

ОБРАЗОВАТЕЛЬНЫЕ ПРАКТИКИ ВЫЧИСЛИТЕЛЬНОЙ ПЕДАГОГИКИ

Патаракин Евгений Дмитриевич,

доктор педагогических наук, ведущий научный сотрудник МГПУ, Москва

В СТАТЬЕ ПРЕДСТАВЛЕНО ПОНЯТИЕ «ВЫЧИСЛИТЕЛЬНАЯ ПЕДАГОГИКА» И ОБЕСПЕЧИВАЮЩИЙ ЕЁ РАЗВИТИЕ ПЕДАГОГИЧЕСКИЙ ДИЗАЙН КАК СОЦИОТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ СРЕДСТВ И СЦЕНАРИЕВ ДЕЯТЕЛЬНОСТИ, НАПРАВЛЕННОЕ НА ОСВОЕНИЕ УЧАЩИМИСЯ УМЕНИЙ ВЫЧИСЛИТЕЛЬНОГО МЫШЛЕНИЯ, ВЫЧИСЛИТЕЛЬНОГО УЧАСТИЯ И ВЫЧИСЛИТЕЛЬНОЙ РЕФЛЕКСИИ. ВЗАИМОСВЯЗАННОЕ ФОРМИРОВАНИЕ ВЫЧИСЛИТЕЛЬНОГО МЫШЛЕНИЯ, ВЫЧИСЛИТЕЛЬНОГО УЧАСТИЯ И ВЫЧИСЛИТЕЛЬНОЙ РЕФЛЕКСИИ ПРЕДСТАВЛЕНО НА ПРИМЕРЕ ОРГАНИЗАЦИИ СОВМЕСТНОЙ СЕТЕВОЙ ДЕЯТЕЛЬНОСТИ В СРЕДЕ И СООБЩЕСТВЕ SCRATCH.

• совместная деятельность • вычислительная педагогика • вычислительное мышление
• вычислительное участие • учебная аналитика • сообщество • Scratch

Введение

В последние годы кроме распространённого понятия «цифровая педагогика» (digital) всё чаще используется понятие «вычислительная педагогика» (computational pedagogy) [1], где прилагательное «вычислительная» подчёркивает использование общих правил и общего языка, на котором говорят все участники деятельности в цифровой среде вне зависимости от того, являются ли они человеческими или программными агентами. Понятие «вычислительная педагогика» акцентирует внимание на том, что добавляют к учебной деятельности компьютерные программы. Использование компьютеров и компьютерных программ освобождает человека от необходимости выполнения массы рутинных действий и переносит его деятельность на тот уровень, где она ещё остаётся творческой. С созданием и развитием компьютерных сетей человечество перешло на новый уровень совместной деятельности. Совместная сетевая деятельность требует творческой личности нового уровня. Одно из условий участия в творчестве в цифровой среде — это способность и готовность взаимодействовать с людьми и программами.

В вычислительной педагогике могут быть выделены следующие направления:

- социотехническое проектирование средств и сценариев деятельности, направленное на освоение учащимися уме-

ний вычислительного мышления (computational thinking);

- социотехническое проектирование средств и сценариев деятельности, направленное на освоение учащимися умений вычислительного участия (computational participation);
- социотехническое проектирование средств и сценариев деятельности, направленное на освоение учащимися умений вычислительной рефлексии.

Эти направления во многом соответствуют типологии умений, предложенной А.Л. Журавлёвым в рамках динамической концепции совместной деятельности [2]. Отправной точкой анализа совместной деятельности и её коллективного субъекта является взаимодействие участников совместной деятельности. В основании динамической концепции совместной деятельности находится концептуальный «треугольник», объединяющий три направления:

- предметно-направленное взаимодействие (взаимодействие, направленное на изменение предмета совместной деятельности);
- субъектно-направленное (взаимодействие, направленное на изменение характеристик индивидуального субъекта совместной деятельности);
- организационно-направленное (взаимодействие, изменяющее способы и стиль выполнения деятельности).

Среда вычислительной педагогики

Перечисленные направления развития вычислительной педагогики наиболее полно могут быть представлены на материалах языка и сообщества Scratch. Этот выбор основан на следующих положениях.

История и родословная языка: 50-летняя история, наследование философии конструкционизма и лучших особенностей среды Лого, множество успешных потомков — учебных языков визуального программирования. В середине 60-х годов XX века М. Минский, С. Пейперт совершили настоящий переворот в использовании компьютеров, показав, что ученик может контролировать деятельность компьютера и компьютерных агентов, выступать в роли дизайнера. Развитие этого направления было связано с разработкой интерактивного компьютерного дизайна для детей и опиралось в первую очередь на традицию инструментального и исследовательского обучения, обоснованного работами Дж. Дьюи и Дж. Брунера. С. Пейперт разработал подход, в рамках которого использовался обучаемый ребёнком компьютерный агент — Черепашка языка Лого. Это была первая работа по внедрению в педагогическую практику принципиально новых агентов-партнёров, при помощи которых люди могли думать и учиться. В дальнейшем это направление послужило основой для формирования направления, в рамках которого были созданы среды, позволяющие ученикам создавать и управлять многочисленными агентами. В книге «Переворот в сознании» Пейперт писал о необходимости «объекта, при помощи которого можно лучше думать». В поведении такого объекта ученик может увидеть своё собственное поведение. Впервые этот подход был воплощён в языке Лого, когда ученик получил в своё распоряжение агента — Черепашку, которой можно было давать простые команды: вперёд, назад, направо, налево, повтори и т.д. Из этих простых кирпичиков постепенно складываются значительно более сложные микромиры. Основной методологический принцип, который развивал С. Пейперт и его последователи, состоит в том, что: «Обучение происходит наиболее эффективно, если учащийся вовлечён в создание общественного объекта (*public entity*), будь то замок из песка, машина, книга или компьютерная программа» [3]. Потомок Лого, Scratch в свою очередь является предшественником

для множества современных сред визуального программирования, среди которых AppInventor, Blockly, Kodu, Pencil Code, PictoMir, Pocket Code и другие.

Массовость и продуктивность сообщества: многие среды вычислительной педагогики направлены на поддержку вычислительного участия. В 2018 году сообщество Scratch насчитывает более 26 миллионов участников и более 30 миллионов проектов. Значение, которое в Scratch уделяется совместной деятельности, возможностям для обмена цифровыми историями, является ответом на критические замечания М. Минского о том, что «литература начинается не с грамматики и не с правил, а с увлекательных историй, которые привлекают наше внимание и повествуют о вещах, которые имеют для нас значение» [4]. Эти слова в равной мере справедливы для мира историй, книг, театральных пьес, фильмов, музыкальных произведений, игровых партий и компьютерных программ. Учитывая эти критические замечания и опираясь на опыт сообщества Moose Crossing, создатели Scratch уделили внимание педагогическому дизайну сообщества, в котором ученики могли бы обмениваться увлекательными цифровыми историями [5]. Основные элементы дизайна сообщества Scratch использовались при проектировании других сообществ вычислительного участия — NetLogo, Pocket Code, StarLogo TNG.

