

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
федеральное государственное бюджетное образовательное учреждение высшего  
образования  
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ  
им. В.П. АСТАФЬЕВА  
(КГПУ им. В.П. Астафьева)

Институт/факультет

Институт математики, физики и информатики  
(полное наименование института/факультета/филиала)

Выпускающая кафедра

Базовая кафедра информатики и  
информационных технологий в образовании  
(полное наименование кафедры)

**Коротенко Марина Николаевна**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема Коллективно-распределенная проектная деятельность учащихся при  
обучении программированию в условиях образовательной  
технологической платформы «Мега-класс»

Направление подготовки 44.03.05 Педагогическое образование  
(код и наименование направления)

Профиль Математика и информатика  
(наименование профиля для бакалавриата)

ДОПУСКАЮ К ЗАЩИТЕ

Заведующий кафедрой  
д.п.н., профессор Пак Н.И.  
(ученая степень, ученое звание, фамилия, инициалы)

(дата, подпись)

Руководитель к.п.н., Сокольская М.А.

(ученая степень, ученое звание, фамилия, инициалы)

Дата защиты \_\_\_\_\_

Обучающийся Коротенко М.Н.

(фамилия, инициалы)

(дата, подпись)

Оценка \_\_\_\_\_

(прописью)

Красноярск 2017

## ОГЛАВЛЕНИЕ

Введение.....	3
Глава 1. Коллективно-распределенная проектная деятельность учащихся при обучении программированию в условиях образовательной технологической платформы «Мега-класс».....	7
1.1 Сущность образовательного кластера на основе образовательной технологической платформы «Мега-класс».....	7
1.2 Проектная деятельность учащихся и ее роль в современном образовании .....	11
1.3 Метод проектов при обучении программированию в старшей школе	15
Выводы по главе 1.....	20
Глава 2. Организация и сопровождение программных проектов учащихся старшей школы в условиях образовательной технологической платформы «Мега-класс».....	21
2.1 Организационно-методические условия проектной деятельности учащихся по программированию в условиях образовательной технологической платформы «Мега-класс».....	21
2.1.1. Общие организационно-методические условия построения и проведения мега-урока .....	21
2.1.2. Условия организации и сопровождения проектов по программированию в мега-классе .....	25
2.2 Комплекс проектных заданий по программированию .....	41
2.3 Методические рекомендации для сопровождения проектов по программированию условиях образовательной технологической платформы «Мега-класс».....	66
Выводы по главе 2.....	75
Заключение .....	76
Список литературы .....	78

## Введение

Стремительно развивающиеся информационные технологии приносят глубокие изменения во все области жизни. В нашей стране и за рубежом интенсивно идет поиск новых моделей образования, при этом в качестве движущей силы модернизации всех образовательных процессов рассматривается развитие инновационных подходов к организации обучения на основе широкого и активного использования информационных и коммуникационных технологий. Информатизация образования рассматривается как одно из основных направлений государственной образовательной политики [1].

Отличительной чертой нового образования становятся виртуальные и «средовые» методы обучения, опирающиеся на мобильные устройства и сервисы Интернет. Образовательный процесс виртуально выходит за рамки школы, осуществляется уже в информационно-технологической сетевой инфраструктуре, в которой интегрируются традиционные и инновационные технологии обучения. Однако образование едва успевает перестраиваться и идти в ногу с современными трендами общества. Углубляется и актуализируется противоречие между необходимостью совершенствовать подготовку будущих педагогов к профессиональной деятельности в ИКТ-насыщенных образовательных средах, менять формы и методы классно-урочной модели обучения школьников и отсутствием эффективных технологий, обеспечивающих малозатратный, массовый и непрерывный учебный процесс, адекватный настоящим вызовам времени [8].

Динамизм ИКТ заставляет постоянно пересматривать концепцию и содержание курсов информатики, как в школе, так и в вузе. Классно-урочная система не готова изучать и использовать сетевые технологии, электронные и дистанционные образовательные технологии в реальной жизненной практике. В этой связи обучение информатике целесообразно осуществлять в корпоративной, практико-ориентированной и исследовательской среде, в частности в образовательных кластерах [19].

Проект «Мегакласс» ставит своей целью объединить усилия учителей школ и педагогов высшего образования, школьников и студентов вуза в поиске новых моделей школьного и высшего педагогического образования в интегрированном сетевом информационном образовательном пространстве для достижения каждой групп участников лично-значимых образовательных и профессиональных результатов [7].

Алгоритмизация различных процессов — один из важнейших инструментов информатики, ее фундаментальные основы, позволяющие передать компьютеру решение сложнейших интеллектуальных задач самого разного содержания. Одним из фундаментальных принципов работы информатики является принцип программного управления компьютера. Понять его невозможно, не зная, что такое программа для ЭВМ. Понимание фундаментальных идей алгоритмизации в контексте автоматизации процессов решения разнообразных задач — одна из важных задач общеобразовательного курса информатики.

Любой школьный предмет наряду с образовательной, развивающей, воспитательной функцией должен выполнять и профориентационную функцию. Программирование является важнейшей профессиональной областью деятельности, весьма значимой, современной, престижной.

В курсе старшей школы программирование является довольно сложным разделом информатики, и, разумеется, перед каждым учителем встаёт вопрос о выборе средств и методов обучения с целью обеспечения максимальной эффективности усвоения материала. Успешность обучения всецело зависит от мотивации школьника, от наличия стремления получения знаний. При организации поисковой, исследовательской деятельности при решении проблем происходит развитие умения формулировать и проверять гипотезы, углубляются представления о возможностях компьютера в автоматизированной обработке и хранении информации. Также актуализируется значимость человеческого интеллекта: программа настолько хороша, насколько умён её создатель. Проектная деятельность в совокупности

с обучением в образовательном кластере поможет не только повысить мотивацию, развить творческие потенциалы учеников, но и обеспечить более сильное, разностороннее усвоение материала.

Систематическое и целенаправленное использование проектной деятельности в среде образовательного кластера в процессе обучения школьников программированию повышает уровень эффективности обучения, способствует развитию и поддержанию интереса к разделу, а также развитию различных форм мыслительной деятельности.

Таким образом, **объектом** нашего исследования является процесс обучения программированию в общеобразовательной школе.

**Предмет** исследования — проектная деятельность учащихся старшей школы в среде образовательного кластера как средство обучения программированию.

**Целью** работы является разработка комплекса проектных заданий для обучения программированию на основе коллективно-распределенной проектной деятельности в условиях образовательного кластера.

**Задачи исследования:**

1. Изучить литературу по теме исследования с целью выявления особенностей образовательной технологической платформы “Мега-класс”.
2. Изучить сущность и особенности проектной деятельности в обучении информатике в школе.
3. Выделить организационно-методические условия организации проектной деятельности при обучении программированию в условиях образовательной технологической платформы “Мега-класс”.
4. Разработать комплекс проектных заданий по программированию разных уровней и типов, адаптированных для использования в условиях образовательной технологической платформы “Мега-класс”.
5. Разработать методические рекомендации для использования проектной деятельности в обучении программированию в условиях образовательной технологической платформы “Мега-класс”.

Базой эксперимента являются участники образовательной технологической платформы “Мега-класс” - инженерно-технологический кластер (10-е классы), в составе: МАОУ Гимназия № 9 (г. Красноярск), Школа-лицей “Туран” (г. Алматы), Лицей №1 (г. Ачинск), Лицей 28 (г. Красноярск), МАОУ “Школа №3” (г. Ачинск), КГПУ им. В.П. Астафьева (г. Красноярск). По промежуточным результатам исследования опубликованы тезисы на конференции (Молодежь и наука 21 века, Красноярск, 19-20 мая 2015г.).

Выпускная квалификационная работа состоит из введения, двух глав, заключения и списка литературы.

# **Глава 1. Коллективно-распределенная проектная деятельность учащихся при обучении программированию в условиях образовательной технологической платформы «Мега-класс»**

## **1.1 Сущность образовательного кластера на основе образовательной технологической платформы «Мега-класс»**

Современные парадигмы образования в условиях глобальной информатизации и коммуникации связывают с электронным обучением, искусственным интеллектом, облачными технологиями. Они нацеливаются на развитие когнитивных способностей человека, самообразование и производство знаний высоких технологий, инноваций и информационных услуг. Отличительной чертой нового образования становятся виртуальные и «средовые» методы обучения, основанные на Интернет-технологиях, Интернет-индустрии. Образовательный процесс виртуально выходит за рамки школы, осуществляется уже в информационно-технологической сетевой инфраструктуре, в которой интегрируются беспроводные, сенсорные, семантические сегменты смарт-Интернета. Однако, в силу своего консерватизма, образование едва успевает перестраиваться и идти в ногу с современными трендами общества. Налицо кричащее противоречие между необходимостью обеспечить новыми профессиональными компетенциями действующего педагога, совершенствовать подготовку будущих учителей в условиях насыщенной ИКТ-среды, изменить классно-урочную модель обучения на сетевую, реально-виртуальную систему обучения и отсутствием эффективных моделей, обеспечивающий малозатратный, массовый, непрерывный и интегрированный образовательный процесс, адекватный настоящим вызовам времени [5].

Сетевое общество формирует новую педагогику сетевого взаимодействия, которая определяет науку учиться на расстоянии, учиться с использованием дистанционных ресурсов, дистанционных средств и инструментов, учиться с помощью коллективного разума в совместных

сетевых проектах [14]. Классно-урочная система не готова изучать и использовать сетевые технологии, электронные и дистанционные образовательные технологии в реальной жизненной практике. В этой связи обучение информатике целесообразно осуществлять в корпоративной, практико-ориентированной и исследовательской среде, в частности в образовательных кластерах.

Под образовательным кластером понимают совокупность взаимосвязанных учреждений профессионального образования, объединенных по отраслевому признаку и партнерскими отношениями с предприятиями отрасли. В образовательных кластерах потенциально возможно интегрировать науку, образование и жизнь, осуществлять непрерывную практико-ориентированную профессиональную подготовку молодежи без коренной реконструкции сложившихся способов жизнедеятельности, организации рабочего пространства участников кластера (школа, вуз, производство) за счёт преимуществ облачных технологий, электронных форм и средств обучения. Для организации целесообразной и планируемой деятельности в кластере необходимо обозначить контуры такой образовательной технологической платформы, в которой без «капитальной реконструкции» существующих сфер образования, науки и бизнеса возможна их кооперация и корпорация. Подобная платформа может быть выстроена на основе идей и принципов проекта «Мега-класс» [7].

При реализации модели обучения по технологии Мега-класс достигаются следующие эффекты:

1. Для школьника — существенное повышение интереса к обучению в школе за счет смены классно-урочной на кластерную парадигму обучения «все-со-всеми» в условиях реальной жизни, равного и доступного образования, возможности получить качественное образование для будущего, удовлетворяющего его родителей, непосредственного развития своих коммуникативных компетенций, когнитивных способностей.

2. Для студента педвуза - существенное повышение интереса к обучению



за счет смены скучной учебной аудиторной работы к деятельности в рамках парадигмы «мастер-подмастерье» при участии в проведении реальных мега-уроков, что обеспечивает ему мотивированное обучение предметам и педагогическую практику, непосредственное приобретение профессиональных умений и навыков в учебной работе.

3. Для практикующего учителя — непрерывное повышение квалификации во время его непосредственной профессиональной деятельности за счет совместной в кластере работе с профессурой педвуза, учеными и специалистами IT-фирм.

4. Для преподавателей и ученых вузов — сближение академической и педагогической науки с реальной школьной практикой, с жизнью.

5. Для работников IT-индустрии — привлечение трудовых и интеллектуальных ресурсов для продвижения своих товаров и услуг, предпринимательской деятельности [20].

Объединяющим элементом кластера является информационно-образовательная среда, представляющая собой совокупность сервисов и технологий сети, наполненных дидактическими и методическими материалами, предоставляющих различные возможности для общения и организации коллективно-распределенной работы. Наполнением среды совместно занимаются все участники кластера, но есть и разделение сфер деятельности. Кластерная научная лаборатория объединяет ученых разных университетов, представителей бизнеса и «генерирует» идеи проектов и исследований, в которых могут на том или ином этапе принять участие школьники (самостоятельно или под руководством студентов учителей или непосредственно исследователей). Учителя и преподаватели корректируют содержательную часть и ведут методические разработки. Студенты принимают участие в разработке ЦОР и также вносят свои (иногда очень интересные и необычные) идеи в методическую составляющую процесса обучения.

В структуру проектируемой образовательной технологической

платформы «Мега-класс» на текущем, экспериментально-проектировочном этапе включены:

1. Инженерно-технологический кластер;
2. IT-кластер;
3. Северо-арктический кластер.

Рассмотрим состав участников, основные задачи и некоторые аспекты работы инженерно-технологического кластера, так как именно он послужил для частичной апробации материалов данной дипломной работы.

Инженерно-технологический кластер - открытая платформа с базовыми организациями: СФУ, СибГАУ им. академика М.Ф. Решетнева, КГПУ им В.П. Астафьева, КазНПУ имени Абая (Алматы, Казахстан), школы с классами инженерно-технологического профиля (МАОУ Гимназия № 9 г. Красноярск, Школа-лицей «Туран» г. Алматы, МОУ Лицей №1 г. Ачинск, МБОУ Лицей 28 г. Красноярск, МАОУ «Школа №3» г. Ачинск). Штаб кластера — МАОУ гимназия 9 г. Красноярск. Деятельность кластера нацелена на развитие инженерного мышления старшеклассников, формирование у всех участников компетенций в сфере коллективной распределенной деятельности. Предметная основа работы кластера: информатика, робототехника, математика и физика. Систематическая работа данного кластера началась в сентябре 2016 года. На данный момент у учителей информатики отмечается не только рост интереса к проекту, но и повышение творческой инициативы, увеличение количества интересных идей, постоянный поиск методов обучения и форм организации учебной деятельности в новых условиях. Учащиеся проявляют больший интерес к урокам информатики, находят для себя темы и направления для дальнейшего изучения [21].

В области информатики большое внимание уделяется программированию, как одному из ключевых разделов, образующих предметное содержание профильного курса школьной информатики. В дальнейшей работе все внимание сконцентрировано именно на процессе обучения программированию в связи со сложностью и объемом изучения

данного раздела и, в то же время, с большим развивающим потенциалом программирования.

## **1.2 Проектная деятельность учащихся и ее роль в современном образовании**

Пересмотр знаний в сложившейся системе образования стал причиной того, что в последние годы в обществе сложилось новое понимание главной цели образования: формирование личности, готовой к саморазвитию. Реализация этой цели требует выполнения целого комплекса задач, среди которых:

- обучение деятельности — умению ставить цели, организовывать свою деятельность для их достижения и оценивать результаты своих действий;
- формирование личностных качеств — нравственных качеств, познавательных мотивов деятельности;
- формирование картины мира, адекватной современному уровню знаний и уровню образовательной программы.

