

Оглавление

Оглавление	2
Введение.....	3
Глава 1. Теоретическое обоснование использования кинестетических тренажеров при обучении программированию.....	7
1.1. Особенности обучения теме «Алгоритмизация и программирование» в школьном курсе информатики.....	7
1.2. Роль телесного подхода при формировании алгоритмического мышления.....	9
1.3. Условия реализации телесного подхода в обучении программированию.	11
Глава 2. Практические аспекты использования кинестетических тренажеров при обучении программированию.	22
2.1. Проектирование кинестетических тренажеров по теме «Циклы».....	22
2.2. Методические рекомендации по использованию кинестетических тренажеров при обучении теме «Циклы».....	27
2.3. Результаты апробации	28
Заключение	31
Список литературы	33
Приложения	35
Приложение А	35
Приложение Б.....	38

Введение

Актуальность:

Когда говорят о развитии алгоритмического мышления у студентов и школьников, в основном подразумевается, что этот процесс происходит на уроках информатики и математики, поскольку именно там учащиеся погружаются в алгоритмическую деятельность. И особенно активизируется этот процесс при изучении темы «алгоритмизация и программирование».

Основной проблемой при изучении этих тем является то, что, как правило, сформулировав задачу, преподаватель изображает блок-схему, иллюстрирующую алгоритм ее решения или же даже сразу пишет программный код. Но этот подход ориентирован на аудиторию с достаточным уровнем сформированности алгоритмического мышления, способную воспринять решение задачи на том достаточно высоком уровне абстракции, каковым является блок-схема или еще более высоким – если сразу код программы, каким бы несложным и коротким он казался с первого взгляда.

Следовательно, необходимо создание новых средств формирования алгоритмического мышления, которые были бы нацелены на кинестетические каналы восприятия и активизацию моторной области памяти. И по нашему мнению это будет актуально не только для учащихся с преобладающим кинестетическим типом восприятия информации, поскольку, как известно деление по типам восприятия достаточно условно, редко встречаются чисто выраженные типы, чаще смешанные. Поэтому активизация всех каналов восприятия при изложении учебного материала позволит существенно повысить его уровень понимания.

Проблема:

Поскольку тема «Алгоритмизация и программирование» остается одной из самых сложных тем школьного курса информатики, следовательно,

существующие подходы к обучению этой теме не являются достаточно эффективными, и необходим поиск новых подходов к решению данной проблемы.

Цель:

Теоретически обосновать возможность и необходимость использования кинестетических тренажеров при обучении программированию, спроектировать примеры таких тренажеров и методические рекомендации по их использованию в учебном процессе.

Объект:

Обучение теме «Алгоритмизация и программирование» в школьном курсе информатики

Предмет:

Использование кинестетических тренажеров при обучении алгоритмизации и программированию в школьном курсе информатики.

Гипотеза:

Если в процессе обучения алгоритмизации и программированию использовать кинестетические тренажеры, как средства обучения и средства диагностики и контроля результатов обучения, то это будет способствовать развитию алгоритмического мышления и более успешному усвоению темы «Алгоритмизация и программирование» в школьном курсе информатики.

Задачи:

1. Выявить особенности обучения теме «Алгоритмизация и программирование» в школьном курсе информатики;
2. Рассмотреть теоретические основы телесного подхода к теории мышления;

3. Определить условия реализации телесного подхода в обучении программированию;
4. Разработать проекты кинестетических тренажеров по теме «Алгоритмизация и программирование» школьном курсе информатики и рекомендации по их использованию в учебном процессе;
5. Провести апробацию разработанных материалов.

Теоретическая и практическая значимость:

Теоретическая значимость: обоснована возможность и необходимость использования тренажёров в обучении теме «Алгоритмизация и программирование» в школьном курсе информатики;

Практическая значимость: разработан кинестетический тренажер как средство обучения и контроля результатов обучения по теме «Алгоритмизация и программирование» и предложены методические рекомендации по его использованию в школьном курсе информатики.

Методологический аппарат исследования:

Методологическим аппаратом исследования служит телесный подход к теории мышления, а также деятельностный подход к теории и методике обучения информатике .