Возможности освоения цифровой культуры, которые язык и сообщество Scratch открывают для международного и российского образования. Scratch — это объектно-ориентированная среда программирования, в которой блоки программ собираются из разноцветных кирпичиков и которая позволяет детям создавать собственные анимированные интерактивные истории, игры и модели. Среда Scratch позиционируется разработчиками и идеологами не столько как очередное средство для изучения информатики или основ программирования, но как среда для освоения современной цифровой культуры, создания современной детской литературы, в которой маленькие участники сообщества могут включаться в творчество, создавать собственные произведения (цифровые истории, игры, модели) на основе уже существующих образцов. Поскольку язык и сообщество

Scratch локализованы на множество языков добровольцами более чем из сорока стран, ученики включаются в деятельность на своём родном языке, но при этом могут осваивать опыт, накопленный всеми участниками международного сообщества.

Российский конкурс командных проектов в среде Scratch

В 2018 году эти возможности используются и исследуются в совместном проекте, включающем организацию в российских регионах серии образовательных Scratch-хакатонов и конкурс школьных команд по визуальному программированию на языке Scratch «Collab Challenge» — <http://scratch.mgpru.ru>. При разработке условий и материалов конкурса внимание уделялось поддержке участников не только как создателей конкурсных работ, но и как субъектов собственного образования, с тем чтобы учащиеся могли самостоятельно отслеживать формирование личных навыков вычислительного мышления, участия и рефлексии.

В рамках проекта были объединены Scratch-хакатоны и соревнования школьников Collaborative Challenge. Для проведения хакатонов и соревнований создают шаблоны — прототипы сценариев образовательной практики. Первый педагогический сценарий — это описание образовательной практики, которое включает последовательность действий. Описание каждого акта построено как ответ на вопрос «Как» (How to?):

- Как создать и использовать учительский аккаунт?
- Как зарегистрировать участников?
- Как создать студию для работы над общим проектом?
- Как пригласить в студию участников своей команды?
- Как создать новый спрайт для нового героя?
- Как научить героя перемещаться по экрану?
- Как посчитать, сколько шагов сделал персонаж?
- Как организовать взаимодействие персонажей?
- Как создать ремикс проекта?
- Как использовать дерево ремиксов?
- Как правильно оформить описание проекта?

- Как самостоятельно оценить качество кода?
- Как использовать Dr.Scratch?
- Как оценить совместность деятельности?

В этом сценарии уже заложен проект Scratch, который автор сценария образовательной практики выполняет от лица ученика (учеников) — это задача, которую он ставит перед учениками, от их лица пытается решить и понять, насколько правильно организован процесс деятельности. Было создано несколько таких исходных авторских сценариев учебной деятельности. Все они были опубликованы на сайте [Letopisi.org](http://letopisi.org) в категории ScratchHackathon. Описание сценариев и перечень ссылок на сценарии опубликованы в обобщающей статье <http://letopisi.org/index.php/Scratch/HowTo/Hackathon>.

Сценарий образовательной практики является не просто отвлечённым описанием деятельности, а всегда связан с конкретным объектом, которым является Scratch-проект.

На следующем этапе совместной методической работы исходные сценарии учебной деятельности обсуждались и дорабатывались командой организаторов хакатонов. В ходе совместной деятельности в сценарии вносились изменения. Некоторые строительные блоки, из которых выстраиваются образовательные практики, видоизменялись или заменялись на полностью новые строительные блоки. Например, в сценарии создания сказки о Колобке механизм взаимодействия персонажей в результате тестирования и обсуждения был полностью изменён на более простой и управляемый на основе анализа значения переменной, в которой хранилось число шагов главного героя. В результате в категории статей, объединяющей сценарии образовательных практик, появились новые блоки, которые можно было использовать для создания видоизменённых сценариев образовательной практики. Деятельность на этом этапе напоминает создание ремиксов Скретч-проектов. Кроме того, в связи с изменением сценария образовательной практики меняется и Scratch-проект, вокруг которого выстраивается сценарий образовательной деятельности. Все статьи ремиксы, включающие фрагменты кода и ссылки на Scratch-проекты, были объединены в составе единой статьи — сборнике

методических материалов по организации образовательных практик. Этот сборник доступен в сети и как вики-статья и как pdf-документ: <http://letopisi.org/index.php/Scratch/HowTo/Hackathon/Manual>

На третьем этапе учителя — организаторы школьных мастер-классов и хакатонов получили готовые методические материалы в виде документа и сборника вики-статей. При подготовке школьных и внешкольных мероприятий многие учителя проводили дополнительное тестирование предложенных сценариев учебной деятельности и создавали свои собственные ремиксы образовательных практик. В Москве и других городах России были организованы хакатоны, в ходе которых ученики осваивали новые образовательные практики, направленные на формирование умений совместной сетевой деятельности. Формат хакатонов был выбран как наиболее соответствующий поиску инновационных решений. Всего было проведено более 90 хакатонов, в которых приняло участие более четырёх тысяч школьников. В ходе хакатонов внимание уделялось умениям и инструментам совместной деятельности. В ходе тестирования учителя находят слабые и неясные места предложенной образовательной практики, вносят в шаблон изменения. Эта работа проходила большей частью на местах — в школах и центрах дополнительного образования. Наибольшее внимание в ходе мастер-классов и хакатонов уделялось использованию инструментов совместной деятельности.

К сожалению, большая часть ремиксов создавалась учителями либо в Google-документах, либо на страницах региональных вики, и нам удалось собрать небольшое число ремиксов образовательных практик. Однако было получено множество отчётов организаторов Scratch-хакатонов, включающих рассказы о совместной деятельности, фотографии участников и ссылки на студии и Scratch-проекты, сделанные на основе сценариев образовательных практик, подготовленных учителями.

Несколько фрагментов кода из проектов, созданных учителями для организации хакатонов, были перенесены из Scratch в вики в виде визуального кода и включены в общий сборник методических материалов

по организации образовательных практик.

Школьники в ходе хакатонов и мастер-классов переходят к собственной образовательной практике, которая наследует некоторые элементы педагогического сценария и является его настоящим тестированием.

Полностью завершить работу и реализовать всю последовательность образовательной практики удалось 339 командам участников. При этом учителя — организаторы совместной деятельности команд оценивали отношение числа команд, сумевших довести работу до конца и представить работу на конкурс, к общему числу участников примерно в 60 процентов. В основном потери были связаны со сложностями именно командной деятельности, с тем, что участникам было сложно договориться между собой, разделить роли и обязанности.

На финальном этапе жюри оценивало не только сам заключительный проект, но и материалы студии, дерева ремиксов, т.е. то, насколько деятельность участников команды соответствовала разработанному сценарию образовательной практики и формированию вычислительного мышления, участия и рефлексии.