Все эти цели и задачи успешно реализуются при деятельностном подходе в обучении, который лежит в основе ФГОС второго поколения. Отличительной особенностью нового стандарта является его деятельностный характер, ставящий главной целью развитие личности учащегося. Система образования отказывается от традиционного представления результатов обучения в виде знаний, умений и навыков, формулировки стандарта указывают реальные виды деятельности, которыми учащийся должен овладеть к концу обучения [22]. Качество образования на современном этапе понимается как уровень специфических, надпредметных умений, связанных с самоопределением и самореализацией личности. На первый план выходит личность ученика, готовность его к самостоятельной деятельности по сбору, обработке, анализу и организации информации [29].

Исходя из этого, иной становятся задачи учителя — не поучить, а побудить, не оценить, а проанализировать. Учитель по отношению к ученику перестает быть источником информации, а становится организатором получения информации, источником духовного и интеллектуального импульса, побуждающего к действию. Главной задачей учителя является организация учебной деятельности таким образом, чтобы у учащихся сформировались потребность в осуществлении творческого преобразования учебного материала с целью овладения новыми знаниями.

Деятельностный подход — это обучение, реализующее принцип деятельности, метод обучения, при котором обучающийся не получает знания в готовом виде, а добывает их сам в процессе собственной учебно-познавательной деятельности. Деятельностный подход ломает многие привычные стереотипы подготовки и проведения уроков, меняет саму систему взаимоотношений «учитель — ученик» [2].

Особенностью нового образовательного стандарта является то, что учитель должен творчески подходить к процессу обучения и воспитания, при этом используя как можно больше на своих уроках существующую информационную базу. Поэтому одной из основных технологий, обеспечивающей системно — деятельностный подход к воспитанию и обучению, является метод проектов, который практически вбирает в себя элементы многих других современных технологий, например, обучение в сотрудничестве, проблемный метод обучения и др.

Метод проектов — это технология достижения требуемой дидактической цели через подробную проработку конкретного вопроса, которая в результате должна иметь какой-либо оформленный вид [16]. Данный метод был разработан для того, чтобы найти различные способы и пути активного самостоятельного развития мышления у ребенка с целью не просто научить его запоминать и воспроизводить знания, получаемые им в школе, а уметь применять их на практике. В целом, метод проектов является целенаправленной, полностью самостоятельной деятельностью обучающихся,

но руководимой учителем, направленной на решение какой-либо исследовательской или социально значимой проблемы, а также на достижение конкретного результата в виде какого-либо идеального или материального продукта. Другими словами, результатом проекта ученика может стать идеальный продукт (выработанное после изучения информации умозаключение, вывод, какое-либо полученное знание) или материальный продукт (участие в какой-либо деятельности, сопровождаемое ведением записей в дневнике, альбом, рекламный проспект, выпуск газеты и пр.).

Хочется отметить, что само по себе обучение с помощью проектов активно воздействует на мотивационную сферу учащегося, что на сегодняшний момент достаточно актуально [10]. Работа над проектом позволяет находить решение к таким важным задачам, как:

- обучение планированию (учащийся должен уметь четко определить цель, описать основные шаги по достижению поставленной цели, концентрироваться на достижении цели на протяжении всей работы);
- формирование навыков сбора и обработки информации, материалов (учащийся должен уметь выбрать подходящую информацию и правильно ее использовать);
- формирование умения анализировать (креативность и критическое мышление);
- формирование умения составлять письменный отчет (учащийся должен уметь составлять план работы, презентовать информацию);
- формирование позитивного отношения к работе (учащийся должен проявлять инициативу, энтузиазм, стараться выполнить работу в срок в соответствии с установленным планом и графиком работы);
- организация проектной деятельности с целью формирования проектного мышления учащегося;
- мотивация детей на получение знаний;
- включенность всех учащихся в самостоятельную работу;
- самостоятельное «добывание» требуемых знаний из различных

информационных источников (в том числе сети Интернет);

- развитие умения пользоваться полученными знаниями для разрешения новых познавательно-практических задач.

К важным положительным факторам проектной деятельности можно отнести:

- повышение мотивации и развитие творческих способностей происходит из-за наличия в проектной деятельности ключевого признака — самостоятельного выбора;

- развитие творческих способностей и смещение акцента от инструментального подхода к технологическому происходит благодаря необходимости осмысленного выбора инструментария и планирования деятельности для достижения лучшего результата;

- формирование чувства ответственности происходит подсознательно — учащийся стремится доказать, в первую очередь, самому себе, что он сделал правильный выбор.

Основные требования к использованию метода проектов:

1. Наличие значимой в творческом, исследовательском плане проблемы, требующей интегрированного знания, исследовательского поиска для ее решения (например, исследование влияния антропогенной нагрузки на экосистему леса/пруда, изучение проблемы утилизации полимерных бытовых отходов, изучение вопросов энергоэффективности с последующей разработкой конкретных мер по экономии энергоресурсов и т. д.).

2. Практическая, теоретическая, познавательная значимость предполагаемых результатов (например, доклад в соответствующие службы о состоянии данной экосистемы, факторах, влияющих на это состояние, тенденциях, прослеживающихся в развитии данной проблемы, совместный выпуск газеты, альманаха с репортажами с места событий, проект закона об охране лесов в разных местностях, видеоклип и т. д.)

3. Самостоятельная (индивидуальная, парная, групповая) деятельность учащихся.

4. Структурирование содержательной части проекта (с указанием поэтапных результатов).

Таким образом, грамотно организованная проектная деятельность обучающихся позволит реализовать главную цель современного образования — научить детей учиться [22].

### **1.3 Метод проектов при обучении программированию в старшей школе**

Главная проблема школьного обучения программированию — отсутствие системного подхода в этом вопросе. В большинстве своем в школе учат не решению проблем с помощью программирования, не разработке программ, а лишь языку программирования как таковому. Школьные уроки сводятся, по сути, к изучению конструкций языка и выполнению каких-либо заданий на эти конструкции, но не учат намного более важному умению — применять их для решения возникающих на практике задач [18]. Те из учащихся, кто имеет талант к программированию, учатся этому самостоятельно, остальные получают представление о программировании как «скучном и нудном занятии для ботаников».

Д. Дьюи был твердо уверен, что ученик в школе не должен получать готовый результат чужого опыта, он должен сам принимать участие в «активности», ведущей к известным заключениям. Любой учебный предмет должен быть связан с какой-нибудь жизненной ситуацией, таким образом, он сможет увидеть приложение к жизни своего знания [13].

Разработка методики обучения программированию, направленной на развитие самостоятельности, критического мышления, творческой активности - это сложный процесс соотнесения синтаксиса и семантики языков программирования, принципов разработки программ, специализированных алгоритмов с методами их усвоения. В настоящее время для актуализации и закрепления знаний, умений и навыков по программированию используются

небольшие задачи (решение каждой из них не превышает одного занятия). Однако в формировании мотивационной сферы обучаемых при этом возникают существенные затруднения, не создаются условия для самостоятельного приобретения знаний учащимися, объем полученных знаний не находит своего применения в конкретных жизненных ситуациях [26].

Разработка методики, позволяющей оптимально сочетать изучение фундаментальных знаний с формированием практических умений и навыков является важной проблемой при обучении программированию.

Основу данной методики составляет "метод проектов", классическое понимание которого представляет собой широкое внедрение учебных проектов (проблемных задач высокой сложности, рассчитанных на долговременность решения) в процесс обучения. Он ориентирован на гуманистичный, индивидуальный подход к личности ученика, предлагающий индуктивный путь учебного познания (учение через открытие, через разрешение проблемных ситуаций). Применение метода проектов характеризуется высоким мотивационным потенциалом, возможностями развития творческих способностей и самостоятельности, что позволяет рассматривать его как эффективный способ познания.

Однако внедрение метода проектов в процесс обучения программированию учащихся старших классов сопровождается рядом частных противоречий, требующих своего разрешения. Д. А. Слинкин в своей работе о использовании метода проектов при обучении программированию выделяет следующие противоречия:

- между требованием к систематичности и фундаментальности знаний в старших классах общеобразовательной школы и усложнением организации получения учеником таких знаний при обучении на основе метода проектов по сравнению с традиционными методиками;
- между широкими возможностями метода проектов в развитии мотивационной сферы обучаемых и недостаточной разработанностью



теоретических положений, обосновывающих направленность познавательной активности школьника на достижение целей учебного познания через реализацию программного проекта.

Каждое из противоречий является фактором, сдерживающим применение метода проектов. Во-первых, слабая разработанность технологической базы затрудняет обучение школьников проектной деятельности на основе системного подхода. Во-вторых, слабый контроль над направленностью познавательной активности школьника в старших классах снижает ее эффект с точки зрения получения обучаемым запланированных в программе курса знаний, умений и навыков [26].

Проекты сложны и объемны сами по себе, а проекты по программированию в свою очередь, зачастую, дополнительно требуют знаний по другим предметам (например, английский язык, математика, биология и др.), или по темам, которые не предусмотрены в школьном курсе информатики. Интернет, конечно, великая сила, но там очень много лишней информации, поэтому, необходимо предоставлять множество пособий для выполнения проекта, чтобы в случае недостаточности знаний ученик в любой момент мог легко их восполнить. Также, нужно обеспечивать мотивацию школьников на каждом этапе выполнения проекта, чтобы при возникновении сложностей проект не забрасывался, не откладывался в сторону.

Дополнительной мотивацией будет выступать постоянный контроль выполнения проектов, чтобы отслеживать этапы работы, оценивать достижения, а также, в случае наличия затруднений, вовремя помочь. Иначе, ученик не будет видеть прогресса, или вообще, не сумеет самостоятельно разобраться с трудностями, потеряет интерес. Для этих же целей необходимо наличие постоянных консультаций, тьюторов, с которыми у учащихся была бы налаженная связь, также для решения возникающих вопросов. В традиционной школьной практике обучения программированию роль тьютора автоматически ложится на учителя информатики.

Конечно, очень сложно обучать школьников проектной деятельности на

основе системного подхода. На первоначальном этапе можно внедрить проектную деятельность точно, для конкретных уроков. Но, если проекты будут динамические, и постепенно расширяться по мере изучения новых тем, то можно добиться некоторой системности и комплексности.

Для учителя нужно выделить важность предварительной подготовки материалов проекта. Предварительная подготовка включает в себя следующие обязательные части:

1. Кейс с техническим заданием. Перед тем как отдать задание на разработку, нужно не только соответствующе оформить его, но и предварительно обсудить с учащимися, внести правки, может и у учеников появятся кое-какие идеи, и их не в коем случае не стоит оставлять без внимания. Самое главное, чтобы ТЗ было понято и принято всеми участниками образовательного процесса, поэтому так важно предварительно обсуждать его, комментировать этапы и устранять недостатки.

2. Список материалов по теме проекта. Нужно всегда продумывать все наперед и учитывать разносторонность и любознательность учеников. Вдруг он уйдет куда-то в сторону, или глубже чем подразумевает задание, энтузиазм ведь всегда приветствуется. Да и просто в случае недостаточности знаний необходимо предоставлять ученику как можно больше ресурсов для их пополнения. Важно чтобы они были достоверными, не устаревшими и легки в самостоятельном освоении. Это могут быть не обязательно книги, а, например, онлайн курсы, вебинары, вами же созданные вспомогательные средства для обучения, форумы и т.д.

3. Поиск консультантов. Стоит учитывать, что вы не можете знать всего, и заранее подстраховать себя. Если вы знаете, например, практикующих программистов; других учителей, которые сильны в некоторых направлениях больше вас; преподавателей из университетов и т.д., следует подумать о том что они могли бы вам помочь с консультированием учеников по некоторым вопросам, и заранее договориться с ними об этом. Не каждый, конечно, согласится тратить время на работу, за которую ему не платят, вот тут-то и

назревает проблема. Было бы хорошо найти людей, с которыми взаимовыгодно будет сотрудничать в этом вопросе. Как раз в этом нам и может помочь образовательный кластер.

## **Выводы по главе 1**

1. В связи с усугублением противоречий между динамичным научно-технологическим прогрессом информационного общества и недопустимо отстающими от него методами и средствами обучения в традиционной классно-урочной системе, обучение информатике целесообразно осуществлять в корпоративной, практико-ориентированной и исследовательской среде, в частности в образовательных кластерах.

2. Участие в проектной деятельности позволяет учащимся приобрести уникальный опыт, невозможный при других формах обучения. Внедрение данной технологии в процесс обучения улучшает качество образовательной работы, повышает производительность труда учителя, активизирует познавательную деятельность учащихся на уроке, способствует повышению самостоятельности учащихся при изучении нового материала.

3. Использование метода проектов при обучении программированию позволяет приблизить процесс обучения к научно-исследовательской, практико-ориентированной деятельности, что положительно сказывается на создании познавательных потребностей. Применение метода проектов при обучении программированию позволяет существенно повысить мотивацию учащихся к обучению, развивает самостоятельность, самоорганизованность, и творческие способности.

## Глава 2. Организация и сопровождение программных проектов учащихся старшей школы в условиях образовательной технологической платформы «Мега-класс»

### 2.1 Организационно-методические условия проектной деятельности учащихся по программированию в условиях образовательной технологической платформы «Мега-класс»

#### 2.1.1. Общие организационно-методические условия построения и проведения мега-урока

*Организационно-методические условия* достижения новых профессиональных и образовательных результатов участников проекта МЕГА-класс находят свое отражение в следующих положениях:

1. Программа курса информатики в старших классах школ-участников проекта *согласуется и ставится единой* для всех участников.

2. Система Мега-уроков *органично встраивается* в учебную программу курса информатики старшей школы, реализуемую в общеобразовательной школе.

3. Мега-уроки *фокусно встраиваются* в тематическое планирование курса уроков информатики по предварительному согласованию запросов учителей и преподавателей вуза не реже одного Мега-урока в месяц.

4. Сетевое *профессиональное сотрудничество* в сообществе МЕГА-учитель (учителя школ, студенты, магистранты, преподаватели вуза и др.) является одним из важнейших требований при проектировании, как целостной системы МЕГА-уроков, так и каждого урока в отдельности, осуществляется непрерывно в форме еженедельных веб-семинаров всех участников или участников конкретного Мега-урока.

5. Студентов можно привлекать для работы в качестве тьюторов, разработки материалов, но настоящее исследование не акцентируется на этом вопросе [7].

Для организации мегауроков создается их методическое обеспечение в

виде сценариев взаимодействия всех участников кластера, облачных сервисов (заготовки и шаблоны презентаций, видео, аудио материалов, электронные журналы и пр.), указаний каждому учителю, студенту педвуза, преподавателям и привлекаемым профессорам и ученым.

Ведет мега-урок модератор, который обеспечивает координированную деятельность учителей, тьюторов, студентов и преподавателей, вовлеченных в этот учебный процесс.