Методологическая основа исследования:

– фундаментальные труды и научные исследования в области телесного подхода к теории мышления (Р.Бир, Р.Брукс, Т. ванГелдер, Э. Кларк, Ж.Лакофф, П.Маес, Э. Прем, Э. Телен, Ф.Варела, Алюшин, Тхостов, Князева);

– исследования в области теории и методики обучения программированию (Д. Кнут, Э. Дейкстра, Н. Вирт, Б. Страуструп, А.П. Ершов, А.Г. Гейн, А.Г. Кушниренко, М.П. Лапчик, И.Г. Семакин, Е.К.

Хеннер, Н.Д. Угринович, К.Ю. Поляков, В.Е. Жужжалов, С.М. Окулов,
Т.А.Степанова, И.В.Баженова, М.А.Сокольская, Э.А.Нигматулина и др.)

Глава 1. Теоретическое обоснование использования кинестетических тренажеров при обучении программированию

1.1. Особенности обучения теме «Алгоритмизация и программирование» в школьном курсе информатики.

В соответствии с пунктом федерального государственного образовательного стандарта основного общего образования, утвержденного министерством образования Российской Федерации, установлены требования к результатам освоения основной общеобразовательной программы учащимися образовательных учреждений. Обучающийся должен уметь «соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия...». [14]

В современном мире при планировании своей деятельности важно верно выстроить алгоритм своих действий, учесть все возможные пути достижения своей цели. Способность человека к составлению таких алгоритмов составляет основу алгоритмического стиля мышления. Когда говорят об успешном человеке, часто оценивают уровень развития его алгоритмического мышления.. Процесс формирования алгоритмического мышления происходит на уроках математики и информатики, поскольку именно там учащиеся погружаются в алгоритмическую деятельность. И особенно активно алгоритмическое мышление формируется при изучении темы «алгоритмизация и программирование». [3]

Именно поэтому программирование занимает одну из важных ступеней при изучении предмета «информатика». Тема «алгоритмизация и программирование» изучается на базовом уровне в 9 классе, на ее изучение отводится 19 часов. Конечно, данного количества часов недостаточно, чтобы у учащихся сформировалось полное представление об изучаемом языке

программирования. Объем охватываемых тем велик и приходится сталкиваться в проблемами при изучении тем. Учащимся в начале изучения сложно понять, как работает программа, какие шаги выполняются, что происходит с переменными, трудности возникают при считывании данных из файла, при изучении массивов, особенно двумерных, при работе с циклами.

Изучив учебно-методические комплексы разных авторов Угринович Н.Д., Семакин И.Г., Босова Л.Л. можно проследить, что независимо от того какой язык программирования предлагает автор, темы для изучения предлагаются одинаковые. [5],[9],[13]. Следовательно, неважно какой язык программирования изучается важно то, сформируется ли у обучающихся алгоритмическое мышление в процессе изучения темы.

Независимо от того, как представлена линия алгоритмизации и программирования в конкретном УМК, существующие средства обучения – презентации, ментальные карты, исполнители, - нацелены на учащихся с визуальным и аудиальным типами восприятия, задействуют визуальные каналы, а не моторные и тактильные.

Нельзя считать материал усвоенным, если учащийся вызубрил материал. Ведь правильная организация урока обеспечивает понимание материала через его осмысление. В зависимости от особенностей восприятия и обработки информации, людей можно условно разделить на три категории: аудиалы, визуалы и кинестетики.

Большинство людей эффективно пользуются несколькими каналами восприятия, но один из каналов они предпочитают больше. Они проводят в нём больше времени, лучше соображают, и этот способ восприятия для них как бы более важен, чем остальные.

Наиболее распространенный тип – кинестетики, за ними идут визуалы, и аудиалы.

Аудиалы – те люди, кто в основном получает информацию через слуховой канал. Аудиалы хорошо воспринимают вариации голоса

(громкость, паузы, высоту) и нуждаются в озвучивании поставленных передними задач.

