

Завершено
Директор МБОУ / Р. М. Тасанова /



Протокол № 2

Заседания АУИ Сергокалинского района от 21.11.2019 года.

Присутствовали: методист УО Магомедова У.К., учителя информатики Сергокалинского района.

Повестка дня:

Тема: «Оценка качества основного общего образования в условиях реализации ФГОС ООО» село Краснопартизанск-слет ассоциации»

1. Открытый урок- Чанкаева М.О. – 8 класс- тема: «Основы алгоритмизации»
2. Открытое мероприятие (КВН) в 10-11 классах
3. Самоанализ и анализ урока.
4. Презентация опыта работы.
5. Доклад-«Современные проблемы в преподавании информатики» (загирбекова П.К.- Кичигамринская СОШ)

Ход заседания

1.Открытый урок- Чанкаева М.О. 8 класс – «Основы алгоритмизации».

Решение. Принять к сведению.

2.Открытое мероприятие –КВН в 10-11 классах

Решение. Принять к сведению.

3.Самоанализ и анализ урока.

По третьему вопросу выступила Чанкаева М.О. и члены АУИ. Чанкаева М.О. – учитель доволен ответами учащихся, учащиеся приняли активное участие и хорошо решали задачи. Вместе с учащимися составили алгоритмы, запустили их в программе Кумир.

Магомедова Г.Ш.- хотя учащихся в классе мало урок прошел интересно, ученики отвечали хорошо.

Загирбекова П.К.- на уроке использованы ЭОР, учащиеся активно работали.

Решение. Принять к сведению.

4.Презентация опыта работы.

По четвертому вопросу выступил Алиев З.М.-учитель информатики МКОУ «Мюрегинская СОШ». Он поделился опытом работы подготовки учащихся к ЕГЭ, методическими руководствами по проведению уроков, показал презентацию о своей работе.

Решение. Учителям ассоциации делиться опытом работы

5.Доклад- «Интернет защита для точки доступа Wi-Fi»

» (Джалилова П.Б.- Аялизимахинская СОШ)

По пятому вопросу выступила Джалилова П.Б. с докладом (доклад прилагается)

Решение: принять к сведению доклад Джалиловой П.Б.

Руководитель ассоциации

/Адзиева Х.З./

Секретарь

/Магомедова Г.Ш./

Заверен
Директор



/Р.М. Тасанова/

ДОКЛАД

на тему:

«Задания по алгоритмике и программированию в ГИА

Подготовила: Загирбекова П.К.

Информатика в школьном курсе дает несколько особых знаний и умений, без которых невозможно ни быть успешным на рынке труда сегодня, ни получить образование, которое позволит остаться успешным завтра. Одно из самых важных человеческих умений – это умение составить, а затем и претворить в жизнь план некой будущей деятельности. Заглянув в Энциклопедический словарь, можно обнаружить: такой план называется программой. Решение задач по физике и информатике в реализации задач на программирование можно связать, да и нужно связать. Привычка тратить время и силы на обдумывание, запись и отработку планов будущей деятельности себя самого, других людей или больших коллективов называется алгоритмическим стилем мышления. Овладеть алгоритмическим стилем мышления непросто. Для этого нужно научиться заранее предсказывать ситуации, которые могут случиться в будущем, и предусматривать в планах правильное поведение в этих ситуациях. С другой стороны, как и другие человеческие навыки, алгоритмический стиль мышления можно развивать и тренировать путем целенаправленно подобранной системы упражнений.