*Мега-урок* проводится одновременно во всех школах кластера, с участием преподавателей и студентов, которые совместно с учителями школ готовили очередной урок и проводят согласно концепции сетевого курса. Облако представляет порталы обучающих средств (ментальные учебники, видеолекции и пр.), диагностик качества обучения, компетенций учащихся, разработок учащихся, студентов и преподавателей по актуальным проблемам науки и общества [17].

Уроки предусматривают регламент сетевого взаимодействия всех участников по аналогии с деятельностью международных промышленных корпораций (например, автомобильная промышленность).

Модель мега-учителя определяет сообщество учителей и преподавателей, распределенных по школам и вузам, связанных кластерными отношениями с главным модератором — организующим ход всего урока (дирижер оркестра).

Работа проекта, подобного проекту «Мегакласс», возможна при следующих условиях (с точки зрения основного координатора всей работы):

1. Наличие координаторов со всех сторон.
2. Наличие общего координирующего центра, объединяющего деятельность участников проекта.
3. Заинтересованность в эксперименте всех участников проекта на всех уровнях, готовность принимать на себя часть работы по подготовке очередного урока.
4. Возможность еженедельных встреч для планирования

следующего урока у всех участников проекта.

5. Достаточный канал линий связи, позволяющий поддерживать одновременно и видеоконференцсвязь, и массовое подключение к внешним сервисам.

С точки зрения учителя в школе:

1. Наличие основного координатора всей работы, регулирующего процесс подготовки к занятиям, разработки кейсов и заданий: процесс распределения задач по подготовке урока между участниками кластера.

2. Наличие инструкции для организации учебной деятельности, а также дидактических и методических материалов, в разработке которых учитель принимает непосредственное участие.

3. Достаточный канал линий связи, позволяющий поддерживать одновременно и видеоконференцсвязь, и массовое подключение к внешним сервисам.

4. Возможность постоянного контакта с учениками своей школы для своевременной и качественной организации учебного процесса в условиях мегакласса.

С точки зрения привлекаемых тьюторов:

1. Наличие основного координатора всей работы.

2. Доступ к материалам урока и техническому заданию выполняемых учащимися проектов.

3. Наличие контакта с участниками образовательного процесса, в частности, с группой, к которой он прикреплен.

4. Способ поддержки контакта с группой учащихся: соцсети, телефонные приложения и т.д.

С точки зрения экспертов со стороны:

1. Наличие основного координатора всей работы.

2. Владение информацией о регламенте урока, о тематике, и системе оценивания деятельности учащихся.

3. Предварительное согласование своей деятельности с учителями и

модератором: от содержания лекционного фрагмента и диапазона тематик для дополнительных вопросов до критериев оценивания всех видов деятельности.

4. Достаточный канал линий связи, позволяющий поддерживать одновременно и видеоконференцсвязь, и массовое подключение к внешним сервисам.

Процесс подготовки мега-урока проходит в несколько этапов:

1. Согласование темы и целей урока.
2. Поиск и отбор содержания, максимально соответствующего современному уровню развития предметной области.
3. Определение способов взаимодействия участников в сетевом формате.
4. Выбор или разработка методов и приёмов обучения, применимых на мега-уроке, согласно целям и содержания темы.
5. Распределение ролей и согласование регламента проведения урока.
6. Подготовка и согласование дидактических и методических материалов.

При планировании проектной работы по программированию к перечисленным этапам добавится:

Для учителя:

1. Подготовка и обсуждение ТЗ с учениками. Оформление кейса.
2. Определение графика работ, распределение ролей
3. Подготовка списка литературы и различных пособий для помощи в выполнении проекта.
4. Привлечение экспертов для оценки проекта.

Все этапы подготовки могут проводиться как в виде онлайн-совещаний участников, так и в оффлайн режиме в условиях совместной работы в облачных сервисах. Тем не менее, как показывает практика, проведение онлайн-совещаний не реже раза в неделю необходимо.

Основное отличие в организации и сопровождения проектов по программированию в условиях образовательной технологической платформы



«Мега-класс» в сравнении с обычной классно урочной системой состоит в непрямом контакте всех участников образовательного процесса, следовательно, следует рассмотреть все этапы подготовки и поддержки проектов более тщательно, и наметить пути решения возможных проблем.

### **2.1.2. Условия организации и сопровождения проектов по программированию в мега-классе**

В обучении программированию применение метода проектов приобретает особую важность. Решение глобальных задач информатизации предполагает создание столь же объемных программных проектов, обладающих высоким уровнем системности и разрабатываемых группами программистов [27].

Конечно, не все ученики готовы к такого рода проектам, в этой проблеме нам и может помочь образовательный кластер. Во-первых, привлечение экспертов со стороны в роли консультантов и оценщиков не может не отразиться на интересе, и как следствие, мотивации учащихся. Поработать в команде над проектом как «настоящие» программисты, да еще и если вас курирует практикующий опытный программист, вот это уже по-настоящему интересный опыт, не то что выполнение заданий в классе. Во-вторых, не малозначимым плюсом будет являться взаимодействие с учениками из других городов и школ, с совсем другим опытом и знаниями, обретение единомышленников и работа в распределенной команде, удаленно, как и работают сейчас большинство фрилансеров - программистов.

#### **Этапы работы над программным проектом:**

##### **1. Разработка технического задания (ТЗ).**

ТЗ — обязательный документ при создании любого продукта

Миф о том, что для простых проектов оформление ТЗ не нужно, не выдерживает критики. Нужно! И для простого, и для сложного. Просто для простых проектов пишите простое ТЗ, а для сложных — сложное, если вы

верите, что для ТЗ такая градация существует в принципе.

ТЗ в общем случае обозначает требования к продукту, но не требования к тому, как этот продукт будет делаться. Это не касается тех случаев, когда метод неотделим от продукта. Автор ТЗ не имеет права навязывать разработчикам решение хотя бы потому, что он не может быть компетентен во всех вопросах.

Хорошее техническое задание:

- должно полностью описывать результат работ,
- должно быть максимально простым и понятным,
- должно быть наглядным,
- не должно быть излишне длинным,
- его должно быть интересно читать и делать.

ТЗ разрабатывается после этапа дизайна, но до начала верстки. Если делать ТЗ до дизайна, то останете вносить изменения в этот документ, и потратите кучу сил, времени и прочее. Техническое задание не обязательно должно диктовать требования к дизайну, а фиксировать дизайн, т.е. должно быть концептуальное описание, функциональное описание и прототип [15]. Поэтому так важно обсуждать задание заранее с учениками, вносить правки, и уже потом давать на разработку. Добивайтесь четкого понимания ТЗ любыми силами.

*На примере ТЗ на разработку мобильного приложения, покажем, из каких разделов оно может состоять:*

1. Терминология. Очень важно договориться о терминах заранее, ведь одно и то же слово вы (учитель-заказчик) и разработчик (ученик) можете понимать по-разному.

2. Цель создания системы, то есть приложения. Здесь вы подробно описываете, что пользователь сможет делать в приложении, какие действия и какой результат он получает.

3. Требования к приложению. Важно понимать, что каждый шаг будущего пользователя должен быть продуман до технических мелочей. К

примеру, из ТЗ должно быть четко ясно, что случится, если нажать на вот эту красную кнопку.

4. Сценарии использования. Что происходит, когда человек впервые заходит в приложение? Нужно ли регистрироваться? А что произойдет, когда он зайдет повторно? Все эти пути важно описать в ТЗ, ведь это поможет выявить, какие функции нужны приложению и что именно должно в нем происходить.

5. Описание экранов. Некоторые ТЗ содержат прототипы экранов, если они уже готовы. В некоторых есть даже первичный дизайн. В любом случае описать экраны хотя бы словами просто необходимо. Ведь для многих пользователей приложение — это и есть экран.

6. Требования к платформе, CMS, архитектура системы... и еще много страшных слов, если вы не разработчик или технический писатель.

Ваше ТЗ может не содержать тех или иных разделов только в том случае, если это не существенно именно для вашего проекта, например, описание экранов не нужно для консольного приложения, но даже для него общее описание текстового интерфейса следует сделать.

**2. Определение общего алгоритма решения и выделение подзадач: дизайн интерфейса, алгоритмические подзадачи, общая сборка.**

После того как ТЗ составлено, и учащиеся уже понимают, что от них требуется, следует подумать, а как можно, собственно, реализовать то что требуется. Если с интерфейсом все понятно после описания дизайна в ТЗ, то над алгоритмом функций, которые выполняет приложение (программа) нужно еще подумать. Если проект сложный, и включает в себя несколько этапов и алгоритмов решения, то можно в самом ТЗ разбить процесс общего решения на этапы и подзадачи. Также, если проект выполняют несколько человек нужно организовать распределение работы между ними, и договориться на каком этапе проводить общую сборку.

**3. Распределение работ, выполнение и консультации.**

Важной особенностью обучения в образовательных кластерах является

непрямой контакт участников образовательного процесса, а значит, напрашивается проблема поддержки и сопровождения проектов в виртуальной среде. А именно:

- как организовать взаимодействие учащихся, учителей, и организаторов платформы;
- какие материалы использовать для помощи в освоении материала и/или выполнению самостоятельных работ;
- как распланировать работу групп учеников с разным уровнем подготовки;
- как оценивать готовность как на каждом этапе выполнении проекта, так и в целом.

### ***Подбор команд и оценка их деятельности.***

Изначально, чтобы не было затруднений в этом плане нужно стараться как можно качественнее распределить учеников по уровню подготовки. Конечно, идеала все равно не добиться, поэтому мы советуем предоставить как можно больше средств по самообучению. Списки книг, статей, видео уроки, онлайн тренажеры, и др. Но, если получится так, что в одной группе окажутся учащиеся с абсолютно разным уровнем подготовки, только самостоятельной работой тут не обойдешься. За учениками, у которых особые проблемы, следует закрепить тьютора, проводить дополнительные занятия, и также договориться о том, чтобы учитель из его школы постарался уделять ему больше времени.

Так как работа над проектом требует большого количества времени невозможно организовать работу над ним лишь на уроках. Напрашивается проблема: как можно наблюдать за процессом, отслеживать этапы выполнения проекта. Можно предложить множество вариантов, но самый эффективный, по нашему мнению, это *кейс по разбору проекта*. Нужно заранее создать документ, описывающий все этапы разработки проекта с пустыми полями для заполнения, и соответственно, сроками выполнения. Организаторам только следует вовремя проверять кейсы каждой группы, и оценивать их на каждом

этапе. И поставить условие, что итоговая оценка за проект будет зависеть от оценок на каждом этапе.

Очень важно распределить учеников из нескольких школ в группы по уровню знаний и интересам. Если организовать атмосферу работы в команде, над одним заданием, как команды программистов, это не может не отразиться на результативности. И темы, предоставленные на выбор для проектной деятельности, должны быть как можно более разноплановыми, расширяемыми и практико-ориентированными, чтобы каждый мог выбрать то что ему нравится.

Коллективная работа в сети упрощает контроль над выполнением этапов проекта, ведь создать общий документ (папку) в облаке, куда ученики будут помещать результаты своей деятельности, и наблюдать в реальном времени намного проще, чем бегать за учеником и просить, чтобы он пришел и хоть что-то тебе показал.

### ***Организация взаимодействия.***

В интернете очень много средств взаимодействия, учебников, методик, виртуальных лабораторий и т.п.. Наша задача отобрать эффективные, в первую очередь нацеленные на программирование, классифицировать, и объединить в систему все вышеперечисленное для достижения поставленных ранее целей обучения.

**GitHub** — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Сервис абсолютно бесплатен для проектов с открытым исходным кодом и предоставляет им все возможности, а для частных проектов предлагаются различные платные тарифные планы. Слоган сервиса — «Social Coding» — на русский можно перевести как «Пишем код вместе» (см. рис.1).

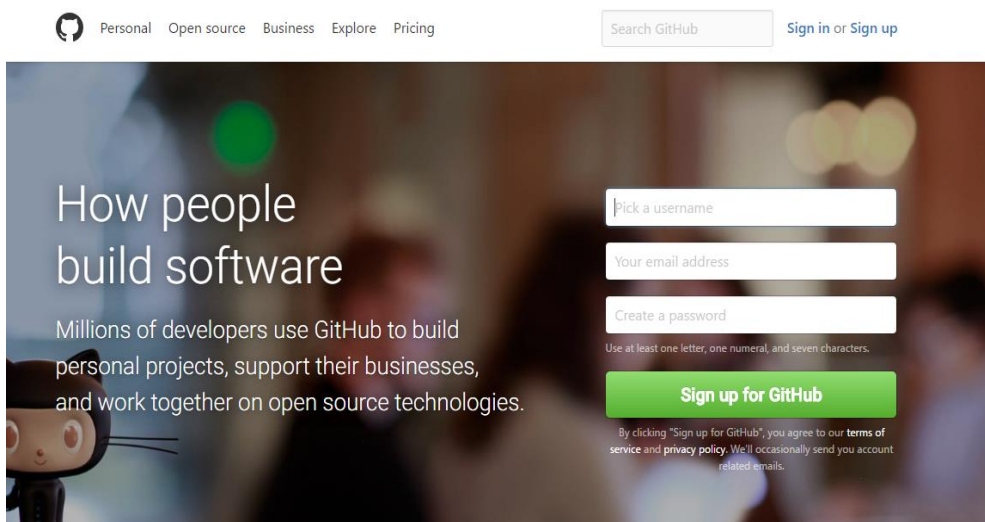


Рис.1. Главная страница веб-сервиса GitHub

Создатели сайта называют GitHub «социальной сетью для разработчиков». Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. С помощью широких возможностей Git программисты могут объединять свои репозитории — GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева. Фактически git - это система контроля версий, встраиваемая в практически все современные программные среды [31].

Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок. Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

А также:

- На платных тарифных планах можно создавать приватные репозитории, доступные ограниченному кругу пользователей.
- Есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса.
- Код проектов можно не только скопировать через Git, но и скачать в виде обычных архивов с сайта. (Для этого достаточно добавить /zipball/master/ в конец адресной строки.)
- Кроме Git, сервис поддерживает получение и редактирование кода через SVN и Mercurial.

- На сайте есть pastebin-сервис [gist.github.com](https://gist.github.com) для быстрой публикации фрагментов кода.

Стоит только зарегистрироваться и огромный набор бесплатных возможностей вам обеспечен. Единственный сайт не русифицирован.

**Сервис RealtimeBoard** - это привычная всем маркерная доска, расположенная в интернете, и не имеющая границ (см. рис. 2).

