителлинга для улучшения метода анимации алгоритмов. Однако прежде чем применять сторителлинг, необходимо доработать анимацию алгоритма так, чтобы она могла включить все его обязательные элементы.

В следующих разделах предложены доработки анимации алгоритмов и предложен метод алгоритмического сторителлинга. Также рассмотрены примеры алгоритмических историй, которые могут быть созданы с использованием предложенного метода.

2.1. Расширение границ анимации алгоритма

Чаще всего, анимация алгоритма начинается уже на подготовленной площадке. К моменту начала все необходимые графические представления уже расставлены на экране. Завершается же анимация в тот момент, когда выполняется последняя операция перед возвратом из функции.

В процесс анимации алгоритма можно вполне органично встроить описание как задачи, решаемой алгоритмом, так и его входных данных. Не обязательно делать это с помощью сложной графики – достаточно просто предварить анимацию текстовой справкой и постепенным появлением графических представлений с комментариями, раскрывающими их назначение. Аналогично можно поступить и с выходом из анимации. Достаточно добавить демонстрацию выходных данных с текстовой интерпретацией результата.

Даже при такой простой реализации, анимация алгоритмов становится более информативной: становится ясным, что анимация относится к определенному алгоритму, есть исходные данные и результат. В случае же, если зрителю эта информация не нужна, перейти к началу анимации тоже не составит сложностей.

Эти улучшения нацелены на представление анимации алгоритма в виде трех этапов:

1. Введение. В этой части рассматривается задача, решаемая алгоритмом, а также переменные, участвующие в решении этой задачи.
2. Основная часть. В этой части воспроизводится сама анимация алгоритма.
3. Заключение. В этой части подводится итог: рассматривается результат, полученный в ходе исполнения алгоритма и дается интерпретация этого результата.

Неотъемлемой составляющей истории являются персонажи. В статье Ермолаевой и др. описываются 5 типовых сценариев, которые могут использоваться в педагогическом сторителлинге [6]. Все эти сценарии содержат как минимум 1-2 персонажей.

В алгоритме нет переменных, которые могли бы быть соотнесены с персонажами. Переменные раскрывают контекст, степень решенности задачи, промежуточные результаты и т.д. Персонажа же следует воспринимать как дополнительную сущность, как рассказчика, который объясняет действия алгоритма.

2.2. Понятие алгоритмического сторителлинга

После внесения доработок, анимация алгоритма может быть использована как основа для рассказа историй. Истории, созданные для описания алгоритмов, будем называть алгоритмическими историями.

Алгоритмическая история – это история, созданная в соответствии с принципами образовательного сторителлинга, основу которой составляет решение некоторой задачи с помощью заданного алгоритма. Объяснение алгоритмов при помощи таких историй будем называть алгоритмическим сторителлингом.

Алгоритмическая история может быть построена как по принципам классического сторителлинга, где зритель выступает только в роли наблюдателя, так и по принципам активного сторителлинга, где зритель активно участвует в обсуждении истории.

Алгоритмический сторителлинг можно рассматривать как подвид цифрового сторителлинга, так как при его реализации будут использованы технологии мультимедиа. Отличительная особенность алгоритмического сторителлинга заключается в том, что история в этом случае является алгоритмом. История должна обладать свойствами конечности и универсальности: она должна иметь логичное завершение при любых исходных данных. Более того, история должна обладать свойством детерминированности: при одних и тех же исходных данных, история имеет один и тот же результат.

Таким образом, алгоритмическая история описывает не один возможный сценарий, а спектр возможных исходов в рамках некоторой ситуации. Как разрешится ситуация – зависит от исходных данных алгоритма.

2.3. Элементарный пример алгоритмической истории

Для демонстрации возможностей подхода разработано несколько примеров алгоритмических историй. Первый пример основан на элементарных алгоритмах: алгоритмах линейного и бинарного поиска. История будет построена на основе сюжета «День сурка», описанного в статье Ж.Е. Ермолаевой и др. [6].

В истории главный герой – атлет – попадает в незнакомый спортзал. Для упражнений ему необходимо подобрать гантели подходящего веса. Но вот проблема – для подписи веса используются неизвестные единицы, и он не знает, как их перевести в килограммы.

Он обращается за помощью к администратору или тренеру, но тот лишь грубо отмахивается, и говорит: «Вон стойка с гантелями. Они выставлены по порядку от легких к тяжелым. Выбирай сам, не трать мое время!» Теперь герой вынужден сам перебирать гантели.