Поэтому большое внимание в преподавании я обращаю на раздел «Алгоритмы. Программирование». За время моей работы учителем информатики и физики актуальность темы «Понятие алгоритма. Программирование» претерпела значительные изменения. Большая часть времени отводится на преподавание тем цикла «Информационные и коммуникационные технологии». Наряду с этим несколько не изменились требования к уровню усвоения знаний и умений этого раздела программы по информатике, так как он остается основой фундаментальных знаний по предмету. Вместе с тем ФИПИ требует знаний учащихся в старом объеме курса информатики, особенно в части знания программирования. Учащиеся для успешной сдачи экзамена должны не только знать основные алгоритмические конструкции и операторы изучаемого языка программирования, но и иметь опыт самостоятельной записи алгоритмов и программ, решения практических задач методом разработки и отладки компьютерной программы. Следует уделять больше внимания формализации записи и исполнения алгоритмов, так как результаты экзамена показывают, что у части учащихся так и не формируется умение формального исполнения алгоритмов. Поэтому для себя я решил, что необходимо изыскать все возможности, чтобы преподавание темы «Понятие алгоритма. Программирование» оставалось на достаточном высоком теоретическом и практическом уровне.

Мной была включена в календарно-тематическое планирование 8 класса тема «Исполнители алгоритмов», а в 9 классе «Программирование». Хорошо известно, что развитие алгоритмического (процедурного) мышления учащихся происходит тем эффективнее, чем раньше оно начинается. Дидактические средства для этого хорошо отработаны - это разнообразные учебные исполнители алгоритмов: черепахи, роботы, чертежники. За основу я взяла учебник Угриновича «Информатика 8-9 класс, а так же Босова Л.Л. Дети разрабатывают алгоритмы и пишут программы для исполнителя стрелки, знакомятся с понятием алгоритма и основными алгоритмическими конструкциями, а также с процедурами и подпрограммами. В ходе уроков используются игровые моменты, при написании программы все действия исполнителя отображаются на экране, причем при выполнении программы используется анимация. Использование таких исполнителей с методической точки зрения очень эффективно. Основные достоинства – понятность решаемых задач, наглядность работы исполнителя, поддержка структурной методики алгоритмизации. Если кто-то думает, что это лишь игрушки для младших школьников, то он ошибается. Из личного преподавательского опыта знаю, с каким интересом ученики 8-х и даже 10-х классов решают алгоритмические задачи для исполнителей. Уровни сложности задач можно варьировать от тривиальных до головоломных, адаптируя к способностям учеников. Задача развития структурного алгоритмического мышления учащихся решается в полной мере на учебных исполнителях, работающих «в обстановке».

Обучение школьников алгоритмизации и программированию с методической точки зрения является одной из самых трудных задач. Составление программ – весьма сложный процесс, включающий в себя значительное число качественно разнообразных этапов. Наиболее сложные из них – постановка задачи и ее алгоритмизация. Именно этим этапам уделяю основное внимание при разработке методик по преподаванию разделов алгоритмизации и программирования. Как правило, в учебных пособиях при объяснении данных разделов приводятся конкретные задачи различных типов и соответствующие им алгоритмы с некоторыми частными пояснениями. В основном рассматриваются примеры линейного, разветвляющегося и циклического типов, а также алгоритмов с «вложенными циклами». Если дается решение более сложных задач, то приводятся готовые алгоритмы решения без их описания процесса построения алгоритмов. Считается, что, изучив приведенные примеры,

учащийся сможет составить алгоритм любой задачи. Однако это далеко не так.

При обучении алгоритмизации необходимо использовать унифицированную и формализованную процедуру перехода от словесно-формульного описания метода решения задачи к схеме алгоритма этой задачи, причем такой схеме, которая может быть формально перекодирована в программу на алгоритмическом языке. Эта процедура позволяет строго логически выводить формулы и условия, составляющие «начинку» алгоритма (программы). Особенность такого подхода еще и в том, что он требует от учащегося подробного описания процесса вывода алгоритма, поэтому процесс обучения алгоритмизации становится хорошо наблюдаемым для учителя, а значит, и хорошо управляемым. В качестве основного средства описания алгоритмов я выбираю блок-схемы алгоритмов – наиболее наглядный и понятный, а кроме того, и наиболее естественный для человека способ, т.к. человек мыслит образами и ему легче воспринимать образы, нежели текст.

Данная методика вполне доступна любой категории учащихся на начальной стадии обучения информатике и программированию. Она базируется на известных подходах к построению программ – структурном программировании и проектировании «сверху вниз» и является их развитием. Методика достаточно универсальна.