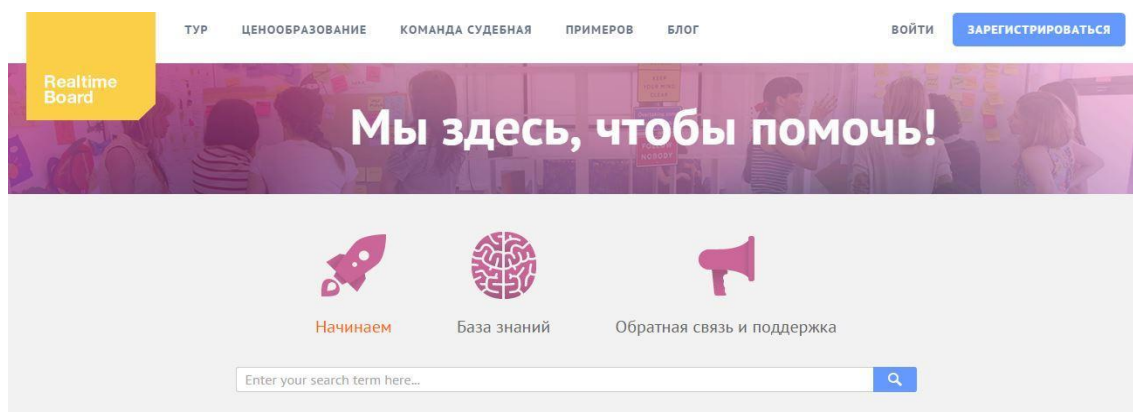


Рис.2. Начальная страница сайта [realttimeboard.com](https://realttimeboard.com)

Можно использовать как для записи идей, мыслей, так и для работы над своими проектами с командой из любой точки мира.

Сервис сочетает творческое мышление и технологические знания с целью предоставления пользователю права создавать свои собственные доски (виртуальные рабочие области), публиковать их в Интернете, а также делиться ими с друзьями или коллегами.

Как и обычная офисная доска, она имеет множество различных применений, среди которых выделяются:

- совместное планирование и управление;
- обсуждение дизайна сайтов, макетов, и другое;
- создание интеллектуальных карт, хронологических лент, инфографики, презентаций;
- применение в образовательных проектах, интерактивных уроках и мультимедийных курсах.

В обучении может служить не только отличным наглядным средством, но и платформой для совместной продуктивной деятельности учителя и

учеников. Бесконечная белая доска предоставляет неограниченное место для ваших идей и проектов, а отличные инструменты редактирования позволяют наполнить это пространство нужным вам смыслом.

**HotPotatoes** - универсальная программа-оболочка, позволяющая любому человеку самостоятельно, не прибегая к помощи программистов, создавать интерактивные тренировочно-контролирующие упражнения в формате HTML (см. рис. 3) [30].



### Hot Potatoes Home Page

Рис.3. Начальная страница сайта hotpot.uvic.ca

Программа широко используется во всем мире для создания заданий на различных языках по различным дисциплинам.

Среди многих интерактивных тестовых программ, эту программу отличает то, что интерактивные задания возможно сохранять в архиве и использовать их без подключения к интернету. Созданные с помощью нее различного рода упражнения настолько разнообразны и увлекательны, что могут служить не только для оценки знаний учащихся, но и в качестве их индивидуальной творческой работы. В программе можно составлять всевозможные кроссворды, викторины и тесты. Есть возможность включения различных аудио и видеоматериалов, возможность создавать задания на установление соответствия и по заполнению пропусков в тексте.

Стоит отметить, что в HotPotatoes предусмотрена оценка результатов ответов в процентах: чем больше правильных ответов, тем выше процент. Также HotPotatoes предоставляет возможность распечатать уже созданные вами упражнения или использовать в качестве обычного бумажного текста.

HotPotatoes совершенно несложная в использовании программа, при этом она эффективно выполняет свои главные задачи. Скачивайте,



устанавливайте и создавайте свои собственные тренировочные задания.

**Edmodo** — виртуальная среда для обучения и создания сообществ, своеобразная социальная сеть (см. рис. 4).

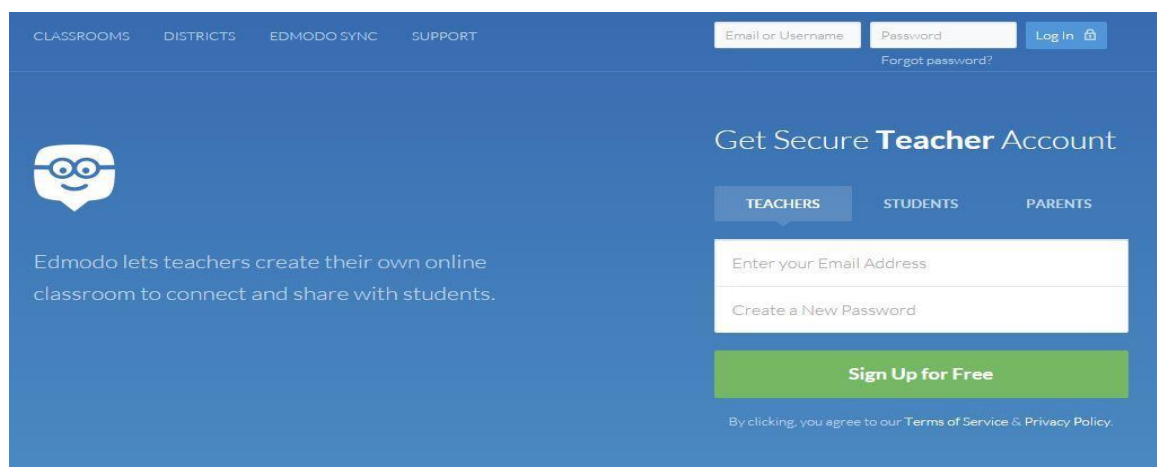


Рис.4. Начальная страница сайта edmodo.com

С помощью этого сервиса есть возможность организовать дистанционное обучение. Каждый класс является закрытой площадкой для общения, создания библиотек документов и ссылок, ведения оценивания, отслеживания статистики успеваемости, выполнения различных заданий. Можно создавать внутри класса мини-группы для обсуждения вопросов и взаимодействия.

Учитель имеет возможность создать группу, получить код на неё, и все обладатели кода (ученики) могут вступать в эту группу. Есть возможность импортировать ленты с сайтов, вкладывать рисунки или же видео. Есть календарь для фиксации расписания зачетов. Так же имеется возможность ввести задание с датой выполнения (например - записать домашнее задание), и раздел для выставления оценок.

Отличия Edmodo от классических социальных сетей:

- Нет внешней рекламы, отвлекающего контента.
- Регистрация очень упрощена, в анкетах не просится указывать дни рождения, точные фамилию, имя отчество и прочую идентифицирующую информацию.
- Пользователи делятся на три группы: учителя, учащиеся, родители.

- Никакой персональной информации по учащимся от школы не требуется. В соглашении о персональной информации точно указывается, какая загруженная на сайт информация может обрабатываться и как.

- Полностью бесплатен [28].

Для пользования этим сервисом больше всего подходят ученики старших классов. И, совсем не обязательно, что это будет дистанционное обучение. Можно просто создать среду для активного взаимодействия учителя с учениками помимо школы, например, для того же контроля знаний.

### ***Учебные материалы, используемые в организации самостоятельной деятельности.***

Правильное визуальное сопровождение процесса разработки или объяснения алгоритма будет способствовать глубокому усвоению учебного материала и пониманию сущности определённой программы.

Для динамической визуализации основных алгоритмических конструкций можно использовать анимационные ролики, построенные, например, в программе Macromedia Flash (см. рис. 5).



Рис. 5. Визуализация цикла с предусловием

Также для визуализации алгоритмов можно использовать **дополнительные библиотеки изучаемого языка программирования**, в C/C++, например, это OpenGL. Такой подход позволяет применить библиотеку в практике изучения языка с целью включения учащихся в процесс разработки визуализации алгоритмов вместе с изучением языка программирования.

Применение этого способа мы предлагаем организовать следующим

образом: на примере визуализации построения фрактала можно наглядно продемонстрировать и закрепить основные понятия рекурсии. В результате проделанной работы перед нами предстает программа, которая строит фрактал итеративным методом. Количество уровней вложенности определяет сам пользователь, нажимая на кнопку «+» для увеличения, и «-» для уменьшения их количества [9].

Модель динамична, показывает нам процесс работы рекурсивного алгоритма, тем самым повышает шансы его усвоить. (см. рис. 6).



Рис.6. Результат визуализации фрактальной фигуры, треугольник Серпинского

Или, например, для визуализации сортировки массива методом пузырька [3], каждый элемент массива представим в виде закрашенных квадратов (чем больше значение у элемента, тем темнее будет цвет). В ходе сортировки квадраты будут меняться местами, и в итоге у нас получится линия из выстроенных квадратов, от самого светлого к самому темному, либо наоборот (см. рис.7).

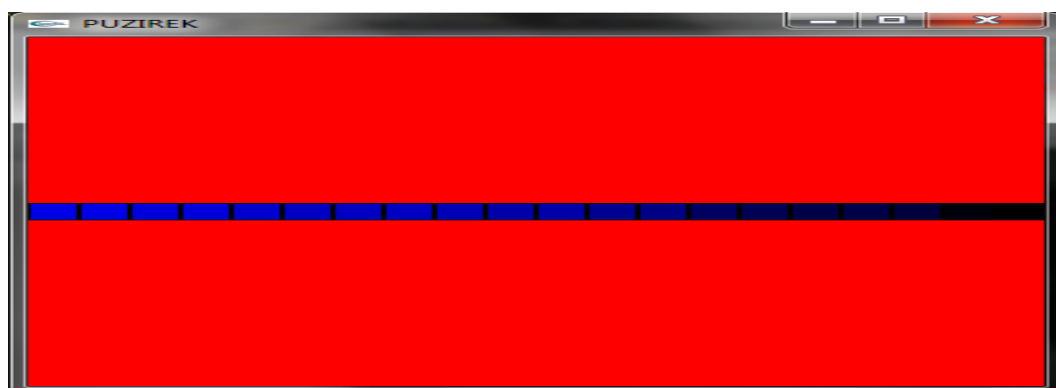


Рис. 7. Результат визуализации сортировки методом пузырька.

После визуализации процесса работы алгоритма уже легче будет представить его в более формальном виде, разбив алгоритм на конкретные

шаги.

**LearningApps** является очень интересным приложением Web 2.0 для поддержки процесса преподавания с помощью интерактивных модулей (см. рис. 8).

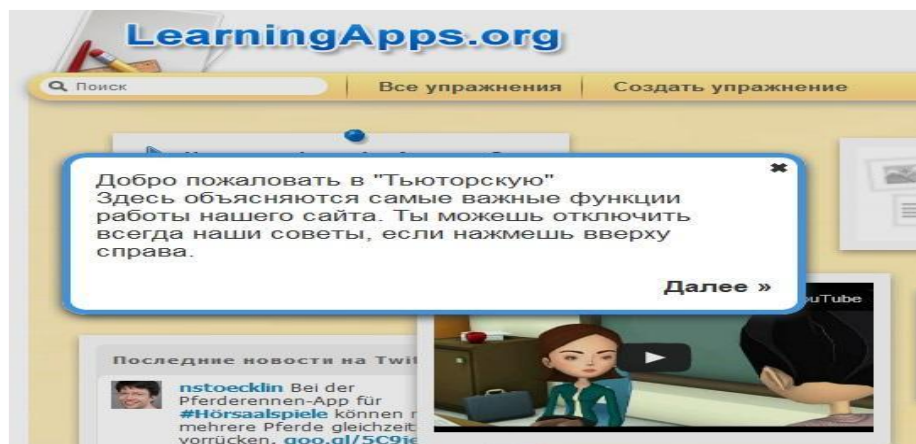


Рис.8. «Тьюторская» приложения LearningApps

На сайте даны все указания по работе, подсказки, в общем, весь материал для подготовки приложений. Можно создать свое упражнение, а можно сделать подобное из вкладки «все упражнения». Упражнения не однотипные, они подразделяются на множество видов. Если захотите ознакомиться со всем этим разнообразием - заходите на сайт. Если же хотите сами попробовать создать нечто подобное, дополнительно нужно будет пройти несложную регистрацию.

Как вы понимаете, конкретно для обучения информатики — это приложение предлагает огромные возможности, особенно для тренировки навыков, или знаний теоретического материала. Например, если предложить ребятам вместо теста по закреплению знаний основных алгоритмических конструкций поиграть в игру «Кто хочет стать миллионером», вряд ли вы столкнетесь с неодобрением.

**Редактор Prezi** — это впечатляющая разработка компании Prezi Inc. Сервис предлагает удивительные возможности для создания многообразных мультимедийных документов.

Работа веб-сервиса Prezi.com основана на технологии масштабирования (приближения и отдаления объектов). В отличие от «классической»

презентации, в Prezi большинство эффектов основано не с переходом от слайда к слайду, а с увеличением отдельных частей этого слайда [25].

Нельзя не выделить единственный минус - язык интерфейса сервиса английский. Но вот текст в презентациях может быть на любом языке, в том числе на русском.

Кроме создания своих собственных презентаций, опция Add people предполагает добавление любого количества пользователей для совместного редактирования. Так что помимо создания необычных, привлекающих внимание презентаций к урокам, вы можете предложить ученикам совместно с вами поработать над их разработкой, что позволит уже заранее вовлечь ребят в изучение темы.

Для учеников, только начинающих изучать программирование, можно использовать среду **Blockly**. Эта среда опирается на использование наглядно-деятельностного уровня мышления, которое в принципе выражено у всех учащихся. Если у человека имеются проблемы в переводе конструкций алгоритма на язык программирования ему может помочь Blockly.

Пример: построение квадрата с помощью блоков в среде Blockly (см. рис.9).

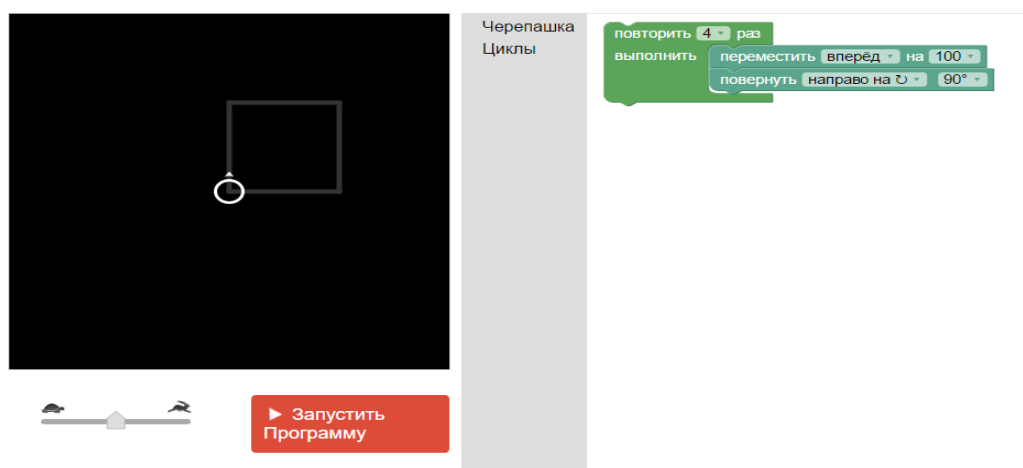


Рис.9. Вид программы построения квадрата в среде Blockly

Важная особенность графической среды: освобождение от заботы о синтаксисе, что позволяет начинающему сосредоточиться на логике программы. Можно интегрировать редактор в собственные веб-приложения, чтобы дать возможность начинающим пользователям создать свой интерфейс.