Вычислительное мышление в среде Scratch

В современной цифровой среде, когда совместная деятельность направлена на создание совместных цифровых историй, для успешного участия в предметно-направленном взаимодействии субъекту совместной деятельности необходимо уметь обращаться со средствами ИКТ, создавать письменные сообщения, создавать графические и мультимедийные объекты. Необходимые для участия в предметно-направленных взаимодействиях умения относятся к сфере информационной компетентности субъекта. В современной зарубежной литературе большая часть информационных компетенций объединяется под зонтиком понятия «вычислительное мышление» (Computational thinking), которое включает способность к абстракции, декомпозиции, построению алгоритмов и распознаванию повторяющихся паттернов. Эти способности могут успешно формироваться в среде Scratch,

поскольку здесь представлены базовые концепции современного программирования. Не случайно Scratch и созданный на его основе Snap! успешно используются в университетском курсе «Красота и радость вычислений» [6].

Примеры реализации концепций вычислительного мышления в среде Scratch.

Контроль и управление. Каждый управляемый агент исполнитель в среде Scratch является отдельным объектом, который называется «спрайт». Спрайты можно создавать и клонировать. Начиная управлять спрайтами в среде Scratch, ученик, как правило, собирает последовательность команд и в длинную цепочку. Более изящное управление предполагает использование управляющих блоков *повторить* () и *всегда* []. Как показано на рис. 1, в обоих случаях на экране будет нарисован квадрат, но во втором решении использованы управляющие структуры.

Представление данных. Ученики начинают использовать данные, встраивая их в скрипты программы, управляющей действиями спрайтов. Если нужно пройти определённое количество шагов или сказать фразу, то это количество шагов или текст фразы просто сообщаются спрайту-исполнителю. Следующий уровень использования данных в среде Scratch связан с использованием численных переменных и списков (рис. 2).



Рис. 1. Переход к управляющим блокам Scratch

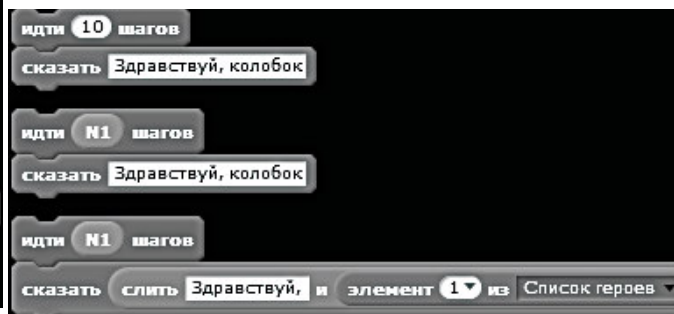


Рис. 2. Переменные и списки Scratch

Абстракция. Б. Харви — создатель Snap! и один из авторов курса «Красота и радость вычислений» в качестве примера абстракции проводит аналогию между машинами и компьютерными программами. Автомобили изготовлены из гаек, болтов, металлических стержней, больших металлических блоков, резиновых или бумажных прокладок, пластиковых контейнеров для жидкостей, заклёпок, проводов и т.д. Но если вы пытаетесь отремонтировать автомобиль, вы не будете думать в этих терминах; если бы вы начали так действовать, то вы бы никогда не сумели обнаружить проблему. Вместо этого вы думаете о двигателе, генераторе, топливных форсунках, тормозах, трансмиссии и т.д. Это абстракция. Прежде чем мы сможем критиковать технологии, мы должны понять, как они устроены, а абстракция — очень мощная организационная идея для описания механизма организации технологий [7]. Действия учеников в Scratch, так же как и в Лого, начинаются с использования уже готовых блоков-команд. В дальнейшем ученик учится объединять команды и создавать на их основе новые командные блоки, как это показано на рис. 3.

Ещё более высокий уровень абстракции связан с механизмом клонов, который позволяет создавать для отдельного спрайта его многочисленные копии (клоны). Например, мы можем создать каплю дождя или снежинку и затем, клонируя этот объект, вызвать на экране дождь или снегопад (рис. 4).

Интерактивное взаимодействие. Все агенты в Scratch реагируют на нажатие клавиш клавиатуры или на нажатие кнопок манипулятора мыши. Например, простейшая программа управления перемещением спрайта выглядит следующим образом (рис. 5).

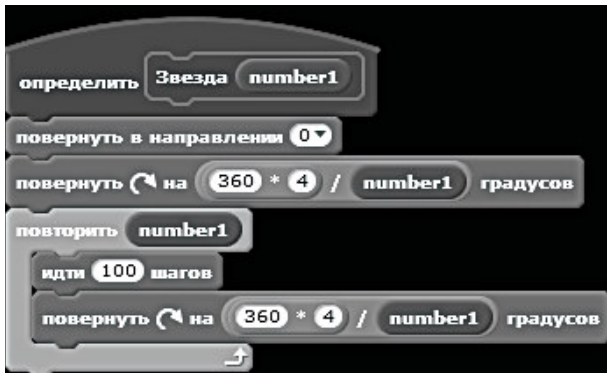


Рис. 3. Создание собственных блоков в Scratch

Синхронизация. Автор сценария управляет взаимодействием спрайтов при помощи команд «передать» [сообщение] и «когда я получу» [сообщение] (рис. 6).

Параллельное выполнение сценария. Любой спрайт в среде Scratch может выполнять параллельно несколько действий: двигаться, поворачиваться, изменять цвет и т.д. Например, мы можем собрать скрипт-программу, которая будет управлять перемещением спрайта, и другой скрипт, который будет управлять изменением внешнего вида нашего спрайта. В результате одновременного выполнения указанных скриптов спрайт одновременно меняет внешний вид и положение на экране (рис. 7). В результате параллельного выполнения

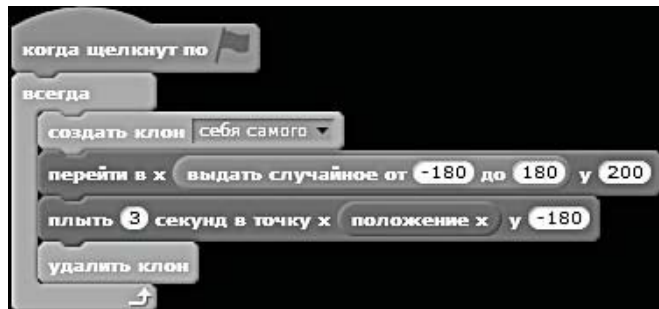


Рис. 4. Клонирование объекта

скриптом у зрителя создаётся впечатление, что спрайт «идёт по экрану».

Логика. Команды, связанные с логическими конструкциями, позволяют ученикам создавать динамические проекты, в которых поведение спрайтов-исполнителей зависят от ситуации. Логические конструкции очень важны при создании игр и моделей, где поведение исполнителя определяется данными из окружающей среды. Например, **если** спрайт «касается чёрного цвета», **то** сделать сколько-то шагов назад. Более сложные управляющие конструкции предполагают выбор **если, то... иначе**. Ещё более сложная логика управления может быть реализована благодаря сочетанию условий — «И», «Или», «Не» (рис. 8).