Визуалы – это люди, которые «видят» окружающий мир, т.е. воспринимающие большую часть информации с помощью зрения. При работе с визуалами учителю необходимо использовать слова, описывающие цвет, размер, форму, местоположение; выделяя цветом различные пункты или аспекты содержания; записывая действия; используя схемы, таблицы, наглядные пособия и др.[4]

Кинестетики чувствуют окружающий мир и воспринимают большую часть информации через осязание, обоняние, ассоциации, выполняемые действия. Такому человеку, чтобы понять окончательно, необходима деятельность. Получая информацию, воспринимая окружающую действительность, кинестетик стремится всё перевести на язык телесных ощущений, вкуса, осязания и обоняния [1]

Следовательно, необходимо создание новых средств обучения программированию и развития алгоритмического мышления, которые были бы нацелены на кинестетические каналы восприятия и активизацию моторной области памяти.

1.2. Роль телесного подхода при формировании алгоритмического мышления.

Психологи считают, что телесный подход играет немаловажную роль в формировании мышления. Вряд ли у учащегося сформируется полное представление о кубе, о его гранях, ребрах, если он никогда не держал в руках простой кубик.

Телесный подход начал интенсивно развиваться в западной когнитивной науке примерно с начала 1990-х годов. Английское словосочетание «embodied cognition approach» точнее было бы переводить как подход с точки зрения «отелесненности» процесса познания, телесной обличенности всякого познающего существа, кратко говорят «телесный

подход». Среди создателей нового подхода такие ученые, как Р.Бир, Р.Брукс, Т.ван Гелдер, Э. Кларк, Ж.Лакофф, П.Маес, Э. Прем, Э. Телен, Ф.Варела и другие [2]

Французский психолог Морис Мерло-Понти сказал: «Тело живет в мире как сердце в организме», «тело — это наш способ обладания миром». Действительно нельзя понять работу человеческого ума, когнитивные функции человеческого интеллекта, если ум человека абстрагировано от организма, его телесности, определенным образом обусловленных способностей восприятия посредством органов чувств (глаз, уха, носа, языка, рук), организма, включенного в особую ситуацию, экологического окружения, имеющего определенную конфигурацию. Ум существует в теле, а тело существует в мире; телесное существо действует, желает чего-то, воспроизводит себя, мечтает, воображает.[2]

Только в совокупности с практическим действием образ выступает в своей генетической функции, если речь идет о формировании связанных с ним логических операций. Это подтверждается в эксперименте: если модель объекта геометрической задачи сконструирована так, что предполагает возможность практических действий с ней, моделирующих логические гипотезы, наглядный образ действия и его результата способствует становлению общей логической структуры поиска решения и это дает свой эффект — сдвиг по сравнению с мысленным неэффективным решением достигается в 100 %, в то время как зрительное восприятие этой же модели также в 100 % случаев оказывается бесполезным.[8]

Эти опыты очень эффективны. Трудно предположить, что наше восприятие так беспомощно, кажется, что задача вообще непосильна испытуемому. Но вот он берет в руки вещественную модель, производит какие-то пробы и оказывается, что он может решить задачу. Здесь слитный образ объекта и действия с ним может выступать в своей генетической функции независимо от возраста. [6]

Современная школа ориентирована преимущественно на логическую и аудиальную стороны мышления. По мнению профессора психологии Гарвардского университета, учебные заведения созданы для детей с логико-математическим и лингвистическим типом мышления. [7] Однако в современных психологических исследованиях кроме них выделяют телесно-кинестетический, визуально-пространственный, виртуальный типы мышления. Согласно полученным данным, лингвистический и логический типы интеллекта вовсе не преобладают среди молодежи, тем более среди подростков или младших школьников. Получается, что большинство детей разного возраста остается за бортом качественного образования.

В зависимости от особенности интеллекта, психологи дают свои рекомендации. [7] Так, в частности, если у ребенка преобладает телесно-кинестетический интеллект, то обработка информации у ребенка проходит на уровне тактильных сигналов. [11] У такого ученика умелые руки и отличное чувство баланса. Он лучше чувствует себя, когда много времени нужно посвящать физической деятельности, он легко управляет предметами. Для максимально эффективной работы ему необходимо прикасаться, двигаться и обрабатывать поступившую информацию посредством тактильных ощущений. [12]

Если ребенок кинестетик, то он предпочитает учиться, выполняя задания. Самое эффективное обучение для таких детей – эксперимент, а также разработка способов решения поставленной задачи. Представляется, что если строить систему обучения, опираясь на эти советы, процесс обучения программированию будет происходить эффективнее.