Созданную с помощью этого редактора программу перевести на код JavaScript, Python, PHP, Dart, или какой-либо другой язык: Приложение может выполнить код по желанию. Blockly поставляется с большим количеством predefined блоков. Все, начиная от математических функций цикла структуры [4].

В сети можно найти достаточное количество примеров и других языков и их редакторов. Плюсы понятны: при программировании с использованием визуальных объектов человек описывается на образы, обучение активизирует различные уровни мышления. Но минус - может произойти торможение в развитии программирования, в человеческой истории все языки, опирающиеся на иероглифы, приводили к дополнительным сложностям в обучении [24].

**Объектно-ориентированная среда Alice** позволяет управлять трехмерными объектами, создавать программы (компьютерные видеоигры и анимационные приложения-фильмы), генерирующие анимацию в виртуальных мирах на основе использования условий, циклов, функций/методов, обработки массивов данных, списков, объектов, наследования, инкапсуляции, полиморфизма и т.д.

В отличие от распространенного в обучении программированию подхода, когда в первую очередь знакомятся с понятием данных и алгоритмов, Alice использует объектно-ориентированное программирование изначально. Проведенные создателями проекта исследования показали, что это способствует более глубокому пониманию концепции объектно-ориентированное программирования — ведь если обучаемые сразу начинают оперировать объектными понятиями, то им нет необходимости переучиваться, что неизбежно происходит при переходе от одной парадигмы программирования к другой.

Среда программирования Alice выпускается для Windows, Mac OS и Linux в двух редакциях: основной (для вузов) и упрощенной (для школ). В среде имеется большая библиотека объемных объектов из реального мира (природа, животные, инструменты быта и т.д.). Их можно двигать, вращать,

менять цвет и размер при помощи мыши, а на основе полученного виртуального мира программно описывать анимацию и создавать игровые модели (см. рис. 10).

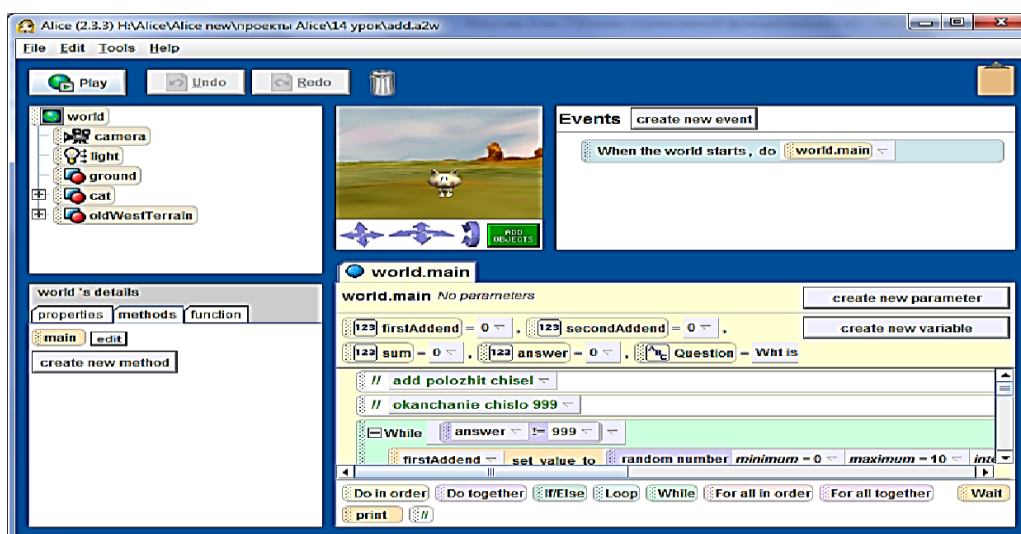


Рис.10. Окно программы в среде Alice

В среде Alice используется собственный встроенный язык программирования, приближенный к синтаксису языков современных объектно-ориентированных языков программирования таких, как Java, C++ или Visual Basic. Поскольку программное обеспечение Alice позволяет создавать только синтаксически правильные команды, то программирование в данной среде сводится лишь к разработке и реализации соответствующих алгоритмов. Учащемуся не нужно запоминать синтаксис какой-либо конструкции, он всегда может воспользоваться набором имеющихся процедур и функций рассматриваемого объекта или всплывающими подсказками. Именно это позволяет школьникам в дальнейшем сконцентрировать свое внимание на сценарии игры, сцены, используемых объектах, их свойствах и методах, а не беспокоиться о синтаксических ошибках своих приложений [12].

#### 4. Сборка проекта и защита.

Как было сказано выше, нужно заранее договориться с учениками на каком этапе работы проводить полную сборку проекта. Важно, чтобы все могли посмотреть и оценить работы друг друга. Нужно избегать таких ситуаций, в которых ученику покажется что его труды были напрасны. Позитивная оценка очень важна в подростковом возрасте, в основном, как

способ самоутверждения. Даже если проект в целом не так хорош, важно найти в нем что-то уникальное, интересное, и выставить это на всеобщее обозрение. Видеть проекты других и сравнивать со своей работой тоже немаловажно. Особенно, если учеников поставить в роль оценщиков. Просмотрев множество других работ, он уже сам будет видеть, где что должно присутствовать, и что было важно не упустить при работе над проектом. И потом уже также оценивая, как бы со стороны, сможет посмотреть на свой проект и увидеть недочеты.

Так как на уроке вряд ли можно успеть показать все, можно устроить что-то вроде виртуальной галереи. Можно сделать сайт, или просто выложить итоговые проекты в онлайн хранилище и обеспечить всем доступ. А на уроке можно выбрать и показать лучшие, и обязательно устроить обсуждение.

Если есть возможность, можно разбить учеников на мини группы по схожим тематикам, языкам исполнения и др. и устроить защиту в группах. А уже потом устроить презентацию лучших проектов, по мнению самих учеников.

Также можно устроить видео презентации своих проектов, регламентировать их по времени, а уже после просмотра можно будет задать ученику некоторые вопросы. Такой способ может помочь сократить время защиты, ведь в открытой презентации можно что-то забыть, потратить время на раздаточный материал, и т.д.

Можно организовать что-то вроде интерактивного семинара, когда выступление основывается на диалоге с аудиторией. Например, для защиты проекта в ходе которого было разработано мобильное приложение, можно сделать что-то вроде мультимедийного макета, с областью навигации, разветвлениями презентации, последовательность защиты будет зависеть от того, какие ответы дают слушатели, какие задают вопросы и т.д.

В общем, конечно, форма защиты очень зависит от тематики. Для большинства проектов по программированию подойдут все вышеперечисленные формы и способы, но также можно придумать и свои.



## **2.2 Комплекс проектных заданий по программированию**

### **1. «Converter» - калькулятор, предназначенный для перевода числа из одной системы счисления в другую (от 2-ной до 36-ной)**

Проект носит учебный характер и может служить для закрепления навыков перевода чисел из одной системы в другую и способствует пониманию алгоритмической основы перевода.

#### ***Для реализации необходимо:***

- знание алгоритмов перевода натурального числа из произвольной системы счисления (СС) в любую другую СС;
- наличие навыков программирования на каком-либо языке программирования (в случае выбора соответствующего уровня сложности, предполагающего разработку мобильного приложения – Java).

#### ***Уровни сложности:***

1. Программа переводит числа из двоичной СС в десятичную и обратно.
2. Программа переводит числа из десятичной системы в любую другую.
3. Программа переводит числа из произвольной системы счисления в любую другую СС (должно участвовать не менее 3-х СС).
4. Мобильное приложение для любого из описанных выше вариантов.

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 1-2 человека, 2 человека только в случае разработки мобильного приложения. Тогда один ученик будет заниматься дизайном и разработкой мобильного приложения, а второй алгоритмом выполнения самого задания.

Для выполнения задания 2-го уровня уже может понадобиться больше учеников. Можно распределить учеников по алгоритмам в зависимости от

количества выбранных СС. И также расширить количество участников если решат разрабатывать мобильное приложение. Для выполнения задания 3-го уровня способ распределения учеников по подзадачам можно оставить тем же.

От тьютора требуется периодически наблюдать за процессом и помогать в случае затруднений. Если он будет не в состоянии решить некоторые вопросы, всегда должен быть человек, который может, тьютору предстоит лишь направлять учеников к нему.

### ***Необходимый материал для начала и продолжения работы:***

Так как выбор учениками языка исполнения, сред разработки и т.д. мы не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков программирования и не только. Для данного проекта это могут быть:

#### Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2009.

Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му. Язык программирования С++. Базовый курс. 5-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2014.

Шилдт Г. Полный справочник по С++. 4-е изд.: Пер. с англ. — М.: Издательский дом "Вильямс", 2006.

Лутц М. Изучаем Python, 4-е издание. — Пер. с англ. — СПб: СимволПлюс, 2010.

Бизли Д. Python. Подробный справочник, 4-е издание. Издательство “Символ-плюс”, 2010.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб: СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.  
Том 1. Основные алгоритмы.

Том 2. Получисленные алгоритмы.

Том 3. Сортировка и поиск.

Том 4, А. Комбинаторные алгоритмы.

#### Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Самое простое руководство по иконографии:  
<https://habrahabr.ru/company/netologyru/blog/331050/>

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Некоторые среды программирования: Blockly, Alice.

Веб-сервис для хостинга: GitHub.

#### Интерактивные тренажеры, упражнения:

Упражнение на определение базовых алгоритмических конструкций:  
<https://learningapps.org/1567206>.

Упражнение по переводу чисел из двоичной СС в десятичную:  
<https://learningapps.org/442997>.

Системы счисления, тренажер по переводу из любой в любую (заполнить пропуски в таблице): <https://learningapps.org/1776378>.

Тренажер “Знаете ли вы Pascal”: <https://learningapps.org/2887689>.

Основы программирования на Pascal: <https://learningapps.org/3035365>.

Упражнения по программированию на C++: [http://ci-plus-plus-snachala.ru/?page\\_id=6](http://ci-plus-plus-snachala.ru/?page_id=6)

**Сроки выполнения (по уровням):**

1. 1,5 недели.
2. 2,5 месяц.
3. 1 месяц.
4. К каждому из вышеперечисленных прибавить 1 неделю.

В предыдущем параграфе было сказано о целесообразности использования кейса с техническим заданием для отслеживания этапов выполнения проекта. Приведём пример оформления такого кейса для проекта первого уровня сложности (см. таблицу 1).

Таблица 1

Пример оформления кейса для проекта 1-го уровня сложности

**КЕЙС ПО РАЗБОРУ ПРОЕКТА**

**1-этап. Постановка задачи**

Разработка калькулятора для перевода чисел из двоичной системы счисления в десятичную.

**2-этап. Математическая модель программы**

Перевод из любой системы счисления в десятичную основывается на свойствах позиционных систем счисления, где положение цифры в числе определяет ее разряд, т.е. для десятичной системы счисления  $152 = 100+50+2$ .

Если взять степени основания системы счисления, то пример можно преобразовать:

$$152 = 100+50+2 = 10^2+5*10^1+2*10^0.$$

То же самое верно для позиционной системы счисления **с любым основанием**. Если записать число через степени основания, а потом произвести вычисления привычным способом (как будто мы в десятичной системе), то мы и получим десятичное число, соответствующее искомому.

Например, для двоичной системы:

$$10011_2 = 1*2^4+0*2^3+0*2^2+1*2^1+1*2^0 = 16_{10} + 0_{10} + 0_{10} + 2_{10} + 1_{10} = 19_{10}$$

Например, для пятеричной системы:

$$3402_5 = 3 \cdot 5^3 + 4 \cdot 5^2 + 0 \cdot 5^1 + 2 \cdot 5^0 = 375_{10} + 100_{10} + 0_{10} + 2_{10} = 477_{10}$$

### **3-этап. Построение интерфейса приложения и управления приложением**

3.1 Каким вы представляете интерфейс приложения калькулятора? (отображение на экране монитора различных надписей, кнопок, сообщений и т.п.). Нарисуйте схематичное изображение.

3.2 Опишите элементы интерфейса (по своему схематичному изображению), их назначение и свойства.

3.3 Опишите управление приложением (сценарий, например, при выводе такого-то сообщения, должно делаться следующее..., при нажатии на такой-то элемент будет происходить...).

### **4-этап. Выработка алгоритма**

- При открытии калькулятора на экран выводится:

- Вводится ...

Словесное описание алгоритма перевода чисел из двоичной системы счисления в десятичную:

Шаг 1.

Шаг 2.

Шаг 3.

Блок-схема алгоритма перевода чисел из двоичной системы счисления в десятичную\* (не обязательно, дополнительные баллы, допустимо вместо словесного описания):

**5-этап. Реализация.**

5.1. Опишите ваш будущий проект: уровень сложности, который вы выбрали; язык на котором будете реализовывать проект; если решили создать мобильное приложение, зарисуйте интерфейс.

5.2. Вставьте код разработанной программы, скриншоты интерфейса.

**2. «Goal» - электронный дневник для постановки и отслеживания целей.**

Проект предназначен для сильных учащихся с высоким уровнем подготовки. В школьный курс подготовка до данного уровня не входит, данное задание можно предложить группе энтузиастов. Требуется опытный консультант для ведения проекта.

*Для реализации необходимо:*

- наличие навыков программирования на языке Java;
- навыки работы с MySQL для работы с базой данных;
- навыки установки и настройки связки Nginx + Apache или др. для реализации серверной части.

**Уровни сложности:**

1. Разработать мобильное приложение — электронный дневник. Оформить цели в виде таблицы, и организовать возможность отмечать их выполнение по дням недели (см. рис. 11).

	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Сделать зарядку							
Обновить свой блог							
Сходить в тренажерный зал							

Рис. 11. Пример оформления мобильного приложения

2. Добавить возможность просмотра диаграммы выполнения целей (сколько процентов из поставленных целей выполнено).

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 4 человека. Один будет заниматься дизайном приложения, два человека будут работать над алгоритмом выполнения задания, а четвертый займется хранением данных. Для выполнения задания второго уровня нужно добавить еще двух человек, на сбор данных для диаграммы и ее отрисовки.

Сопровождать проект данного уровня должен эксперт со стороны, с которым у учеников должен быть налажен постоянный контакт. Так как далеко не каждый учитель или студент (тьютор) может помочь в выполнении данного задания.