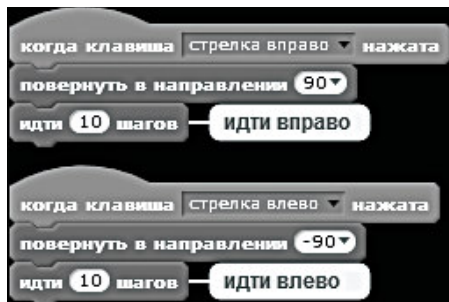


Рис. 5. Интерактивность в Scratch



Рис. 6. Передача сообщений

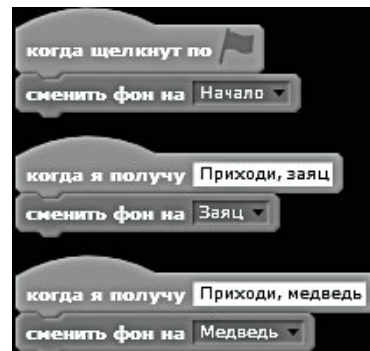


Рис. 7. Параллельное выполнение сценария

**Критериальное оценивание навыков вычислительного мышления
на основании анализа блоков визуального программирования**

Критерий	Высокий уровень	Средний уровень	Низкий уровень
Контроль	В программе используются условия. Например, повторять пока не (<?>)	Используются конструкции циклов. Например, повторить () и всегда []	Все команды программы собраны в одном блоке. Например, сказать [Да]
Представление данных	Для хранения данных используются списки. Например, (элемент (1 v) из [list v])	Для хранения данных используются переменные. Например, здать [X] значение (10)	Все значения определены в тексте программы. Например, идти (10) шагов
Абстракция	В программе создаются клоны объекта. Например, создать клон [себя самого]	Программа содержит блоки процедур, созданных автором. Например, определить РисуемДом	Программа содержит один длинный исполняемый скрипт. Например, идти (10) шагов поднять перо
Интерактивное взаимодействие	В программе используются управляющие блоки с условиями. Например, если < [громкость] > [39]>, то	В программе используются различные управляющие события. Например, когда клавиша [пробел] нажата	В программе используется только первый запускающий блок. Например, когда щёлкнут по зелёному флагу
Синхронизация	Синхронизация поведения спрайтов осуществляется через их реакцию на произошедшие изменения. Например, когда фон меняется на [Лес] или ждать до ((X) = (0))	Синхронизация поведения спрайтов осуществляется через обмен сообщениями. Например, передать [Беги] и реакция другого агента когда я получу [Беги]	Синхронизация поведения спрайтов осуществляется через вставку временных пауз. Например, ждать (1) секунд
Параллельность действий	Параллельные действия запускаются в ответ на сообщения. Например, когда я получу [сообщение1]	Параллельные действия запускаются в ответ на нажатия разных клавиш. Например, когда клавиша [пробел] нажата	Все действия стартуют по первому запускающему блоку. Например, когда щёлкнут по зелёному флагу
Логика	Выбор действий зависит от совокупности условий. Например, если <<касается цвета [красный]?> и <(X) = (0)>>, то	Действия ветвятся в зависимости от условий. Например, если <касается цвета [красный]?>, то думать [Старт] иначе думать [Финиш]	Действие определяется условием. Например, если <касается цвета [красный]?>, то ...

Все перечисленные концепции вычислительного мышления могут быть использованы в качестве критериев оценивания кода проектов в среде Scratch. При этом в зависимости от тех структур, которые использовались при программировании, можно выделить низкий, средний и высокий уровни формирования навыков вычислительного мышления (табл. 1).

Освоение умений вычислительного мышления в среде Scratch происходит в ходе деятельности, направленной на создание проекта — цифровой истории, игры или модели. Освоение умений является побочным результатом творческой деятельности. При этом партнёры по деятельности в ходе создания проекта могут обращаться к помощи компьютерных программ и веб-приложений.

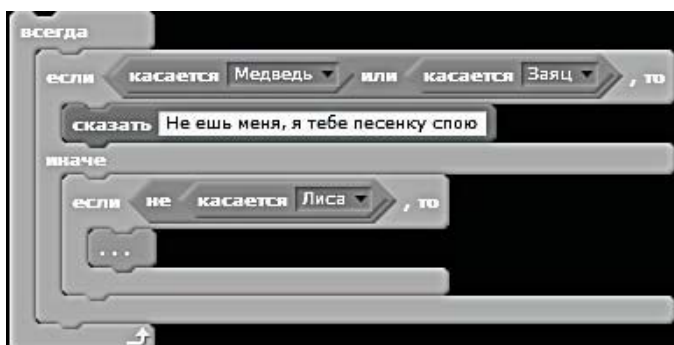


Рис. 8. Использование логических конструкций

Все проекты Scratch публикуются на сайте scratch.mit.edu. Приложение Dr.Scratch (<http://www.drscratch.org/>) позволяет создателям цифровых историй указать адрес проекта и самостоятельно оценить и сертифицировать проект на основании анализа блоков, которые были использованы при создании проекта [8]. Возможность для участников самостоятельно контролировать не только поведение спрайтов, но и собственные достижения, используя для этого сетевые приложения, повышает степень их автономности как субъектов образования.

Вычислительное участие

Субъектно-направленное взаимодействие приводит к изменению характеристик индивидуальных субъектов совместной деятельности. В исследованиях последних лет феномен социальной компетентности рассматривается как общее собирательное понятие, свидетельствующее об уровне социализации и включающее способность к сопереживанию, терпимость к чужому мнению, независимость, способность к коммуникации и творчеству, терпимость к конфликтным ситуациям, умение самостоятельно принимать решения [9]. Сложность освоения социальных компетенций субъектами образования связана с неопределённостью объектов той деятельности, в рамках которой может происходить формирование социальных компетенций. Появление в виртуальной среде «социальных объектов», вокруг которых возникает совместная деятельность и формируются связи, имеет огромное значение для образования. Социальные компетенции связаны с приобретением опыта использовать умения и результаты деятельности других людей и опытом предоставления собственных умений и ре-

сурсов в форме, когда они пригодны для использования другими людьми при создании совместных историй, включающих множество страниц [10]. В сообществе Scratch взаимодействие участников опосредовано цифровым рассказом, который в рамках совместной деятельности является фундаментальным обучающим объектом. Социальные компетенции, необходимые для успешного участия в субъектно-направленных взаимодействиях, формируются в ходе совместной сетевой деятельности по созданию и изменению цифровых рассказов.

Все создаваемые в ходе совместной деятельности продукты могут быть повторно использованы другими участниками совместной деятельности. Можно утверждать, что методологический принцип конструкционизма в среде Scratch получил дополнительное развитие и может быть переформулирован следующим образом: *Обучение происходит наиболее эффективно, если субъект образования вовлечён в создание продукта деятельности, который может обсуждаться, оцениваться и использоваться другими участниками для создания новых объектов.*

В сетевой среде для успешного участия в субъектно-направленных взаимодействиях субъекту совместной деятельности необходимо уметь оценивать и обсуждать объекты, созданные другими субъектами совместной деятельности; принимать оценки и суждения других участников совместной деятельности; отслеживать действия других субъектов; классифицировать предметы, созданные другими субъектами совместной деятельности. Все эти умения свидетельствуют о социальной компетентности субъекта совместной сетевой деятельности. Ж. Кафаи использует термин «вычислительное

участие» (Computational participation) [11], подчёркивая, что в сообществе Scratch «объекты для мышления» становятся «объектами для совместного использования». В результате того, что код становится социальным объектом, программирование из когнитивного умения вычислительного мышления превращается в социальное и культурное умение, необходимое для участия в совместной сетевой деятельности.