Видя стойку с несколькими десятками гантелей, атлет уже представляет, как долго он будет перебирать их по порядку. В этот момент демонстрируется сцена, как он долго перебирает каждую из гантелей, до тех пор, пока спортзал не закрылся. В этот момент происходит анимация алгоритма линейного поиска.

Но тут на помощь приходит друг или другой атлет и объясняет, что можно использовать алгоритм бинарного поиска. Вооружившись новым алгоритмом, атлет начинает подбор гантелей. Эта часть уже относится к анимации бинарного поиска. Атлет выполняет бинарный поиск, проговаривая шаги про себя, и быстро добивается успеха.

Эта история помогает описать алгоритм более полно, чем простая анимация. Сначала герой знакомит зрителей с задачей – выбор гантелей для тренировки. Затем, при встрече с грубым тренером/администратором, описываются исходные данные и их свойства: массив чисел (в виде стойки с гантелями разных весов), который отсортирован по возрастанию. Следующая часть истории – воображение героя, в котором мы видим недостаток линейного поиска – он слишком медленный. И в конце предлагается эффективный способ решения задачи – бинарный поиск, с помощью которого герой добивается результата. По ходу исполнения алгоритма герой проговаривает шаги про себя. Он это делает для себя, чтобы не ничего забыть и сделать все правильно, но в то же время он своими рассуждениями помогает зрителю понять алгоритм.

2.4. Алгоритмическая история для сложных сценариев

Метод можно применить не только для объяснения элементарных алгоритмов. Более сложный пример – анимация симплекс-метода для решения задачи линейного программирования (ЗЛП).

В этом случае за основу берется некоторая задача, которую можно формализовать как ЗЛП. Например, задача о составлении расписания обедов. Задача коротко формулируется так: даны несколько отделов, в каждом из которых работают несколько сотрудников, а также несколько временных промежутков, в которые сотрудники могут пойти на обед. Для каждого отдела указано, сколько из сотрудников должны быть на рабочем месте, а для временных промежутков – ценность обеда в это время. Необходимо составить расписание, которое бы максимизировало общую ценность обедов.

При решении этой задачи главный герой может быть студентом, который решает задачу. Первый этап повествования – объяснить, в чем состоит задача. Дальше – формализовать задачу, что герой будет делать постепенно, объясняя свои решения.

Следующий этап – собственно, решение задачи с помощью симплекс-метода. Решение задачи в общем случае делится на 3 этапа: приведение ЗЛП в каноническую форму, оптимизация с помощью симплекс-таблиц и, опционально, поиск целочисленного решения. На протяжении всех этапов герой объясняет, зачем этот этап нужен, а также сопровождает анимацию своими комментариями, поясняющими суть происходящего.

В самом конце мы получим готовое оптимальное расписание, которое герой рассказа с гордостью предъявит зрителю и всем сотрудникам компании.

Другой пример можно назвать «Экскурсией по металлургическому заводу». В этом случае, за основу истории взят не алгоритм в привычном понимании, а технологический процесс. Такую анимацию можно использовать, чтобы познакомить зрителей с технологиями, используемыми при производстве металлических изделий. Анимация может охватывать этапы от добычи, измельчения и обогащения руды, до получения готовой продукции.

В зависимости от характеристик исходного материала (какая использована руда, какие в ней примеси и т.д.) и желаемых характеристик конечного продукта, может изменяться как количество этапов обработки, так и технологии, используемые при их реализации.

В такой истории герой может выступать в роли экскурсовода, который знакомит зрителя с этапами обработки руды, а также обосновывает выбор конкретных технологий.

При рассказе такой истории была бы очень полезна возможность менять уровень детализации анимации. Таким образом, зритель получил бы возможность перемещаться от обзора процесса обработки руды к изучению особенностей конкретной технологии. Такая анимация была бы полезна и как научно-популярный рассказ, и как наглядное пособие для обучения специалистов.

3. Преимущества подхода

В отличие от существующих систем, реализующих анимацию алгоритмов, алгоритмический сторителлинг обладает важным преимуществом: он позволяет выстроить информацию об алгоритме в единое, связанное анимированное повествование.

Существующие решения используют несколько слабо связанных друг с другом способов описания алгоритма. Задача алгоритма, его сильные и слабые стороны описываются в текстовой справке, следить за логикой алгоритма приходится, наблюдая за псевдокодом, шаги алгоритма объясняются пошаговыми комментариями к коду, а изменения значений переменных можно увидеть, следя за анимацией.