1.3. Условия реализации телесного подхода в обучении программированию.

Чтобы реализовать телесный подход в обучении программированию, необходимо создать средства обучения, задействующие тактильные,

моторные ощущения. Мы дали этим средствам название «кинестетические тренажеры».

Интуитивно преподаватели информатики используют в своей педагогической практике телесный подход, пытаются объяснить алгоритм решения задачи с помощью подручных средств, но специальных, теоретически обоснованных разработок в этом направлении нет.

Кинестетические тренажеры для изучения темы «Алгоритмизация и программирование» встречаются довольно редко, примеров таких тренажеров мало, поэтому создавать тренажеры приходилось с нуля. Первый наш шаг в данном направлении был сделан в статье 2014 года «Кинестетические тренажеры как средства развития алгоритмического мышления». В данной статье нами был рассмотрен тренажер для изучения темы «Массивы». [1]

Рассмотрим проект кинестетического тренажера для изучения статического одномерного массива.

Выглядит этот тренажер как коробка с крышкой, имитирующая область памяти, в которой будет располагаться массив. (Рис. 1)



Рисунок 1. Кинестетический тренажер для изучения составных типов данных: статический одномерный массив

На крышке коробки (на «экране») следует написать инструкцию по работе с тренажером, внутри будет поле размещения элементов массива (ячейки памяти). В первом пункте инструкции попросить ученика

определиться с размерностью его массива, задать количество элементов массива N. (Рис.2)

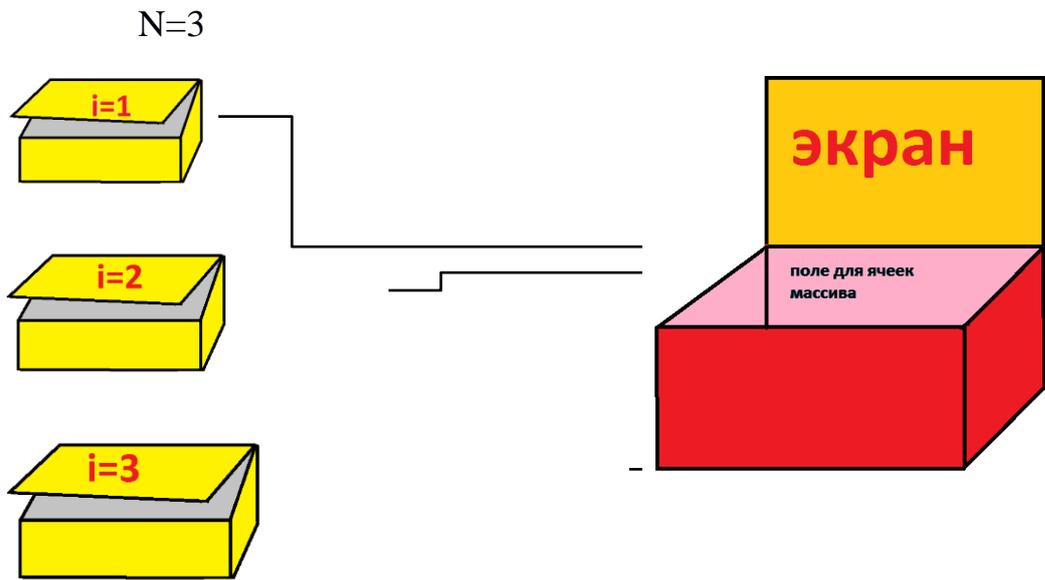


Рисунок 2. Размерность массива

Далее определиться с типом данных, ведь массив это набор однотипных компонентов. Данные заготовлены заранее – цифры на круглых фишках, символы на квадратных и т.д., данные одного типа нанесены на фишки одной формы, чтобы подчеркнуть их однотипность. (Рис.3)