Во всех обучающихся сообществах эффективное участие предполагает, что учащиеся самостоятельно распределяют обязанности внутри команды, работающей над проектом. В настоящее время основными инструментами для организации совместной деятельности внутри сообщества Scratch являются:

- студия, в которой происходит сбор и хранение проектов и материалов к проектам;
- ремикс — возможность создать свою копию проекта, сделанного другим участником, и внести в эту копию дополнения и улучшения;
- рюкзак (backpack) — возможность копировать отдельные спрайты и скрипты из чужих проектов в отдельное пространство и использовать эти объекты в своём проекте (рис. 9).

Умения, связанные с вычислительным участием, и критерии оценивания этих умений гораздо менее разработаны, чем умения и критерии в сфере вычислительного мышления. Исходя из открытых данных, которые доступны для участников сообщества Scratch, мы можем предложить следующие критерии и характеризующие их показатели:

- Готовность делиться своим кодом — характеризуется числом проектов, которыми участник поделился на сайте, и числом ремиксов, которые были сделаны на основе этих проектов.



Рис. 9. Спрайты в рюкзаке

- Готовность использовать чужой код — характеризуется числом проектов, которые участник отметил как «избранные», и числом ремиксов, которые участник создал на базе чужих проектов.
- Расположенность к сотрудничеству — характеризуется числом тех членов сообщества, кто подписался на новости о действиях данного участника.
- Заинтересованность в сотрудничестве — характеризуется числом тех членов сообщества, на новости о действиях которых подписался данный участник.
- Готовность организовывать пространство совместной деятельности — характеризуется числом студий, в которых участник исполняет роль куратора или приглашённого менеджера.

Это неполный перечень показателей. Безусловно, большое значение имеет описание собственных умений и интересов в профайле, описание и комментирование студий, комментирование кода внутри спрайтов и т.д. Однако это качественные характеристики, и они в настоящее время не могут быть переданы от людей к компьютерным программам. Что же касается перечисленных в табл. 2 количественных показателей, то уже сегодня в исследовательских версиях Scratch существуют процедуры, которые позволяют извлекать облачные данные, хранимые на сервере Scratch, и использовать их в исследовательских проектах учащихся. Пример такого использования приведён на рис. 10.

Показательно, что в области учебной аналитики и использования больших данных происходит постепенная передача контроля от исследователей и педагогов к самим учащимся, которым предлагается роль исследователей [12]. Знакомство школьников и учителей с наукой о сетях может начинаться с исследования карт, которые отображают их собственную деятельность в учебных сообществах. Преимущество такого подхода заключается в том, что сетевой подход используется для понимания ситуаций, в которые вовлечены и школьники, и учителя. Таким образом, наука о сетях показывает свои возможности на близком для учеников и учителей материале,

и субъекты образования становятся исследователями своей собственной деятельности.

Необходимо отметить, что в настоящее время Scratch не является единственной средой, где осуществляются эксперименты по организации совместной деятельности, основанной на создании и совместном использовании объектов. Сравнение известных учебных сообществ, близких теории конструкционизма, позволяет увидеть, что практически все они используют идею цикла или спирали действий, которые совершают субъекты над объектами совместной деятельности.

В сети Globaloria учащиеся выполняют действия над компьютерными играми: *Играй -> Планируй -> Разрабатывай прототип -> Программируй -> Публикуй* [13]. Созданный объект может обсуждаться внутри специализированного вики-портала и в дальнейшем использоваться другими участниками при создании их собственных программ.

В сообществе NetLogo разработчики моделей совершают над ними следующую последовательность действий: *Создай -> Запусти -> Поделись -> Прокомментируй -> Видоизмени -> Создай новые версии* [14]. Автор модели может пригласить других

участников к совместному изменению модели. При этом участники могут выполнять различные роли в команде.

В сети StarLogo TNG существуют два взаимосвязанных цикла действий, которые разработчики совершают над моделью как объектом деятельности. Цикл исследования: *Наблюдай/Собирай данные -> Формулируй вопросы -> Проверь/Экспериментируй*. Цикл дизайна: *Проектируй -> Создавай -> Проверь/Экспериментируй -> Проектируй* [15]

В сети Looking Glass создатели трёхмерных моделей на языке Alice совершают следующие действия над моделями: *Создай -> Анимируй -> Выдели модуль -> Поделись* [16]. Платформа позволяет выделить в ремикс поведение отдельного объекта-персонажа и затем передать это поведение другому персонажу.

В сети CloudWorks учителя совершают действия над картами — сценариями учебного процесса (объект в среде CompendiumLD): *Найди -> Поделись -> Обсуди* [17]

В сети социальных исследований WebGrid: исследователи совершают следующие действия с репертуарными решётками: *Покажи -> Выдели кластеры -> Построй карту -> Сравни* [18]

Таблица 2

Критериальное оценивание структуры Scratch студии

Критерий	Высокий уровень	Средний уровень	Низкий уровень
Продуктивность	Граф содержит множество созданных участниками объектов	Число объектов примерно равно числу участников	Число объектов значительно меньше числа участников
Связанность	Все субъекты и объекты деятельности объединены в одном графе	Участники и объекты деятельности объединены в небольшое (3–4) число компонентов	Граф студии разбит на множество несвязанных компонентов
Сплочённость	На социограмме представлена одна сплочённая группа (клика), в которой все участники связаны друг с другом через страницы	Участники образуют несколько малочисленных групп	На социограмме практически нет групп, что говорит об отсутствии взаимодействия
Устойчивость	На социограмме представлены несколько ключевых игроков, связи которых обеспечивают устойчивость совместной деятельности	На социограмме представлены 2–3 участника, удаление которых приведёт к тому, что граф рассыплется на несвязанные компоненты	На социограмме есть единственный ключевой игрок, через которого идут все информационные процессы. Удаление этого узла разрушает сеть

Вычислительная рефлексия

Коллективная субъектность формируется и возрастает благодаря взаимодействиям, которые приводят к росту взаимосвязанности и взаимозависимости индивидов в группе, к росту способности группы проявлять совместные формы активности, выступать в качестве единого субъекта деятельности по отношению к другим группам, к росту способности группы к саморефлексии, в результате которой формируются чувства «Мы» и «образ-Мы» [19]. Исходя из этого, ключевым умением, необходимым коллективному субъекту совместной сетевой деятельности для участия в организационно-направленных взаимодействиях, является умение групповой саморефлексии и формирование образа «Мы». Под групповой рефлексивностью обычно понимается способность группы к рефлексии — анализу своей деятельности и отношений, которые складываются между участниками совместной деятельности. М. Вест отмечает, что наряду с рефлексивностью, направленной на решение задачи, существует социальная рефлексивность группы, то есть склонность её членов к анализу сложившихся между ними отношений [20]. В отечественной психологии рефлексия предстаёт как особый вид кооперации, механизм развития совместной деятельности, позволяющий её участникам

порождать единый, нормативный смысл ситуации. Исследования групповой рефлексивности свидетельствуют о том, что она выступает одним из важнейших факторов эффективности совместной деятельности. **Системные компетенции** связаны с приобретением опыта системного мышления, анализа сетевых ценностей и сетевых структур, способностью и готовностью участвовать в проектах совместной сетевой деятельности. Понимание организации, как правило, начинается с описания структуры, т.е. того, как разделён в организации труд, распределены власть и влияние. Это описание структуры формализуется в виде схемы, которая описывает желаемое состояние, но ничего не говорит о действительном статусе какого-либо человека, его взаимодействии с коллегами.