И в дополнение, как обязательный элемент обучения алгоритмизации должна входить отладка алгоритма каждого вида. Она позволяет учащимся «прочувствовать» суть каждого вида алгоритмов, предлагает им правила самоконтроля – правила нахождения ответов к задачам алгоритмизации, позволяет понять и закрепить принципы отладки программ. Как показывает практика, использование в обучении информатике методики формализованного перехода от схемы алгоритма к программе на алгоритмическом языке позволяет научить основную массу учащихся составлять алгоритмы и программы достаточно сложных задач в отличие от традиционного подхода, при котором можно говорить лишь о знакомстве с понятиями «алгоритм», «программа» и пр. в процессе изучения этого предмета.

Второй аспект заключается в связи линии алгоритмизации и программирования с линией компьютера, с более глубоким раскрытием понятия программного управления.

При наличии небольшого объема учебного времени, программирование в базовом курсе может изучаться лишь на уровне

введения. Основная задача ограничивается рамками все той же линии компьютера: раскрывается понятие программного управления работой компьютера. Изучение происходит на примерах простых программ на Паскале. Показывается, как организуется простейший диалог компьютера с человеком: компьютер спрашивает, ученик отвечает, компьютер реагирует на ответ в соответствии с его содержанием. Показывается, как организуются простейшие вычисления, например, вводится числовая последовательность, выводится ее среднее арифметическое значение; или вводятся два числа, выводится их наибольший общий делитель (алгоритм Евклида) и т.п. И все! Этого вполне достаточно с точки зрения поставленной цели.

Изучение программирования как прагматической цели заключается в освоении основ профессионального программирования. Сегодня программирование на любительском уровне с практической точки зрения не представляет интереса. Используя прикладные программы можно сделать гораздо больше, чем с помощью языков программирования на ученическом уровне. Такую цель можно ставить только перед профильным или элективными курсами информатики.

Конечно, очень тяжело уложиться в имеющиеся учебные часы. Любой практикующий учитель поймет, что и этот план очень напряженный и не прост в реализации (особенно в режиме 1 урок в неделю). А уж где там до массивов, списков, деревьев, графов, не говоря уж об освоении программирования динамических структур в базовом курсе.

Работа по развитию логического и алгоритмического мышления проходит не только на уроках, а также и на внеклассных занятиях.

Также, в нашей школе имеется возможность изучать программирование в 10-11 классах на базе Точка Роста. В этом случае можно последовательно продвигаться от одного практического занятия к другому.

Ведь для успешного участия в олимпиаде по программированию школьник должен не только владеть языком программирования, но и уметь придумывать и реализовывать алгоритмы решения задач, оценивать время их работы, тестировать и отлаживать свои программы.



Протокол № 4

Заседания АУИ Сергокалинского района от 12.03.2020 года.

Присутствовали: методист УО Магомедова У.К., учителя информатики Сергокалинского района.

Повестка дня:

Тема: «Проектная деятельность в основной школе в аспекте содержания ФГОС ООО» село Нижнемулебки

1. Сообщение – презентация из опыта работы по теме «Методика работы над проектами в школе»
2. Сообщение-презентация из опыта работы «Личностное развитие школьника в проектной деятельности»
3. Доклад- «Методика обучения языкам программирования Scratsh»- (Загирбекова П.К.)
4. Мастер-класс «Что значит «Уметь программировать- Scratsh» (Мургукская СОШ) - учитель информатики –Магомедова А.М.)
5. Самоанализ и анализ урока

Ход заседания

1. Сообщение – презентация из опыта работы по теме «Методика работы над проектами в школе» -выступила учитель информатики Карачева П. «Урахинская СОШ»
2. Сообщение-презентация из опыта работы «Личностное развитие школьника в проектной деятельности»

Решение. Принять к сведению.

3. Доклад-«Методика обучения языкам программирования Scratsh»- (Загирбекова П.К.)

Решение: принять к сведению доклад Загирбековой П.К.