**Необходимый материал для начала и продолжения работы:**

Так как выбор учениками языка исполнения, сред разработки и т.д. мы

не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков программирования и не только. Для данного проекта это могут быть:

Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб: СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.  
Том 1. Основные алгоритмы.  
Том 2. Получисленные алгоритмы.  
Том 3. Сортировка и поиск.  
Том 4, А. Комбинаторные алгоритмы.

Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Самое простое руководство по иконографии:  
<https://habrahabr.ru/company/netologyru/blog/331050/>

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Веб-сервис для хостинга: GitHub.

***Сроки выполнения (по уровням):***

1. 2 месяца.



2. 3 месяца.

### **3. «MatrixCalculator» - матричный калькулятор.**

Проект носит учебный и межпредметный характер и может служить для изучения и закрепления навыков выполнения основных действий с матрицами. Однако данный материал не входит в курс основной школы, поэтому проект можно предложить заинтересованным учащимся как задание, которое полностью подразумевает самостоятельное обучение с консультационной помощью.

#### ***Для реализации необходимо:***

- знание алгоритмов выполнения элементарных операций с матрицами;
- наличие навыков программирования на каком-либо языке программирования (в случае выбора соответствующего уровня сложности, предполагающего разработку мобильного приложения – Java).

#### ***Уровни сложности:***

1. Программа складывает и вычитает матрицы любой размерности.
2. Добавляется возведение матрицы в степень и ее транспонирование.
3. Добавляется получение определителя матрицы, и вычисление обратной матрицы.
4. Мобильное приложение любого описанных выше вариантов.

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 1-2 человека, 2 человека только в случае разработки мобильного приложения. Тогда один ученик будет заниматься дизайном и разработкой мобильного приложения, а второй алгоритмом выполнения самого задания.

Для выполнения задания 2-го уровня уже может понадобиться больше учеников. Можно распределить учеников по алгоритмам в зависимости от количества их количества. И также расширить количество участников если решат разрабатывать мобильное приложение. Для выполнения задания 3-4

уровня способ распределения учеников по подзадачам можно оставить тем же.

От тьютора требуется периодически наблюдать за процессом и помогать в случае затруднений. Если он будет не в состоянии решить некоторые вопросы, всегда должен быть человек, который может, тьютору предстоит лишь направлять учеников к нему. Также может понадобиться консультант по математике.

### ***Необходимый материал для начала и продолжения работы:***

Так как выбор учениками языка исполнения, сред разработки и т.д. мы не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков программирования и не только. Для данного проекта это могут быть:

#### Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Керниган, Брайан У., Ритчи, Деннис М. Язык программирования C, 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2009.

Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му. Язык программирования C++. Базовый курс. 5-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2014.

Шилдт Г. Полный справочник по C++. 4-е изд.: Пер. с англ. — М.: Издательский дом "Вильямс", 2006.

Лутц М. Изучаем Python, 4-е издание. — Пер. с англ. — СПб: СимволПлюс, 2010.

Бизли Д. Python. Подробный справочник, 4-е издание. Издательство “Символ-плюс”, 2010.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic

Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб: СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.  
Том 1. Основные алгоритмы.

Том 2. Получисленные алгоритмы.

Том 3. Сортировка и поиск.

Том 4, А. Комбинаторные алгоритмы.

#### Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Операции над матрицами, видео урок:  
<https://www.youtube.com/watch?v=mSEvJfD1om8>.

Самое простое руководство по иконографии:  
<https://habrahabr.ru/company/netologyru/blog/331050/>

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Некоторые среды программирования: Blockly, Alice.

Веб-сервис для хостинга: GitHub.

#### Интерактивные тренажеры, упражнения:

Упражнение на определение базовых алгоритмических конструкций:  
<https://learningapps.org/1567206>.

Тренажер “Знаете ли вы Pascal”: <https://learningapps.org/2887689>.

Основы программирования на Pascal: <https://learningapps.org/3035365>.

Упражнения по программированию на C++: [http://ci-plus-plus-snachala.ru/?page\\_id=6](http://ci-plus-plus-snachala.ru/?page_id=6)

#### ***Сроки выполнения (по уровням):***

1. 1,5 недели.
2. 2,5 месяц.

3. 1 месяц.
4. К каждому из вышеперечисленных прибавить 1 неделю.

#### **4. «Quest» - игра жанра «текстовая RPG».**

Проект хоть и имеет игровой характер, все же несет массу положительных результатов обучения, начиная от мотивации (какому ученику не захочется написать свою «игрушку»), и заканчивая углубленными теоретическими и практическими знаниями. При создании собственной игры возникает множество алгоритмических задач, в данном случае: от вопросов хранения информации, до решения проблемы передвижения персонажа в игровом поле с обходом препятствия (теория графов и матричные алгоритмы). Крайне важен дизайн и поддержка интереса пользователя.

##### ***Для реализации необходимо:***

- наличие навыков программирования на языке Java;
- навыки работы с MySQL для работы с базой данных (при выборе усложненного варианта технического задания);
- навыки установки и настройки связки Nginx + Apache или др. для реализации серверной части (при выборе усложненного варианта технического задания).

##### ***Уровни сложности:***

1. Создать мобильное приложение-игру жанра «текстовая RPG», организованную в виде квеста, в которой реализовано:
  - Игровое поле с отображением количества здоровья, силы, основных навыков персонажа (таких как урон, магия, защита, дополнительные артефакты и их свойства), а также поле с выводом текста квеста и поле выбора действия (radiogroup).
  - В поле вывода текста описывается ситуация, в которую попал персонаж, в поле выбора действия предлагается выбор путей развития ситуации.
  - Квест считается выполненным, если достигнута цель, поставленная в начале. (например, добыть диск из пещеры тролля, перевезти

артефакты через реку).

– \*можно ориентироваться на текстовые квесты в игре «Космические рейнджеры».

2. Добавить больше навыков персонажа (для хранения данных понадобится база данных), а также функцию сохранения игры на любом этапе, и, как следствие, регистрацию персонажа в начале игры.

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 4 человека. Два человека будут заниматься дизайном приложения и отрисовкой графики, два человека будут работать над алгоритмом выполнения задания. Для выполнения задания второго уровня нужно добавить еще двух человек, для организации хранения данных.

Сопровождать проект данного уровня должен эксперт со стороны, с которым у учеников должен быть налажен постоянный контакт. Так как далеко не каждый учитель или студент (тьютор) может помочь в выполнении данного задания.

### ***Необходимый материал для начала и продолжения работы:***

Так как выбор учениками языка исполнения, сред разработки и т.д. мы не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков программирования и не только. Для данного проекта это могут быть:

#### Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб: СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.  
Том 1. Основные алгоритмы.

Том 2. Получисленные алгоритмы.

Том 3. Сортировка и поиск.

Том 4, А. Комбинаторные алгоритмы.

#### Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Самое простое руководство по иконографии:  
<https://habrahabr.ru/company/netologyru/blog/331050/>

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Веб-сервис для хостинга: GitHub.

3DG.ME – блог о 3D-графике и разработке игр:  
<http://3dg.me/ru/gamedev/basics/sozdanie-igr-dlya-nachinayushchih>

#### ***Сроки выполнения (по уровням):***

1. 2,5 месяца.
2. 3 месяца.

#### **5. «Space» - эмулятор космических объектов.**

Проект носит как учебный, так и игровой характер и может служить для закрепления некоторых знаний по физике, и графике в программировании.

#### ***Для реализации необходимо:***

- Наличие навыков программирования на каком-либо языке программирования (в случае выбора соответствующего уровня сложности, предполагающего разработку мобильного приложения – Java).
- Опыт программирования графики на любом языке программирования.

### ***Уровни сложности:***

1. Создать эмулятор космических объектов 2D, содержащий:
  - Поля ввода, где пользователь задает начальное положение планет в космосе, их массу и скорость
  - Кнопку «симуляция», при нажатии на которую система показывает, как планеты летают вокруг солнца.
  - Для расчета траектории взять формулы из физики.
2. Добавить в задание отображение цвета планет, и, как следствие, отрисовку эмулятора в цвете.
3. Мобильное приложение для любого из описанных выше вариантов.

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 4 человека. Два человека будут заниматься дизайном приложения и отрисовкой графики, два человека будут работать над алгоритмом выполнения задания. Для выполнения задания второго уровня нужно добавить еще двух человек, для отрисовки в цвете. Для третьего уровня добавление участников в группу не обязательно.

Сопровождать проект данного уровня должен эксперт со стороны, с которым у учеников должен быть налажен постоянный контакт. Так как далеко не каждый учитель или студент (тьютор) может помочь в выполнении данного задания.

### ***Необходимый материал для начала и продолжения работы:***

Так как выбор учениками языка исполнения, сред разработки и т.д. мы не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков программирования и не только. Для данного проекта это могут быть:

#### Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2009.

Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му. Язык программирования С++. Базовый курс. 5-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2014.

Шилдт Г. Полный справочник по С++. 4-е изд.: Пер. с англ. — М.: Издательский дом "Вильямс", 2006.

Лутц М. Изучаем Python, 4-е издание. — Пер. с англ. — СПб: СимволПлюс, 2010.

Бизли Д. Python. Подробный справочник, 4-е издание. Издательство “Символ-плюс”, 2010.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб: СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.  
Том 1. Основные алгоритмы.

Том 2. Получисленные алгоритмы.

Том 3. Сортировка и поиск.

Том 4, А. Комбинаторные алгоритмы.

Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Самое простое руководство по иконографии:



<https://habrahabr.ru/company/netologyru/blog/331050/>

Некоторые среды программирования: Blockly, Alice.

Веб-сервис для хостинга: GitHub.

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Интерактивные тренажеры, упражнения:

Упражнение на определение порядка планет:

<https://learningapps.org/253359>

Упражнение «Астронавигация»: <https://learningapps.org/378331>

Упражнение на определение базовых алгоритмических конструкций:

<https://learningapps.org/1567206>.

Тренажер «Знаете ли вы Pascal»: <https://learningapps.org/2887689>.

Основы программирования на Pascal: <https://learningapps.org/3035365>.

Упражнения по программированию на C++: [http://ci-plus-plus-snachala.ru/?page\\_id=6](http://ci-plus-plus-snachala.ru/?page_id=6)

***Сроки выполнения (по уровням):***

1. 2 месяца.
2. 3 месяца.
3. Сроки остаются те же.

## **6. «Assistant» - помощник для изучения программирования.**

Проект носит учебный характер и может служить для закрепления навыков программирования и способствует пониманию основных алгоритмических конструкций.

***Для реализации необходимо:***

- наличие навыков программирования на каком-либо языке программирования (в случае выбора соответствующего уровня сложности, предполагающего разработку мобильного приложения – Java).
- навыки работы с MySQL для работы с базой данных (при выборе усложненного варианта технического задания);
- навыки установки и настройки связки Nginx + Apache или др. для реализации серверной части (при выборе усложненного варианта

технического задания).

### ***Уровни сложности:***

1. Создать программу для изучения программирования, содержащую:

- Основной экран, который разбит на 3 части.
- В 1-ой части результат выполнения программы.
- Во 2-й части исходный код программы, причем строки перемешаны в

произвольном порядке.

- В 3-й части поле, куда надо переносить строки из 2-й части (в нужном порядке).

- Кнопка «проверить».

2. Добавить функции:

- Регистрация и накопление баллов.
- Дается 3 попытки для правильного упорядочивания строк программы,

иначе задание начинается заново и вычитаются баллы

3. Мобильное приложение для любого из описанных выше вариантов.

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 4 человека. Два человека будут заниматься дизайном приложения, два человека будут работать над алгоритмом выполнения задания. Для выполнения задания второго уровня нужно добавить еще двух человек, для организации хранения данных. Для третьего уровня добавление участников в группу не обязательно.

Сопровождать проект данного уровня должен эксперт со стороны, с которым у учеников должен быть налажен постоянный контакт. Так как далеко не каждый учитель или студент (тьютор) может помочь в выполнении данного задания.

### ***Необходимый материал для начала и продолжения работы:***

Так как выбор учениками языка исполнения, сред разработки и т.д. мы не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков

программирования и не только. Для данного проекта это могут быть:

Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Керниган, Брайан У., Ритчи, Деннис М. Язык программирования C, 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2009.

Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му. Язык программирования C++. Базовый курс. 5-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2014.

Шилдт Г. Полный справочник по C++. 4-е изд.: Пер. с англ. — М.: Издательский дом "Вильямс", 2006.

Лутц М. Изучаем Python, 4-е издание. — Пер. с англ. — СПб: СимволПлюс, 2010.

Бизли Д. Python. Подробный справочник, 4-е издание. Издательство “Символ-плюс”, 2010.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб: СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.  
Том 1. Основные алгоритмы.  
Том 2. Получисленные алгоритмы.  
Том 3. Сортировка и поиск.  
Том 4, А. Комбинаторные алгоритмы.

### Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Самое простое руководство по иконографии:  
<https://habrahabr.ru/company/netologyru/blog/331050/>

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Некоторые среды программирования: Blockly, Alice.

Веб-сервис для хостинга: GitHub.

### Интерактивные тренажеры, упражнения:

Упражнение на определение базовых алгоритмических конструкций:  
<https://learningapps.org/1567206>.

Тренажер “Знаете ли вы Pascal”: <https://learningapps.org/2887689>.

Основы программирования на Pascal: <https://learningapps.org/3035365>.

Упражнения по программированию на C++: [http://ci-plus-plus-snachala.ru/?page\\_id=6](http://ci-plus-plus-snachala.ru/?page_id=6)

Упражнение «Составь программу»: <https://learningapps.org/2076877>

### ***Сроки выполнения (по уровням):***

1. 2 месяца.
2. 3 месяца.
3. Сроки остаются те же.

### **7. «Language» - помощник в изучении английского языка.**

Проект носит учебный характер и может служить для закрепления знаний по английскому языку и способствует развитию навыков программирования.

### ***Для реализации необходимо:***

- наличие навыков программирования на каком-либо языке программирования (в случае выбора соответствующего уровня сложности, предполагающего разработку мобильного приложения – Java).
- навыки работы с MySQL для работы с базой данных (при выборе усложненного варианта технического задания);

- навыки установки и настройки связки Nginx + Apache или др. для реализации серверной части (при выборе усложненного варианта технического задания).

- владение английским языком на уровне Elementary.

### ***Уровни сложности:***

1. Создать программу обучения английскому, в которой реализовано:

- Различные интерфейсы для заданий

- Задания, на которые надо ответить «да» или «нет» (например, пары слов: long-длинный; power-простой)

- Задания на правописание (например, вставить недостающие буквы в слове)

- Задания на сопоставление пар слов или картинок (например: rabbit – хорек, кролик, удав)

- Выводить правильные варианты ответов после каждого курса заданий, с указанием где ошибся пользователь.

2. Добавить функции:

- Если правильных ответов меньше N% - повторить курс и вычитание баллов.

- Регистрация и накопление баллов.

3. Мобильное приложение для любого из описанных выше вариантов.