Основоположник психологической теории компетентности и автор книги «Компетентность в современном обществе» Дж. Равен в своих исследованиях уделяет первоочередное внимание системному мышлению и формированию системных компетенций, всё чаще обращаясь к идеям социо-кибернетики, которая представляет собой применение системной динамики и системно-динамических моделей к исследованию социальных процессов [21]. Ключевое значение в формировании компетенций имеет объект



Рис. 10. Определение показателей отдельного участника в среде Scratch

деятельности, который задаёт целеполагание, форму и сценарий деятельности. Определяя пути формирования системного мышления и системных компетенций, необходимо определить образовательный объект деятельности, в ходе которой у субъектов образования формируется системное мышление. Деятельность субъектов образования должна быть связана с изучением системного объекта, который помогает им понять функционирование системы совместной деятельности. Таким фундаментальным системным образовательным объектом могут служить статические или динамические модели систем различного рода — диаграммы, социограммы, системограммы, возможности для построения и получения которых в последние годы необычайно увеличились благодаря информационным технологиям.

Внутри сообщества Scratch средством, поддерживающим процесс групповой рефлексии, является инструмент дерева ремиксов. Для каждого проекта этот инструмент позволяет увидеть проекты, которые были созданы как его видеоизмененные копии. На рис. 11 представлено дерево ремиксов для проектов учебной студии «Путешествие в Сомовку».

Субъекты деятельности могут не только совместно изменять объекты — все эти изменения записываются в журнал изменений, а данные этого журнала могут использоваться для построения карты совместной сетевой

деятельности субъектов образования. Эта карта позволяет оценивать образовательную социотехническую систему по показателям, характеризующим совместную сетевую деятельность субъектов образования является сложной адаптивной системой, для понимания которой необходимы специальные инструменты. Такими инструментами, позволяющими получить карты и схемы связей между субъектами совместной сетевой деятельности, являются сетевые модели, основанные на данных о действиях субъектов и изменении объектов совместной деятельности.

Различные типы современных социотехнических систем, в которых реализуется совместная деятельность участников, сохраняют в журнале записи обо всех действиях участников. В общем виде история, сохранённая в журнале, может быть представлена как запись игровой партии, состоящей из множества ходов, каждый из которых содержит следующие три обязательных элемента:

Субъект деятельности / Объект деятельности / Вид деятельности

Различные формы совместной сетевой деятельности (совместное редактирование документа, создание и комментирование записей в блоге, создание и совместное редактирование статей в вики и т.п.) могут быть сведены к единой схеме, которая

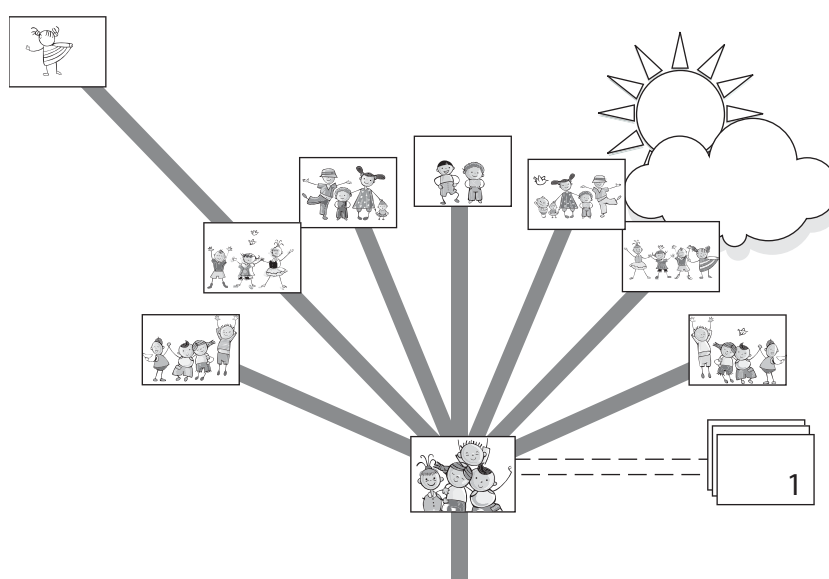


Рис. 11. Дерево ремиксов студии «Путешествие в Сомовку»

Уровни совместной деятельности для различных практик

Поля и типы деятельности	Игровое поле (футбол, баскетбол, волейбол)	Поле настольных игр (шахматы, шашки, го)	Поле сетевой деятельности (вики, GitHub, Alice, Scratch)
Индивидуальный (предметно-направленные взаимодействия)	Быстро бегать, попадать по мячу, обводить противников, индивидуальная техника владения мячом — объектом деятельности	Знать правила, дебюты, эндшпили, узнавать повторяющиеся игровые паттерны, просчитывать варианты	Вычислительное мышление — уметь кодировать, создавать игры, цифровые истории, модели
Коллективный (субъектно-направленные взаимодействия)	Отдавать мяч, принимать мяч, видеть партнёров, открываться	Уметь читать, записывать, обсуждать партии. Уметь заимствовать идеи из записей чужих партий. Уметь делать ремиксы	Вычислительное участие — уметь обсуждать, использовать проекты других участников, делать ремиксы, использовать рюкзак, делиться своими проектами, организовывать деятельность других участников
Системный (организационно-направленные взаимодействия)	Понимать сетевую динамику игры, анализировать игровые диаграммы и менять организацию игры на основе анализа этих диаграмм	Рассматривать свою школу, как коллективного субъекта деятельности	Вычислительная рефлексия — уметь анализировать данные о совместной деятельности — викиграммы, деревья ремиксов

позволяет анализировать и сравнивать деятельность участников. Перефразируя утверждение Латура, можно утверждать, что как только каждое редактирование страницы в вики, каждый ремикс и каждая модификация программы в Scratch, Alice, StarLogo Nova, NetLogo, каждое изменение, оценивание или обсуждение карты учебного дизайна в CloudWorks, каждое заполнение и видоизменение решётки репертуарных конструкторов в WebGrid начинает сопровождаться записью в электронном журнале, все субъекты образования получают возможность усовершенствовать процесс обучения. Возможность критериального оценивания совместной сетевой деятельности при помощи социограмм была реализована ранее для образовательной вики-системы Letopisi.org [22]. При разработке таблицы критериального оценивания совместной сетевой деятельности использовались следующие критерии: продуктивность, связанность, сплочённость и устойчивость.