Алгоритмическая история же – это единый, упорядоченный рассказ. Рассказ начинается с описания задачи и исходных данных. Затем постепенно он переходит к анимации самого алгоритма, по ходу которой герой истории показывает свои действия и объясняет их. Опционально, в анимацию может быть включено объяснение преимуществ и недостатков алгоритма, как описано в примере, или применения алгоритма.

Более того, алгоритмическая история может быть построена из нескольких алгоритмов, что помогает в их сравнении. Существующие решения предлагают лишь тематическую группировку алгоритмов. Современная веб-платформа *VisuAlgo*, например, предлагает формат электронной лекции, которая в рамках одного повествования охватывает несколько алгоритмов. Тем не менее, эти лекции лишь группируют анимации алгоритмов по теме, но не предлагают сравнительной анимации. Алгоритмические истории же могут быть использованы в обоих форматах.

Не менее важно, что новый метод может быть использован не только для объяснения алгоритмов, но и в целом для объяснения историй, которые можно представить в виде алгоритма. Это открывает возможности по применению метода не только для обучения студентов алгоритмам, но и в других направлениях в образовании, например, в дополнительном профессиональном образовании.

Как уже отмечалось, в алгоритмической истории персонаж проговаривает свои действия. Это решение позволяет включить объяснения логики работы алгоритма прямо в анимацию, что позволяет не только обозначить моменты принятия решений, но и обосновать сами решения.

Проговаривание, именование образов персонажами позволяет решить проблему различимости образов. Хотя некоторые понятия могут обозначаться одним и тем же образом, персонаж может назвать предмет, с которым взаимодействует, или кратко описать его особенности.

Использование историй, основанных на решении практических задач, позволяет найти прототипы для образов среди предметов, с которыми зрители сталкиваются едва ли не каждый день. Это в значительной мере решает проблемы с узнаваемостью образа. Вне зависимости от того, поймет ли зритель по изображению, что за предмет изображен, герой анимации даст ему наименование. Это позволит снизить риск возникновения ситуации, при которой зритель не узнает предмет по его изображению.

Если алгоритмическая история будет построена по принципам активного сторителлинга, можно будет добиться активного обсуждения алгоритмов на занятиях. Таким образом, использование алгоритмического сторителлинга может решить проблемы с вовлеченностью студентов в визуализацию.

Заключение

Алгоритмический сторителлинг является очень перспективным направлением развития метода анимации алгоритмов. Этот метод позволяет устранить большинство проблем анимации алгоритмов, и получает возможность развития в иных направлениях в образовании, помимо обучения алгоритмам.

Однако предложенный метод смещает баланс в визуализации алгоритмов к очень точному и выразительному их изображению. Это значит, что могут возникнуть проблемы, связанные со сложностью разработки новых анимаций и их излишней детализацией. При программной реализации метода следует обратить особое внимание на обеспечение простоты в создании новых анимаций и на способах борьбы с излишней детализацией. В частности, отмечается полезность функции переключения между несколькими уровнями детализации.

Список источников

1. Токтарова В. И., Семенова Д. А. Цифровой образовательный сторителлинг: возможности и перспективы // Казанский педагогический журнал. – 2023. – №1. – С. 57–67.
2. Brown M. H., Sedgewick R. Techniques for Algorithm Animation // IEEE Software. – 1985. – Vol. 2, № 1. – P. 28-39.
3. Learning algorithms with unified and interactive web-based visualization / S. Halim, Z. C. Koh, V. B. H. Loh, F. Halim // Informatics in Education. – 2012. – Vol. 11, № 1. – P. 53–72.
4. Naps, T. L. JHAVÉ: Supporting Algorithm Visualization // IEEE Computer Graphics and Applications. – 2005. – Vol. 25, № 5. – P. 49–55.
5. A decade of research and development on program animation: The Jeliot experience / M. Ben-Ari, R. Bednarik, R. Ben-Bassat [et al.] // Journal of Visual Languages & Computing. – 2011. – Vol. 22, № 5. – P. 375–384.
6. Сторителлинг как педагогическая техника передачи явного и неявного знания в вузе / Ж. Е. Ермолаева, О. В. Лапухова, И. Н. Герасимова, В. А. Смирнова // Образовательные технологии. – 2017. – № 1. – С. 73-90.