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 4 человека. Два человека будут заниматься дизайном приложения, два человека будут работать над алгоритмом выполнения задания. Для выполнения задания второго уровня нужно добавить еще двух человек, для организации хранения данных. Для третьего уровня добавление участников в группу не обязательно.

Сопровождать проект данного уровня должен эксперт со стороны, с которым у учеников должен быть налажен постоянный контакт. Так как далеко не каждый учитель или студент (тьютор) может помочь в выполнении

данного задания. Также может понадобиться консультант по английскому языку.

***Необходимый материал для начала и продолжения работы:***

Так как выбор учениками языка исполнения, сред разработки и т.д. мы не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков программирования и не только. Для данного проекта это могут быть:

Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2009.

Стенли Б. Липпман, Жози Лажойе, Барбара Э. Му. Язык программирования С++. Базовый курс. 5-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2014.

Шилдт Г. Полный справочник по С++. 4-е изд.: Пер. с англ. — М.: Издательский дом "Вильямс", 2006.

Лутц М. Изучаем Python, 4-е издание. — Пер. с англ. — СПб: СимволПлюс, 2010.

Бизли Д. Python. Подробный справочник, 4-е издание. Издательство “Символ-плюс”, 2010.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб:

СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.

Том 1. Основные алгоритмы.

Том 2. Получисленные алгоритмы.

Том 3. Сортировка и поиск.

Том 4, А. Комбинаторные алгоритмы.

Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Самое простое руководство по иконографии:  
<https://habrahabr.ru/company/netologyru/blog/331050/>

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Некоторые среды программирования: Blockly, Alice.

Веб-сервис для хостинга: GitHub.

Интерактивные тренажеры, упражнения:

Игра «Кто хочет стать миллионером» на практическое применение знаний английского языка: <https://learningapps.org/2384486>

Упражнение «Заполнить пропуски» на применение have to/ has to: <https://learningapps.org/3141784>

Упражнение «Найди пару» на соотнесени текста на английском языке и картинок: <https://learningapps.org/3222262>

Упражнение на определение базовых алгоритмических конструкций: <https://learningapps.org/1567206>.

Тренажер “Знаете ли вы Pascal”: <https://learningapps.org/2887689>.

Основы программирования на Pascal: <https://learningapps.org/3035365>.

Упражнения по программированию на C++: [http://ci-plus-plus-snachala.ru/?page\\_id=6](http://ci-plus-plus-snachala.ru/?page_id=6)

***Сроки выполнения (по уровням):***

1. 2 месяца.
2. 3 месяца.

3. Сроки остаются те же.

**8. «Let's play» - игра-головоломка, в которой требуется составить мозаику из множества фрагментов рисунка.**

Проект хоть и имеет игровой характер, все же несет массу положительных результатов обучения, начиная от мотивации (какому ученику не захочется написать свою «игрушку»), и заканчивая углубленными теоретическими и практическими знаниями. При создании собственной игры возникает множество алгоритмических задач, в данном случае: от вопросов хранения информации, до решения проблемы передвижения фрагментов пазла по игровому полю. Крайне важен дизайн и поддержка интереса пользователя.

***Для реализации необходимо:***

- наличие навыков программирования на каком-либо языке программирования под Android;
- навыки работы с MySQL для работы с базой данных (при выборе усложненного варианта технического задания);
- навыки установки и настройки связки Nginx + Apache или др. для реализации серверной части (при выборе усложненного варианта технического задания).

***Уровни сложности:***

1. Создать мобильное приложение-игру сборки пазлов. Функции:
  - Изображение для пазла может быть использовано как из галереи, так и взято непосредственно с камеры мобильного устройства.
  - Изображение “режется” на пазлы исходя из 3-х видов сложности.
  - Интерфейс представляет собой основной стол для сборки пазла и отдельную область с прокруткой с кусочками этого пазла.
  - Кусочки пазла могут крутиться на 90 градусов.
  - После успешной сборки должно выводиться сообщение.
2. Добавить функции:
  - Сохранения игры на любом этапе, и, как следствие, регистрацию в начале игры.



-Таймер и таблицу рекордов, непосредственно для данного изображения.

Так как задание будет выполняться в группе, нужно распределить обязанности между учениками. На выполнение задания 1-го уровня будет достаточно 4 человека. Два человека будут заниматься дизайном приложения и отрисовкой графики, два человека будут работать над алгоритмом выполнения задания. Для выполнения задания второго уровня нужно добавить еще двух человек, для организации хранения данных.

Сопровождать проект данного уровня должен эксперт со стороны, с которым у учеников должен быть налажен постоянный контакт. Так как далеко не каждый учитель или студент (тьютор) может помочь в выполнении данного задания.

#### ***Необходимый материал для начала и продолжения работы:***

Так как выбор учениками языка исполнения, сред разработки и т.д. мы не ограничиваем, нужно подстраховаться и предоставить материалы для основных (популярных, наиболее часто используемых) языков программирования и не только. Для данного проекта это могут быть:

#### Книги:

Арнольд К., Гослинг Д. Язык программирования JAVA. — СПб.: Питер, 1997-2002.

Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016.

Thornsby J. Android UI Design. Packt Publishing, 2016.

Stroud A. Android™ Database Best Practices.

Brothaler K. OpenGL ES 2 for Android. Copyright © 2013. The Pragmatic Programmers, LLC.

Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство; [пер. с англ. М. А. Райтмана]. — 3-е издание. — Москва : Эксмо, 2015.

Флэнаган Д. JavaScript. Подробное руководство. — Пер. с англ. — СПб: СимволПлюс, 2008.

Кнут Д.Э. Искусство программирования. Москва: Вильямс, 2002-2013.

Том 1. Основные алгоритмы.

Том 2. Получисленные алгоритмы.

Том 3. Сортировка и поиск.

Том 4, А. Комбинаторные алгоритмы.

Сайты и среды программирования:

Онлайн курсы по языкам программирования и не только: coursera.com, stepik.ru, intuit.ru.

Самое простое руководство по иконографии:  
<https://habrahabr.ru/company/netologyru/blog/331050/>

Онлайн школа Android-разработчиков: <https://learndroid.e-legion.ru/>

Веб-сервис для хостинга: GitHub.

3DG.ME – блог о 3D-графике и разработке игр:  
<http://3dg.me/ru/gamedev/basics/sozdanie-igr-dlya-nachinayushchih>

***Сроки выполнения (по уровням):***

3. 2,5 месяца.

4. 3 месяца.

### **2.3 Методические рекомендации для сопровождения проектов по программированию в условиях образовательной технологической платформы «Мега-класс»**

Проекты по программированию в условиях образовательного кластера имеют ряд специфических особенностей, как в плане подготовки, так и в плане проведения. Поэтому целесообразно предоставить методические рекомендации для каждого из участников-организаторов «Мега-класса» по использованию проектной деятельности в обучении программированию.

1. Для модератора.

На первом, вводном уроке следует ознакомить учащихся с планом разбора проекта с точки зрения программиста, чтобы они сразу вошли в роль

и почувствовали, какого это.

Этапы разработки программного продукта:

- 1) постановка задачи;
- 2) построение модели;
- 3) проектирование программы;
- 4) разработка алгоритма (общего и частных);
- 5) написание кода программы;
- 6) отладка программы;
- 7) тестирование программы;
- 8) документирование.

В инженерно-технологическом кластере (его состав описан выше как база эксперимента), мы на примере разработки мобильного приложения, которое строит улитку Паскаля прошлись по первым четырем из вышперечисленных этапов. А именно:

- Посмотрели готовое приложение на построение улитки Паскаля AmazingCurves для того чтобы понять, что вообще требуется в задании;
- построили графическую модель в среде GeoGebra;
- на этапе проектирования разобрали такие вопросы, как:
  - Какие математические модели лежат в основе приложения?
  - Какие элементы интерфейса вы можете выделить в приложении AmazingCurves?
  - Как бы вы описали назначение элементов интерфейса?
  - Что бы вы добавили в функционал приложения при следующей итерации?
- На этапе разработки обсудили:
  - Чего не хватает в алгоритмах геометрического построения для написания программы?
  - Как вы думаете, по какому алгоритму происходит рисование в приложении? (с точки зрения программирования).

Далее дали задание для самостоятельного решения: нужно было познакомиться с документом, описывающим некий проект (ссылка у учителя), понять задачу и ее математическую модель, представить каким должен быть, на их взгляд, интерфейс такого проекта и описать (словесно или блок-схемой) базовый вычислительный алгоритм проекта. Результат: готовый документ с техническим заданием. Ну, и конечно же, представили возможные направления и темы проектов, и обязали их определиться с выбором в течении двух недель.

Так как группы учеников из разных школ и городов могут работать над совершенно несхожими проектами, у них могут возникать как общие, так и абсолютно разные затруднения. Модератору, как основному координатору всей работы, следует заранее

- собирать вопросы от учителей и тьюторов;
- классифицировать и ранжировать вопросы;
- выбирать наиболее часто встречающиеся;
- проводить организовывать и координировать промежуточные разъяснительные уроки.

Как существует огромное количество различных алгоритмических задач, так и существует множество их решений. Поэтому, помимо стандартных этапов подготовки к данному виду урока, описанных выше, предварительно нужно организовывать встречи для обсуждения и выбора более рационального и понятного для учащихся способа решения задачи. Если есть ученики, которые уже справились с задачей, то следует взять за основу их алгоритм.

Например, на одном из промежуточных мега-уроков в том же кластере мы рассматривали задачу коммивояжера, и ее решение с помощью алгоритма Дейкстры. Выбор направления был обусловлен тем, что большинство проектов, выбранных учениками, подразумевали применение теории графов. Урок был организован таким образом: учащиеся сами с помощью презентации изучали (или повторяли) некоторые понятия теории графов, и сущность

алгоритма Дейкстры. Затем выполняли интерактивное упражнение на составление матрицы весов графа (так как она нужна для программного представления данных) в LearningApps (см. рис.12). Далее ученики были разбиты на группы, в которых решали различные задачи с помощью данного алгоритма.

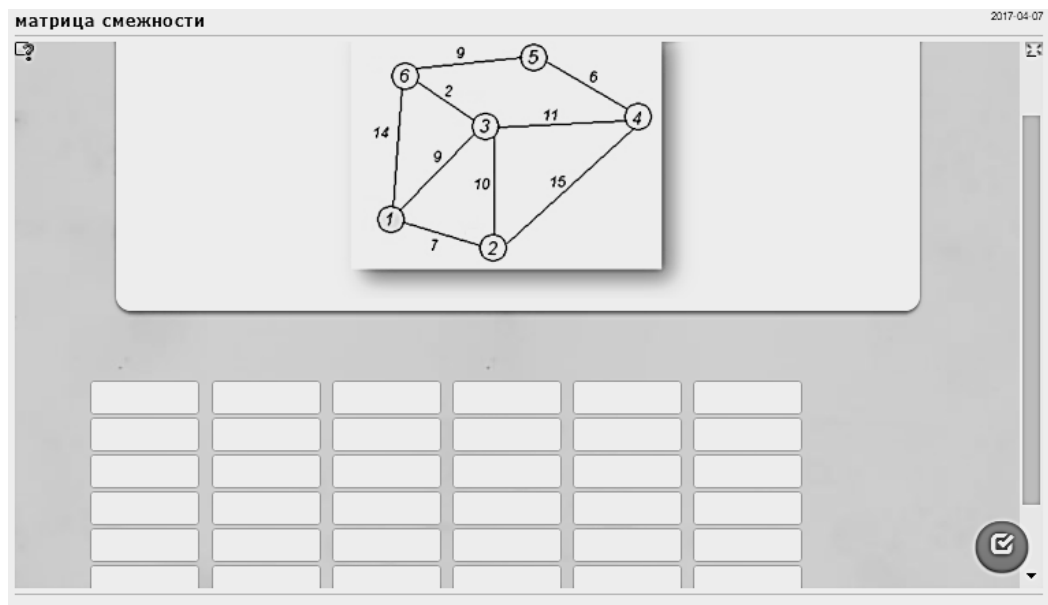


Рис.12. Задание на составление матрицы смежности графа

Во время урока нужно учитывать, что у некоторых групп учащихся нет потребности знать решение конкретно этой задачи, нужно как-то организовать их деятельность. Можно начать проверять их работу посредством кейса, которые они к тому моменту уже должны были начать заполнять. Либо объединить их по схожим проблемам (задачам), если они имеются, и организовать обмен опытом.

Различные способы организации защиты проектов были описаны в предыдущих параграфах. Сейчас лишь рассмотрим, как нам удалось провести заключительный урок.

Основная идея: к среде (перед уроком) ребята (команды) делают представление проекта в сети в произвольной форме (web-страничка, запись в блоге, онлайн презентация, видео, мультимедиа и т.п.), отправляют учителю ссылку.

При этом были сформулированы *требования к содержанию презентационных материалов* (см. таблицу 2).

## Требования к содержанию презентационных материалов

**Материалы в выбранной вами форме представления должны содержать:**

1. Описание сути (задумки) проекта, включая название и цели создания.
2. Назначение проекта (для кого проект).
3. Программный код для экспертной оценки.
4. Запускаемый файл с указанием требований к запуску.
5. Инструкция для запуска приложения, если требуется.
6. Демонстрация работы проекта и его возможностей (например, как видеоролик или серия скриншотов).

Учитель ссылки размещает в виртуальном фойе выставки (авторы, школа, название и ссылка). Среда-четверг все знакомятся с проектами друг друга и готовят оценочное мнение и вопросы. Для этого был выдан некоторый предварительный “план” (см. таблицу 3)

## План ознакомления с проектами

**При знакомстве с проектами:**

1. Обратите внимание на привлекательность описания (понятность, яркость).
2. Попробуйте запустить на своем компьютере исполняемый файл, оцените интерфейс пользователя.
3. Оцените оригинальность и/или полезность идеи проекта.
4. Из всех проектов выберите 2-3 понравившихся больше всего подумайте, в какой номинации вы бы дали этим проектам приз, если бы были в жюри конкурса.
5. ВАЖНО, Напишите список вопросов к авторам просмотренных проектов (не обязательно ко всем), например, к авторам понравившихся проектов. Их вы сможете задать на уроке.

Непосредственно на уроке учащиеся сначала заполняли анкету определяя лучшее приложение или проект (см. рис. 13).

**Анкета для определения лучших проектов по программированию**

\* Обязательно

1. Выберите самый лучший, по вашему мнению, проект. Учитывайте все показатели: задумка, описание, программный код и т.д (внесите название данного проекта). \*

Мой ответ \_\_\_\_\_

2. Выберите проект с самой интересной идеей (внесите название данного проекта). \*

Мой ответ \_\_\_\_\_

3. Выберите самый полезный проект с точки зрения его назначения (внесите название данного проекта). \*

Мой ответ \_\_\_\_\_

4. Выберите проект с самым сложным программным кодом (внесите название данного проекта). \*

Мой ответ \_\_\_\_\_

5. Выберите проект с самой качественной графической реализацией (внесите название данного проекта). \*

Мой ответ \_\_\_\_\_

**ОТПРАВИТЬ**

Никогда не используйте формы Google для передачи паролей.