Поскольку для всех систем продуктивной сетевой деятельности справедливы общие принципы, то мы предположили, что возможно использовать сходный аппарат и для построения социограмм на базе анализа

тех действий, которые совершают участники сообщества Scratch (скретчеры) в рамках студий, созданных для совместной работы. При этом мы полагаем, что студии Scratch по своим функциям аналогичны категориям MediaWiki. Критерии и показатели качества групповой деятельности в студии Scratch представлены в табл. 2.

В экспериментальных целях мы собрали данные о действиях участников в нескольких Scratch студиях и на основании этих данных получили социограммы совместной деятельности. Пример такой социограммы приведён на рис. 12. Фигурки обозначают участников-скретчеров, листы кода — проекты. Сплошная линия от скретчера к проекту означает, что данный скретчер является автором проекта. Пунктирная линия обозначает, что скретчер комментировал проект.

На социограмме мы видим, что данная студия характеризуется средней продуктивностью, высокой связанностью и сплочённостью и низкой устойчивостью. Удаление центрального скретчера и его проектов приведёт к тому, что граф студии рассыплется на множество несвязанных участников.

Заключение

Взаимосвязанное развитие направлений вычислительной педагогики может быть обобщено в одном предложении: «Обучение происходит наиболее эффективно, если субъект образования вовлечён в создание продукта деятельности, который может обсуждаться, оцениваться и использоваться другими участниками для создания новых объектов, а данные о взаимодействиях субъектов образования могут быть представлены в виде карты». Продуктом совместной сетевой деятельности является «цифровая история (рассказ, нарратив), которая может принимать различные формы (текст, презентация, видеоигра, анимация, театральная постановка, модель, сценарий будущего, нормативно-правовой акт и т.п.). Принципиальное значение имеет то, что цифровая история и составные элементы этой истории могут использоваться другими участниками совместной деятельности при создании новых историй.

Поскольку мы находимся в самом начале объединения вычислительного мышления, участия и рефлексии, то интерес представля-

ет сравнение различных ситуаций, в которых происходит формирование умений, необходимых для успешного функционирования в составе коллективного субъекта, которым может быть футбольная команда, школьная или корпоративная вики, школьная или домашняя скретч-студия, шахматная или какая-то другая школа единоборств. Примеры таких ситуаций приведены в табл. 3. Вычислительная техника добавляет дополнительные возможности и расширяет деятельность во всех ячейках таблицы независимо от того, осуществляется деятельность в цифровой среде или нет. Наиболее ярко возможности вычислительной техники по дополнению способностей группы к коллективной рефлексии проявляются сегодня при анализе футбольных матчей, когда игроки и тренеры действуют не только как мастера владения мячом или мастера тактических построений, но и как специалисты по системной динамике. В данной работе мы рассматриваем развитие вычислительной педагогики в среде и сообществе Scratch, поскольку внутри этого сообщества уже сейчас компьютерные программы усиливают и поддерживают индивидуальные и коллективные способности мышления, участия и рефлексии. Найденные

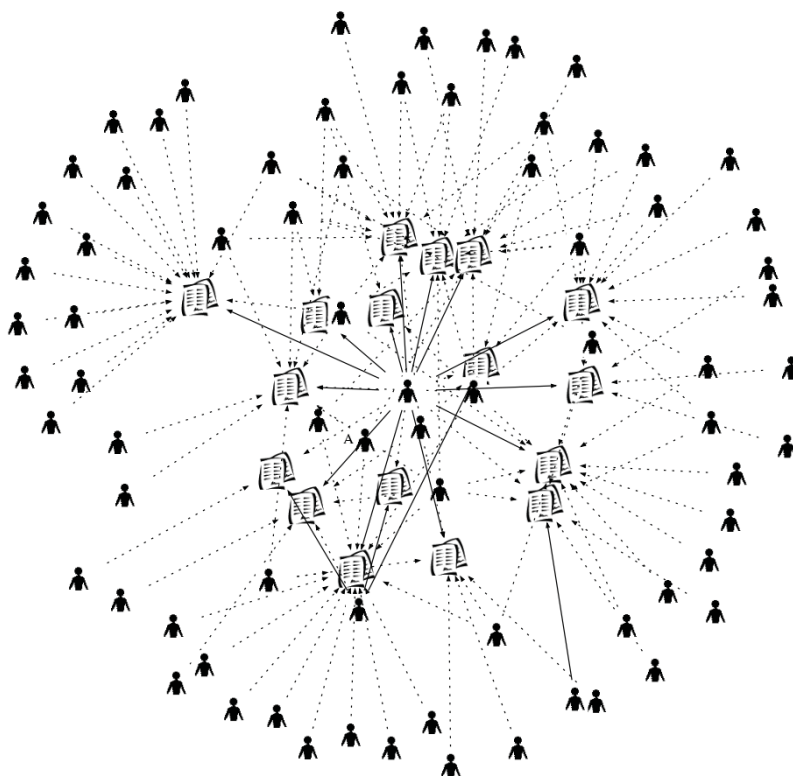


Рис. 12. Социограмма студии Scratch

внутри сообщества Scratch пути развития вычислительной педагогики могут быть перенесены в миры других образовательных сообществ. В первую очередь это будет происходить и уже происходит в сообществах с общей концепцией обучения, основанной на конструкционизме. □