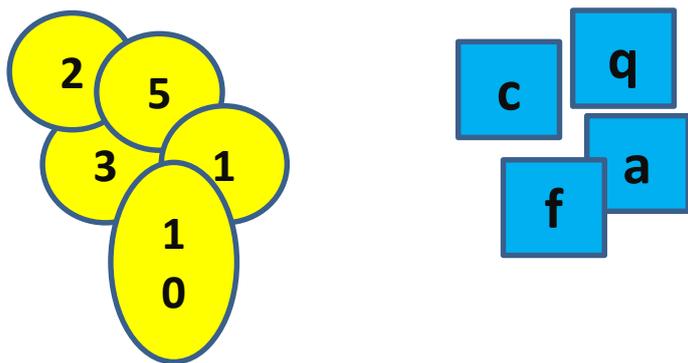


Рисунок 3. Тип данных массива

Созданный массив нужно заполнить элементами, заполнение осуществляется с помощью цикла, поэтому предлагаем по порядку с 1 до N открывать ячейку и класть в нее данные. Здесь важно сделать акцент на том, каким образом, будет заполняться массив: с клавиатуры либо с помощью

генератора случайных чисел. Если с клавиатуры, то ученик берет нужное число, если с помощью генератора, то он берет число наугад. При этом параллельно демонстрируется работа генератора случайных чисел. Действия будут повторяться и здесь усвоится смысл и необходимость применения цикла для заполнения массива.

Примером массивов в жизни может служить набор бильярдных шаров, уложенных на свою подставку. (Рис.4)

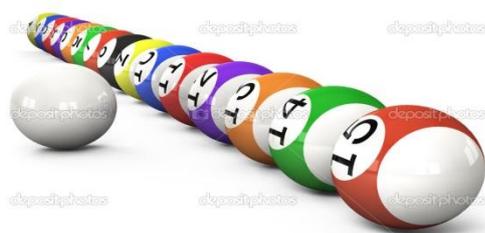


Рисунок 4. Кинестетический тренажер для изучения алгоритмов сортировки массива

На каждом шаре, имитирующем элемент массива, подписан номер – индекс элемента массива, причем этот номер можно стереть и написать заново, поскольку в ходе выполнения алгоритма нам придется менять элементы местами. Шары разного веса. Количество шаров N , для определенности положим $N=10$. Цель – упорядочить шары в порядке убывания веса. Для усиления наглядных ассоциаций, можно окрасить шары массива таким образом, что когда массив будет отсортирован, цвета сложатся в определенной последовательности – радуга или постепенное усиление тона. Имитация выполнения сортировки массива методом пузырька заключается в следующем: поскольку в соответствии с алгоритмом мы можем сравнивать только два соседние элемента массива, ученику предлагается взять в руки два рядом лежащих шара, определить какой из них тяжелее, и если необходимо, поменять их местами, переписав на них индексы. Потом проделать эту процедуру со следующей парой и т.д. до последней пары. Нам придется выполнить $9=N-1$ взвешиваний. Тут наглядно демонстрируется необходимость использования цикла, поскольку

повторяется выполнение одних и тех же действий, и определяется количество повторений $N-1$. После этого самый легкий шар-«пузырек» станет последним - «всплывет на поверхность», но остальные шары останутся неупорядоченными. Для того, чтобы продолжить процедуру, необходимо повторить процесс с самого начала, с первой пары шаров, но выполнять сравнение веса каждой пары нам придется уже не 9, а $8 = N-2$ раз, поскольку последний элемент находится на своем месте – он самый легкий, и нет необходимости взвешивать и сравнивать последнюю пару. После этого второй по легкости шар займет свое место - станет предпоследним – «всплыл» следующий «пузырек» и т.д. Наглядно убедившись, что процесс сравнения пар шаров придется начинать с начала столько раз, сколько элементов в массиве, т.е. N , ученик осознанно понимает необходимость использования конструкции из двух вложенных циклов для реализации алгоритма сортировки. Причем, поскольку в каждом следующем случае приходится сравнивать все меньшее и меньшее количество пар, ему становится понятно, почему индекс вложенного цикла (допустим j) меняется не до N , а до $N-i$, где i - индекс внешнего цикла, который меняется от 1 до N – столько раз, сколько приходится начинать сначала.

Практика использования этой модели показывает, что после выполнения алгоритма вручную, текст программы, реализующей этот алгоритм, становится понятен абсолютно всем ученикам.

Подобный тренажер можно использовать и для наглядной иллюстрации более сложных алгоритмов сортировки – сортировки Хоара, битовой или разрядной сортировки, которые бывают непонятны после первого, словестного объяснения, даже сильным ученикам.