Рис.13. Анкета для определения лучших проектов

После анкеты ребята задавали вопросы авторам проектов, таким способом: вопросы от школы 3 -> желающие выходят, говорят, что есть вопрос к проекту такому-то тех-то авторов и задают вопрос и т.д. Также было предложено выступить экспертам со своими впечатлениями.

## 2. Для учителя.

Во время назначения группы на ТЗ учителю нужно учитывать возможности учащихся, и рекомендовать им конкретный уровень сложности, так как они часто переоценивают свои силы. Для этого необходимо взаимодействовать с учителями из других школ, участники которой входят в одну группу, так как о их возможностях вы знать не можете. Распределением ролей учителя по-прежнему должны заниматься вместе, с учетом пожеланий учащихся. Также на учителей ложится сбор и анализ возникших у учеников

затруднений, своевременная корректировка ТЗ (пример технического задания, который мы предлагали учащимся был предоставлен в предыдущем параграфе).

Главная задача учителя в качестве сопровождающего проектной деятельности учащихся - это контролировать и направлять. Вовремя заметить, если есть затруднения или «торможение» в сроках, и помочь.

Например, у первой группы из Школы-лицея “Туран”, которые выбрали в качестве проекта создание игрового приложения на составление пазлов, возникла проблема сохранения результатов игры. Проект затянулся на долгое время только из-за этой проблемы. Тьютор не смог помочь учащимся, и было решено обратиться к эксперту.

Также учителю предстоит организовать сбор итогового проекта, если он был распределен по абсолютно разным подзадачам и до некоторого момента учащиеся работали индивидуально. Главная трудность здесь — это состав групп. Очень сложно организовать работу если в составе группы есть участники из разных школ. Так как уроки в вышеупомянутом кластере в таком виде проводились впервые, в экспериментальном формате, у нас не было таких групп.

### 3. Для тьютора.

В задачи тьютора главным образом входит договориться с учащимися о времени и способе взаимодействия. Если это социальная сеть или мессенджер-нужно создать группу или диалог, и удостовериться в том, чтобы все участники группы, которую он курирует, там состояли.

В том случае, если он не может помочь при возникшем затруднении, в составе кластера всегда должен быть в запасе человек, который может (эксперт), нужно лишь вовремя направить учащихся к нему, и организовать между ними взаимодействие.

Во время урока у тьютора особая роль: он выступает в качестве консультанта в тех вопросах, в которых может быть компетентен. Например, на вводном уроке на этапе построения графической модели в среде GeoGebra



мы с учителем математики из Гимназии №9 выступали в роли консультантов, так как учитель информатики не знал этот материал и не мог помочь ученикам.

#### 4. Для эксперта.

Эксперт должен заранее быть оповещен о регламенте урока, тематике, и системе оценивания деятельности учащихся. Также заранее согласовывается возможность индивидуальных консультаций в случае возникновения проблем у учащихся, с которыми он им может помочь. Для этого с помощью тьютора он налаживает с ними связь.

Во время урока он может выступать как в роли лектора, на вводных и промежуточных уроках, так и в роли эксперта-оценщика на заключительных мега-уроках. В случае участия в конкретных уроках ему также следует участвовать в подготовке урока, для согласования своей деятельности с учителями и модератором: от содержания лекционного фрагмента и диапазона тематик для дополнительных вопросов до критериев оценивания всех видов деятельности.

***Рассмотрим состав и работу инженерно-технологического кластера более подробно.***

Изначально было заявлено 5 школ, в итоге участвовало 4 школы, общее количество участников - 51 (см. таблицу 4).

Таблица 4

Учащиеся, входящие в состав инженерно-технологического кластера

№	Образовательное учреждение	Количество учащихся
1	Гимназия № 9	10
11	Школа №3	14
25	Школа-лицей Туран	12
37	Лицей №1 Ачинск	15

Конкретно по программированию было проведено 4 урока: вводный, два промежуточных и заключительный урок представления проектов.

Были проблемы в организационном плане, до конца дошла только школа Школа-лицей Туран, у них организовались три группы, проекты которых были

представлены на заключительном уроке. Первая группа выбрала в качестве задания разработку игрового приложения на составление пазлов, вторая – также игровое приложение на тренировку памяти «Найди пару», третья группа разработала приложение для интерактивного тематического тестирования. Все группы достигли поставленных целей, и защитили свои проекты.

Гимназия №9 также участвовала во всех уроках, но проекты на заключительном уроке не представила.

Если не считать заключительного урока, то в основном, работа на промежуточных уроках шла плодотворно, учащимся нравились задания, о чем мы могли судить по результатам тестирования в конце каждого урока. Велась рейтинговая таблица, в которой отмечались результаты каждого участника, и каждый мог видеть в чем он преуспел.

## Выводы по главе 2

1. В условиях открытой образовательной среды организационные формы обучения, используемые на Мега-уроке ориентированы, прежде всего, на *активное сетевое взаимодействие учащихся* различных школ, организованных в межшкольные группы сменного состава, в соответствии с целями и особенностью конкретного урока. Важным условием достижения высокого уровня мотивации учащихся и педагогов является *сетевое деловое сотрудничество* всех участников Мега-урока – учащихся, учителей школ, модератора урока, тьюторов и экспертов, в зависимости от роли, отведенной каждому на конкретном уроке [7].

2. Одно из требований к учебной работе учащихся - ее посильность на соответствующем этапе обучения, наличие у учащихся необходимого запаса знаний и умений. Вместе с тем, посильность задания предполагает не легкость его, а необходимость приложения максимума усилий для выполнения задания, ибо только это дает ощутимые результаты в развитии познавательных способностей. Важно также, чтобы содержание упражнений и самостоятельной работы представляло для учащихся интерес, что в значительной степени повышает мотив познавательной деятельности. Эти требования и определили содержание проектных заданий в системе разноуровневых задач для мега-урока.

## Заключение

Подводя итог всему сказанному, хотелось бы отметить, что использование метода проектов в условиях образовательного кластера позволяет:

- сделать процесс обучения более интересным, ярким и увлекательным, повысить интерес за счет креативного модуля, содержащего живые задачи и новизны такой формы работы для учащихся;
- индивидуализировать процесс обучения за счет возможности создания и использования разно уровневых заданий, усвоения учащимися учебного материала в индивидуальном темпе, с использованием удобного способа восприятия информации [11];
- эффективно решать проблему профорientированности обучения, расширить возможности визуализации учебного материала, делая его более понятным и доступным для учащихся;
- содействовать формированию базового интегрированного предметного тезауруса за счет образовательного модуля, содержащего интеллектуальные обучающие комплексы и ЦОР для самообразовательной деятельности обучающихся;
- способствовать развитию исследовательского и проектного стиля мышления за счет исследовательского модуля, содержащего необходимые источники и материалы, включая виртуальные среды и лаборатории, для проведения исследований школьниками, студентами и учеными в рамках вертикальных кластерных мега-групп [7].

В данном исследовании мы раскрыли сущность образовательного кластера на основе технологии платформы «Мега-класс», выделила особенности использования проектной деятельности в среде образовательного кластера на уроках информатики, разработала комплекс проектных заданий и дала методические рекомендации для сопровождения проектов по программированию. Таким образом, все задачи исследования выполнены,

цель достигнута. В дальнейшем планируется провести как можно больше уроков и вывести статистику результатов обучения.

## Список литературы

1. Белова О.Е. Методика обучения студентов педагогических вузов — будущих учителей математики интегральному исчислению с использованием информационных технологий: дис. ... к-та пед. наук. Красноярск, 2006.
2. Бойцева И. И. Метод проектов: новые подходы к повышению эффективности [Электронный ресурс]. URL: <https://www.pedt.ru/categories/4/articles/287> (дата обращения: 10.06.2017).
3. Гроздев С.И. Визуализация методов сортировки массивов. Республика Болгария, г.София. [Электронный ресурс]. URL: [http://yandex.ru/clck/jsredir?from=yandex.ru%3Byandsearch%3Bweb%3B%3B%2Fweb%2Fitem%2Furlnav%2Cpos%2Cp0%2Csource%2Cweb\\_tdi%2Curl%2Curl](http://yandex.ru/clck/jsredir?from=yandex.ru%3Byandsearch%3Bweb%3B%3B%2Fweb%2Fitem%2Furlnav%2Cpos%2Cp0%2Csource%2Cweb_tdi%2Curl%2Curl) (дата обращения: 05.06.2017).
4. Дацун Н.Н., Уразаева Л.Ю. Использование визуализации при обучении программированию // Преподаватель XXI век. 2015. Выпуск №4.
5. Ивкина Л. М. Формирование методической готовности будущих учителей информатики в условиях образовательной платформы «Мега-класс»: дисс. ... к-та пед. наук. Красноярск, СФУ, 2017.
6. Ивкина, Л.М. Методическое сопровождение мегауроков в условиях глобализации учебного процесса / Л.М. Ивкина, Л.Б. Хегай // Информатика и образование. – 2015. – № 10 (269).
7. Ивкина Л.М., Кулакова И.А., Пак Н.И., Романов Д.В., Симонова А.Л., Сокольская М.А., Хегай Л.Б., Яковлева Т.А. Мегакласс как инновационная модель обучения информатике с использованием ДОТ и СПО: коллективная монография //Краснояр. гос. пед. ун-т им. В.П. Астафьева. — Красноярск, 2014.
8. Ивкина Л.М., Пак Н.И. Технология «Мега-класс» как средство коллективной учебной деятельности в образовательных кластерах // Открытое образование. 2015. № 5. С. 32-38.

9. Коротенко М.Н. Визуализация рекурсивных алгоритмов на примере треугольника Серпинского с помощью библиотеки OpenGL / Молодежь наука XXI века: материалы XVI Междунар. молодеж. конф. «Инновационные средства и методы обучения в условиях ИКТ», Красноярск, 2015.
10. Крылатова Н. Н. Метод проектов при изучении иностранного языка. Томск, 2012. [Электронный ресурс]. URL: <https://www.prodlenka.org/pedagogicheskaja-masterskaja/3242-metod-proektov-pri-izuchenii-inostrannogo-jaz.html> (дата обращения: 10.05.2017).
11. Ларина О.Ю. «Применение информационных технологий на уроках математики». Описание опыта работы учителя математики МКОУ СОШ№39. Тула, 2014. [Электронный ресурс]. URL: <http://nsportal.ru/shkola/materialy-k-attestatsii/library/2014/01/19/primenenie-informatsionnykh-tekhnologiy-na-urokakh> (дата обращения 17.03.2017).
12. Лебедева Т.Н., Юнусова С.С. Alice как объектно-ориентированная среда для разработки компьютерных игр // Вестник Брянского государственного университета. 2015. Выпуск №1.
13. Ляпина О. В. Проектная деятельность как средство самоактуализации старших подростков: дисс. ... к-та пед. наук. Рязань, Рязан. гос. пед. ун-т им. С.А. Есенина, 2009.
14. Мазничевская Л.И. Деятельность педагога-куратора в системе дистанционного обучения // Информатика и образование. 2008. №8.
15. Мазуревич В.В. 10 заповедей технического задания (с толкованием) [Электронный ресурс]. URL: <http://www.cossa.ru/234/125799/> (дата обращения 10.02.2017).
16. Максакова Д.А., Горностаева Т.Н. Применение метода проектов в обучении математики / Сб. трудов Региональной научно-практической конференции «Использование информационно-коммуникационных технологий в современной системе образования», Владивосток, 2017.
17. Мамаева А.А., Баран М.И. Задания в проекте «Мега - класс» /

Материалы Международной научно-практической конференции «Новая наука: современное состояние и пути развития», Оренбург, 2016.

18. Нигматулина Э. А. Практико-ориентированные «живые» задачи в школьном курсе информатики / Студенческая наука XXI века: материалы X Междунар. студенч. науч.-практ. конф., Чебоксары, 2016.

19. Пак Н.И. Инновационная технология «Мега-класс» как синергетическое средство обучения в образовательных кластерах / Сб. трудов Международной научно-практической конференции «Информатизация образования-2015», Казань, 2015.

20. Пак, Н.И. От классно-урочной системы к кластерному образованию: образовательная технологическая платформа «Мега-класс» / Н.И. Пак // 119 Материалы международной научно-практической конференции «Информатизация образования-2016». – Сочи : Изд-во СГУ, 2016.

21. Пак Н.И., Сокольская М.А. Региональная модель образовательного кластера на технологической платформе "Мега-класс" // Педагогическая информатика. - 2017. - №4.

22. Поляничева Н. О. Роль проектной деятельности в достижении современных образовательных результатов // Молодой ученый. — 2015. — №4.

23. Пуденко, Т.И. Образовательные кластеры как модель управления развитием образования на муниципальном уровне, повышающая доступность качественных образовательных услуг / Т.И. Пуденко // Управление образованием: теория и практика. – 2014. – № 3 (15). – С. 33–45.

24. Пушкарева Т.П., Калитина В.В. Визуализированная методика обучения программированию // Современные проблемы науки и образования. 2014. Выпуск №5.

25. Сердюков С. О. Prezi - создание интерактивных мультимедийных презентаций с нелинейной структурой. [Электронный ресурс]. URL: <https://sites.google.com/a/schoolkovrov.ru/ucebnyj-centr-ga-domena-schoolkovrov-ru/cifrovye-obrazovatelnye-resursy/prezi-sozdanieinteraktivnyhmultimedijnyhprezentacijnsnelinejnojstrukturoj> (дата



обращения 2.02.2017).

26. Слинкин Д. А. Использование метода проектов при обучении программированию в курсе информатики: Дис. ... канд. пед. наук. Екатеринбург, 2001.

27. Стригалева Н.А. Применение метода проектов для развития творческих способностей одаренных учащихся / Материалы творческой мастерской №1 «Создание единого информационного образовательного пространства для развития талантливых детей», Дзержинск, 2016.

28. Сухорукова Н. А. Первые шаги в Edmodo / Материалы семинара «Он-лайн обучение школьников: цели, способы, возможности». Лиман, 2012. [Электронный ресурс]. URL: <https://www.slideshare.net/spogrebnaк/edmodo-13948534> (дата обращения 21.02.2017).

29. Теплоухова Л. А. Деятельностный подход в обучении. Понятие проектирования как деятельности // Общепедагогические технологии. – 2008. - № 2.

30. Филиппова Ю. М. Создание интерактивных заданий в программе Hot Potatoes (JMatch) [Электронный ресурс]. URL: <http://festival.1september.ru/articles/645131> (дата обращения 20.04.2017).

31. GitHub / Материал из Национальной библиотеки им. Н. Э. Баумана [Электронный ресурс]. URL: <http://ru.bmstu.wiki/GitHub> (дата обращения 15.04.2017).