- 4.Мастер-класс «Что значит «Уметь программировать Scratsh» (Мургукская СОШ) - учитель информатики -Магомедова А.М.)
- 5.Самоанализ и анализ урока

По 5-му вопросу выступил Алиев З.М. и Магомедов А.М., члены АУИ. Магомедов А. М. – учитель доволен ответами учащихся, учащиеся приняли активное участие и хорошо решали задачи. Гамидова Г.С.- урок прошел интересно, ученики отвечали хорошо, видно, что Магомедов А.М. смог вызвать интерес у учащихся к программированию.

Мирзамагомедов М.М.-учитель информатики Миглакасинская СОШ. Урок по программированию в среде Scratch очень интересен. Используя эту программу можно научить детей составлять свои игры.

Решение. Принять к сведению.

Работа ассоциации была плодотворна, было очень много интересного и нового.

Руководитель ассоциации:

/Адзиева Х.З./

Секретарь:

/Магомедова Г.Ш./

Заведующий
Директор



Тасанова Р.М.
МКОУ «Кичигамринская СОШ»

Методика по преподаванию раздела

«Программирование в среде Scratch»

Подготовил: учитель физики и информатики-Загирбекова П.К.

2020г.

Scratch разрабатывался как новая учебная среда для обучения школьников программированию и позиционировался авторами как альтернатива культе Photoshop.

К настоящему времени накоплен немалый опыт в методике преподавания Scratch в начальной школе. Scratch может использоваться в различных формах: в школах, музеях, центрах дополнительного обучения, дома. Он создавался для учеников 8 – 16 лет, но и дети более раннего возраста могут работать над Scratch проектами вместе с родителями или старшими братьями или сестрами.

«Программирование в среде Scratch» для младших школьников.

Для достижения цели работы определен ряд задач:

- 1) Определение особенностей обучения младших школьников программированию;
- 2) Анализ существующих методик обучения младших школьников программированию;
- 3) Разработка методики преподавания раздела «Программирование в среде Scratch»

Программирование в среде Scratch

Scratch – среда программирования, появившаяся относительно недавно, даёт возможность учащимся младшего и среднего школьного возраста создавать игры, фильмы, анимированные истории и многое другое. Программа Scratch в объектно-ориентированной среде «собирается» из разноцветных блоков команд так же, как собираются из разноцветных кирпичиков в конструкторах Лего различные объекты. Создание программ в Scratch происходит путём совмещения графических блоков в стеках. При этом блоки сделаны так, что их можно совмещать только в синтаксически верные конструкции, что исключает ошибки. Различные типы данных имеют разные формы блоков, подчеркивая совместимость / несовместимость объектов между собой. Имеется возможность внесения изменений в программу даже тогда, когда она запущена, что позволяет экспериментировать с новыми идеями по ходу решения задачи. В результате выполнения простых команд создаётся сложная модель, в которой взаимодействуют множество объектов, наделенных различными свойствами.

Мы можем начинать пользоваться языком с нуля, не обладая никакими предварительными знаниями о программировании. Важно отметить, что Scratch приходит в современный мир вместе с другими важными педагогическими инициативами. Это движение свободных программ и движение «Один ребенок – один компьютер» – «1:1».

Согласно идеологии этого движения ребенок должен осваивать не программы-приложения, а различные способы деятельности: создавать свои собственные истории,

придумывать игры, разрабатывать компьютерные модели. Для таких целей Scratch подходит как нельзя лучше. Разработчики языка думают о том, как дети в среде Scratch смогут самостоятельно осваивать современную культуру, играть с образами, звуками, мультипликацией. В этой среде ученики не используют готовые компьютерные игры, а конструируют свои собственные игры, истории и модели. В ходе этой созидательной деятельности у учеников формируется свобода обращения с различными элементами окружающей медиакультуры.