Далее, один из сложных для понимания моментов – принципы работы с динамическими типами данных. Отчасти эти сложности обусловлены тем, что при работе со статическими типами данных – допустим, с массивами, у обучающихся возникают определенные стереотипы. Например, что по индексу можно обратиться напрямую к

любому элементу, чего нельзя сделать при работе с такими динамическими типами данных как стеки и очереди. Исходя из этого кинестетический тренажер может выглядеть как тубус, в котором хранятся мячи для игры в теннис (Рис.5).



Рисунок 5. Кинестетический тренажер для изучения динамического типа данных стек.

Чтобы проиллюстрировать особенности динамических типов данных преподаватели в качестве примера приводят тубус с теннисными мячами.

Стек – это однонаправленный список, в котором все включения и удаления выполняются с конца списка. Он работает по принципу LIFO (last-in, first-out; «последним пришел – первым вышел»). Его моделью может быть такой тубус для мячей, который открывается только с одного конца. Хорошо, если мячи будут разного цвета, и ученики будут наглядно видеть, что если мы первым опустили в тубус красный мячик, а затем желтый, зеленый и др., то для того, чтобы достать красный (и почитать, что на нем написано, например), нужно выкатить все шары, которые были помещены в стек позже.

Очередь – это однонаправленный список, в котором удаление элементов происходит из начала, а включение новых элементов – в конце. В этом случае работает принцип FIFO (first-in, first-out; «первым пришел – первым вышел»). Его моделью может быть такой тубус для мячей, который

открывается с двух сторон, но с каждой стороны установлены анти клапаны, работающие в только в одном направлении – позволяющие только выкатывать мячи из тубуса в начале и только закатывать мячи в тубус в конце.

Так же в пример можно привести разработку кинестетического тренажера студентами педагогического университета по теме «Динамические структуры данных» вузовского курса информатики. [15]

Для имитации динамического выделения памяти используют кармашки, нашитые на ткань или другое более плотное полотно. Это необходимо, чтобы информация, помещенная в кармашек, становилась невидима для глаз ученика. Все кармашки пронумерованы различными адресами (Рис. 6).

Указатели представлены в виде карточек. Их размер выбран так, чтобы они были видны, т.е часть карточки выглядывает из кармашка. Нам необходимо два указателя: *head* (определяет вершину стека) и *tmp* (вспомогательный указатель).

Различные области памяти узла стека мы представили в виде карточек разного цвета. Данные карточки должны полностью помещаться в кармашки.

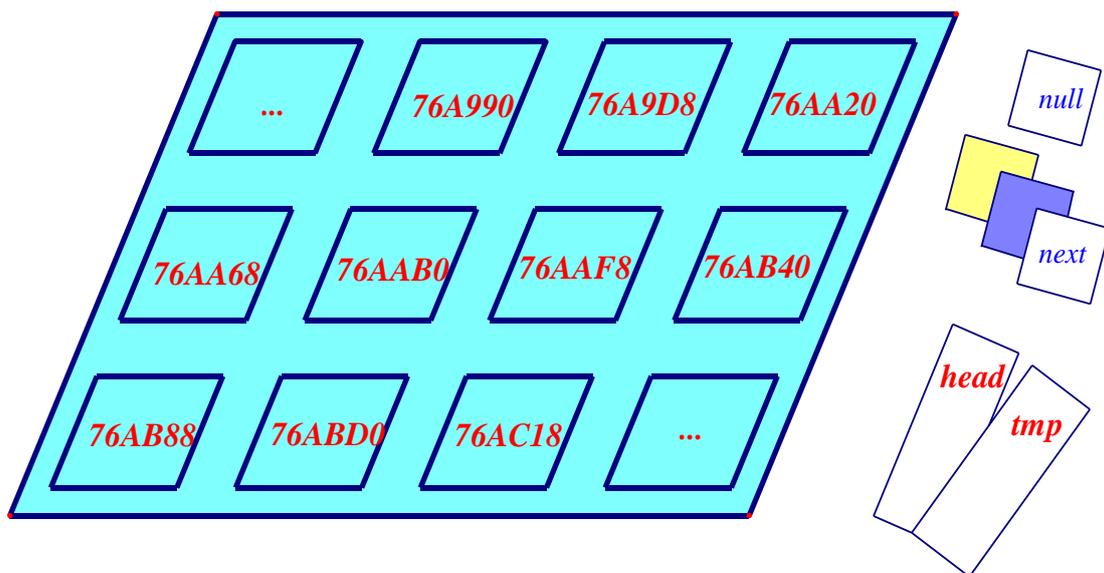


Рисунок 6. Кинестетический тренажер для изучения динамического типа данных стек.