Литература

1. *Yasar O.* Computational Pedagogical Content Knowledge (CPACK): Integrating Modeling and Simulation Technology into STEM Teacher Education. Association for the Advancement of Computing in Education (AACE), 2015. P. 3514–3521.
2. *Журавлев А.Л.* Психология совместной деятельности. — Институт психологии РАН, 2005. — 640 с.
3. *Harel I., Papert S.* Constructionism: research reports and essays, 1985–1990. Ablex Pub. Corp., 1991. 540 p.
4. *Minsky M.* Introduction to LogoWorks // LogoWorks: Challenging Programs in Logo / ed. Solomon C., Minsky M., Harvey B. McGraw-Hill Osborne Media, 1986. P. 388.
5. *Патаракин Е.Д.* Педагогический дизайн социальной сети Scratch // Образовательные технологии и общество (Educational Technology & Society). — 2013. — № 2. — С. 505–528.
6. *Garcia D., Harvey B., Barnes T.* The Beauty and Joy of Computing // ACM Inroads. 2015. Vol. 6, № 4. P. 71–79.
7. *Harvey B.* Professor Harvey's Introduction to Abstraction | Computing Portal [Electronic resource]. URL: <http://www.computingportal.org/node/7716> (accessed: 04.04.2018).
8. *Román-González M., Moreno-León J., Robles G.* Complementary tools for computational thinking assessment // Proceedings of International Conference on Computational Thinking Education (CTE 2017), S.C Kong, J Sheldon, and K.Y Li (Eds.). The Education University of Hong Kong. 2017. P. 154–159.
9. *Царапкина Ю.М., Воробьева К.П.* Развитие социальных компетенций студентов при использовании метода модерации // Историческая и социально-образовательная мысль. — 2014. — № 3 (25). — С. 148–152.
10. *Патаракин Е.Д., Шилова О.Н.* Развитие педагогического дизайна для совместной сетевой деятельности субъектов образования // Человек и образование. — 2015. — № 43 (2). — С. 20–25.
11. *Kafai Y.B., Burke Q.* Connected Code: Why Children Need to Learn Programming. MIT Press, 2014. 200 p.
12. *Dasgupta S., Hill B.M.* Scratch Community Blocks: Supporting Children As Data Scientists // Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. New York, NY, USA: ACM, 2017. P. 3620–3631.
13. *Reynolds R., Caperton I.H.* Comparison of Middle School, High School and Community College Students' Wiki Activity in Globaloria-West Virginia: (Pilot Year-two) // Proceedings of the 5th International Symposium on Wikis and Open Collaboration. New York, NY, USA: ACM, 2009. P. 29:1–29:2.
14. *Lerner R.M.* Agent-Based Modeling as a Social Activity. NORTHWESTERN UNIVERSITY, 2014. 229 p.
15. *Klopfer E. et al.* The Simulation Cycle: Combining Games, Simulations, Engineering and Science Using StarLogo TNG // E-Learning and Digital Media. 2009. Vol. 6, № 1. P. 71–96.
16. *Kelleher C., Pausch R.* Using Storytelling to Motivate Programming // Commun. ACM. 2007. Vol. 50, № 7. P. 58–64.
17. *Galley R. et al.* Cloudworks as a 'pedagogical wrapper' for LAMS sequences: supporting the sharing of ideas across professional boundaries and facilitating collaborative design, evaluation and critical reflection. Oxford, 2010.
18. *Gaines B.R., Shaw M.L.G.* Sociocognitive Inquiry // Social Network Mining, Analysis, and Research Trends: Techniques and Applications / ed. Ting I.-H., Hong T.-P., Wang L.S.-L. IGI Global, 2012. P. 35–55.
19. *Журавлев А.Л.* Коллективный субъект: основные признаки, уровни и психологические типы // Психологический журнал. — 2009. — № 5. — С. 72–80.
20. *West M.A.* Reflexivity and work group effectiveness: A conceptual integration // Handbook of work group psychology / ed. West M.A. Chichester, UK: Wiley, 1996. P. 555–579.
21. *Raven J., Gallon L.* Conceptualising, Mapping, and Measuring Social Forces // Journal of Sociocybernetics. 2010. P. 73–110.
22. *Patarakin E.D.* Wikigrams-Based Social Inquiry // Digital Tools and Solutions for Inquiry-Based STEM Learning. IGI Global, 2017. Vol. 1. P. 112–138.

References

1. *Yasar O.* Computational Pedagogical Content Knowledge (CPACK): Integrating Modeling and Simulation Technology into STEM Teacher Education. Association for the Advancement of Computing in Education (AACE), 2015. P. 3514–3521.
2. *Zhuravlev A.L.* Psychology of joint activities. — Institute of psychology RAS 2005. — 640 p.
3. *Harel I., Papert S.* Constructionism: research reports and essays, 1985–1990. Ablex Pub. Corp., 1991. 540 p.
4. *Minsky M.* Introduction to LogoWorks // LogoWorks: Challenging Programs in Logo / ed. Solomon C., Minsky M., Harvey B. McGraw-Hill Osborne Media, 1986. P. 388.
5. *Patarakin E.D.* Pedagogical design of the social network scratch // Educational technologies and society (educational technologies and society). — 2013. — № 2. — P. 505–528.
6. *Garcia D., Harvey B., Barnes T.* The Beauty and Joy of Computing // ACM Inroads. 2015. Vol. 6, № 4. P. 71–79.
7. *Harvey B.* Professor Harvey's Introduction to Abstraction | Computing Portal [Electronic resource]. URL: <http://www.computingportal.org/node/7716> (accessed: 04.04.2018).
8. *Román-González M., Moreno-León J., Robles G.* Complementary tools for computational thinking assessment // Proceedings of International Conference on Computational Thinking Education (CTE 2017), S.C Kong, J Sheldon, and K.Y Li (Eds.). The Education University of Hong Kong. 2017. P. 154–159.
9. *Tsarapkina Y.M., Vorobyova K.P.* Development of social competencies of students using the method of moderation // Historical and socio-educational thought. — 2014. — № 3 (25). — Pp. 148–152.
10. *Patarakin E.D., Shilova O.N.* Development of pedagogical design for joint network activity of subjects of education. Man and education. — 2015. — № 43 (2). — P. 20–25.
11. *Kafai Y.B., Burke Q.* Connected Code: Why Children Need to Learn Programming. MIT Press, 2014. 200 p.
12. *Dasgupta S., Hill B.M.* Scratch Community Blocks: Supporting Children As Data Scientists // Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. New York, NY, USA: ACM, 2017. P. 3620–3631.
13. *Reynolds R., Caperton I.H.* Comparison of Middle School, High School and Community College Students' Wiki Activity in Globaloria-West Virginia: (Pilot Year-two) // Proceedings of the 5th International Symposium on Wikis and Open Collaboration. New York, NY, USA: ACM, 2009. P. 29:1–29:2.
14. *Lerner R.M.* Agent-Based Modeling as a Social Activity. NORTHWESTERN UNIVERSITY, 2014. 229 p.
15. *Klopfer E. et al.* The Simulation Cycle: Combining Games, Simulations, Engineering and Science Using StarLogo TNG // E-Learning and Digital Media. 2009. Vol. 6, № 1. P. 71–96.
16. *Kelleher C., Pausch R.* Using Storytelling to Motivate Programming // Commun. ACM. 2007. Vol. 50, № 7. P. 58–64.
17. *Galley R. et al.* Cloudworks as a 'pedagogical wrapper' for LAMS sequences: supporting the sharing of ideas across professional boundaries and facilitating collaborative design, evaluation and critical reflection. Oxford, 2010.
18. *Gaines B.R., Shaw M.L.G.* Sociocognitive Inquiry // Social Network Mining, Analysis, and Research Trends: Techniques and Applications / ed. Ting I.-H., Hong T.-P., Wang L.S.-L. IGI Global, 2012. P. 35–55.
19. *Zhuravlev A.L.* Collective subject: basic features, levels and psychological types // Psychological journal. — 2009. — № 5. — P. 72–80.
20. *West M.A.* Reflexivity and work group effectiveness: A conceptual integration // Handbook of work group psychology / ed. West M.A. Chichester, UK: Wiley, 1996. P. 555–579.
21. *Raven J., Gallon L.* Conceptualising, Mapping, and Measuring Social Forces // Journal of Sociocybernetics. 2010. P. 73–110.
22. *Patarakin E.D.* Wikigrams-Based Social Inquiry // Digital Tools and Solutions for Inquiry-Based STEM Learning. IGI Global, 2017. Vol. 1. P. 112–138.