Цель движения «1:1» состоит в том, чтобы не просто дать детям технические средства, но помочь им научиться думать лучше и глубже, чем думают большинство взрослых. Для достижения этой цели создаются новые технические и программные среды – как новые формы электронной бумаги, на которой могут быть представлены новые пути представления новых мощных идей. Персональные компьютеры, доступные для каждого ученика, создают новую среду чтения и письма, в которой люди могут играть с новыми мощными идеями совсем не так, как они это делали с книгами. Образцы этой новой литературы публикуются в сети Интернет, для того, чтобы и взрослые и дети могли бы читать, играть с ними, классифицировать, обсуждать и критиковать. Новый стиль усвоения знаний дает ученикам возможность обдумывать идеи и представлять их в сетевой форме доступной для обсуждения и критики. Как отмечает Алан Кей «возможность обсуждения и критики является одним из самых древних оснований обучения». Опыт взаимодействия с глубокими идеями помогает вырастить людей, которых будет значительно труднее ввести в заблуждение, поскольку у них будет заложен опыт критического отношения и активной работы с любыми мифами и мимами.

Впервые отношение к компьютеру как к дружественному средству, которое ученик может контролировать и при помощи которого он может думать новыми, более эффективными способами, сформулировал Сеймур Пейперт. В своих работах он сместил направление педагогических инноваций с поиска лучших методов преподавания на поиск лучших объектов, при помощи которых можно конструктивно действовать и размышлять о своей деятельности. Пейперт и его коллеги предположили, что построение собственных интеллектуальных структур осуществляется учеником наиболее эффективно в том случае, если он вовлечен в создание реального конечного продукта: замка из песка, машины, книги или компьютерной программы. Лучше всего обучение происходит в том случае, если ученик вовлечен в деятельность, которая сопровождается размышлением и обсуждением о том, что он делает. Люди учатся значительно эффективнее, если они вовлечены в создание объектов, которые имеют для них личное значение. Компьютеры и компьютерные сети позволяют людям получать доступ к новым объектам, создавать и

играть с гораздо большим числом искусственных объектов, чем это было возможно раньше. При помощи этих новых учебных объектов ученики могут конструировать новые истории, новые произведения и объекты. Они могут проводить исследования этих новых объектов и описывать их поведение.

Одним из принципиальных достоинств данной среды является то, что она является свободно распространяемым программным продуктом, таким образом, любое образовательное учреждение может скачать программу из интернета и приступить к непосредственному изучению и работе в новой среде программирования. Scratch не требует установки.

Сама идеология Scratch позволяет использовать при обучении современные методики и технологии обучения, такие как проблемный подход и метод проектов. После изучения основных конструкций языка и возможностей среды ставится задача по созданию и разработке соответствующего проекта. Это могут быть различные истории, тематику которых учитель предлагает с учётом возрастных особенностей учащихся, например, «Моя семья», «Мои увлечения», «Талантливые люди» (традиционная тематика проектов для пропедевтического обучения); рекламные ролики; анимированные истории по стихам и сказкам, изучаемым в школе и просто любимым учениками и т.д.[11]

Данная технология обучения стимулирует учащихся к освоению возможностей языка программирования, изучению предмета «Информатика и ИКТ», подчеркивая их практическую личностную значимость. Несмотря на свою простоту, Scratch предоставляет пользователю разнообразные средства работы с мультимедийными ресурсами, что вызывает интерес у учащихся, способствует развитию положительной мотивации к предмету в целом.

Среда Scratch разработана и поддерживается группой MIT Media Lab из Массачусетского института технологии (<http://scratch.mit.edu>). Scratch является бесплатным продуктом, что немаловажно для российских школ. Его разработка ведётся на базе языка Squeak, одном из наследников Smalltalk. Алан Кей, стоящий у истоков Smalltalk, а значит и у самых корней объектно-ориентированных технологий вообще, очень заинтересованно относится к развитию мышления и креативности детей. Среда (и язык) программирования Scratch, по задумке его создателей, является как раз тем средством, которое способно формировать «способы мышления».

Scratch используется метафора кирпичиков Лего, из которых даже самые маленькие дети могут собрать простейшие конструкции. Дети могут собирать свои программы-процедуры из блоков так же, как они собирали конструкции из разноцветных кирпичиков. Из конструкций и управляющих структур могут быть собраны различные

агенты, выполняющие простые инструкции. Эти агенты могут взаимодействовать между собой и воссоздавать сообщество в среде Scratch. В результате выполнения простых команд может складываться сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Начальный уровень программирования столь прост и доступен, что Scratch рассматривается в качестве средства обучения не только старших, но и младших школьников.

Одной из главных концепций языка Scratch, является развитие собственных задумок с первой идеи до конечного программного продукта. Для этого в Scratch имеются все необходимые средства:

- стандартные для языков процедурного типа: следование, ветвление, циклы, переменные, типы данных (целые и вещественные числа, строки, логические, списки – динамические массивы), псевдослучайные числа;
- объектно-ориентированные: объекты (их поля и методы), передача сообщений и обработка событий;
- интерактивные: обработка взаимодействия объектов между собой, с пользователем, а также событий вне компьютера (при помощи подключаемого сенсорного блока);
- параллельное выполнение: запуск методов объектов в параллельных потоках с возможностью координации и синхронизации;
- создание простого интерфейса пользователя.

Вместе с тем в Scratch пока отсутствуют такие важные компоненты языка вообще и объектно-ориентированного языка в частности, как процедуры и функции, передача параметров и возвращение значений, определение классов объектов, наследование и полиморфизм, обработка исключений, текстовый ввод и файловый ввод и вывод. Однако не вызывает сомнений, что в ближайшем будущем язык пополнится этими и, вероятно, некоторыми другими компонентами. Scratch действительно имеет богатые возможности. При этом для начала его изучения не требуется ничего, кроме умения читать, поскольку программа составляется из готовых цветных блоков. Этому уровню соответствуют практически все первоклассники.

Одной из важнейших особенностей проекта Scratch является большое постоянно действующее сообщество пользователей. Собственные разработки можно выкладывать в интернет и затем просматривать их на любом компьютере, где установлена Java (или сам Scratch). В настоящее время актуальной является версия Scratch 1.3, поддерживающая кириллицу.

Уже в начальной школе дети легко могут освоить такие понятия как «параллельность» и «синхронизация». При этом важным является не «знания» терминологии, но понимание взаимной связи выполняющихся потоков.

Scratch берет все лучшее от вычислительной техники и дизайна интерфейсов для того, чтобы сделать процесс программирования более привлекательным и доступным для детей, подростков и тех, кто хочет научиться программированию. [15]

Основные особенности Scratch:

- Блочное программирование. Для создания программ в Scratch, вы просто совмещаете графические блоки вместе в стеках. Блоки сделаны так, чтобы их можно было собрать только в синтаксически верных конструкциях, что исключает ошибки. Различные типы данных имеют разные формы, подчеркивая несовместимость. Вы можете сделать изменения в стеках, даже когда программа запущена, что позволяет больше экспериментировать с новыми идеями снова и снова.

- Манипуляции данными. Со Scratch вы можете создать программы, которые управляют и смешивают графику, анимацию, музыку и звуки. Scratch расширяет возможности управления визуальными данными, которые популярны в современной культуре – например, добавляя программируемость, похожих на Photoshop фильтров.

- Совместная работа и обмен. Сайт проекта Scratch предлагает вдохновение и аудиторию: вы можете посмотреть проекты других людей, использовать и изменить их картинки и скрипты, и добавить ваш собственный проект. Самое большое достижение – это общая среда и культура, созданная вокруг Scratch.

Scratch предлагает низкий пол, высокий потолок и широкие стены. В работе со Scratch уделяется особое внимание простоте, иногда даже в ущерб функциональности, для большей понятности.

Когда учащиеся работают над проектом в Scratch, они имеют шанс выучить важные вычислительные концепции, такие как повторения, условия, переменные, типы данных, события и процессы. Scratch уже был использован для представления этих понятий учащимися различных возрастов, от элементарной школы до колледжа. Некоторые учащиеся перешли от традиционных, текстовых, языков программирования, после того, как им показали программирование в Scratch.

При создании сложных проектов ученик не просто освоит азы программирования, но и познакомится с полным циклом разработки программы, начиная с этапа описания идеи и заканчивая тестированием и отладкой.