К тренажеру необходимо приложить инструкцию. В первом пункте инструкции ученика просят определиться с типами данных, которые будут располагаться в каждом узле стека. За каждым типом данных зарезервирован свой цвет карточки. Белые карточки будут обозначать указатели next, которые отвечают за связь узлов стека между собой.

Во втором пункте написана часть кода программы, которая отвечает за формирование стека. Здесь предложено повторить данный код при помощи тренажера.

Выделение памяти для указателя tmp при заполнение узла стека данными (Рис. 7):

- a) `stek *head = NULL` – указатель head объединяется с карточкой null;
- b) `stek *tmp` – в работу вступает указатель tmp;
- c) `tmp = newstek` – указатель tmp помещается в произвольный кармашек(1).
- d) Соответствующие карточки заполняются информацией выбранного типа данных и помещаются в кармашек (2).

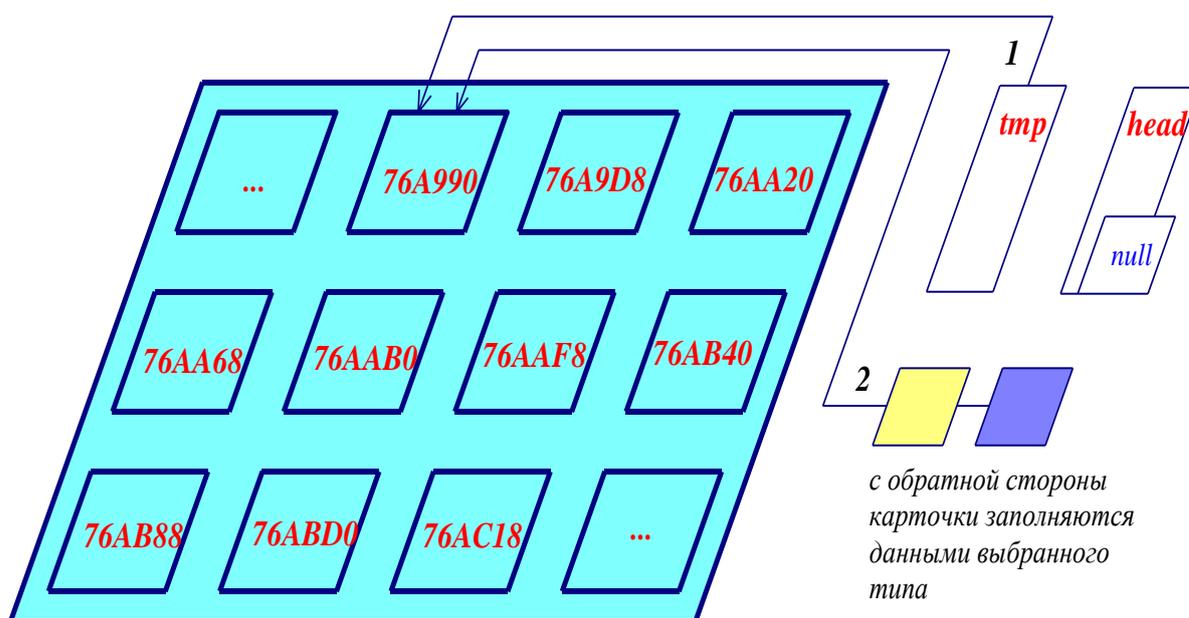


Рисунок 7. Выделение памяти для указателя tmp при заполнение узла стека данными

Перенаправление указателя нового звена на начало стека head и направление указателя - головы списка на созданное звено (Рис. 3):

- e) $tmp \rightarrow next = head$ – карточка null помещается в кармашек к $tmp(3)$;
- f) $head = tmp$ – карточка $head$ помещается к tmp (4);

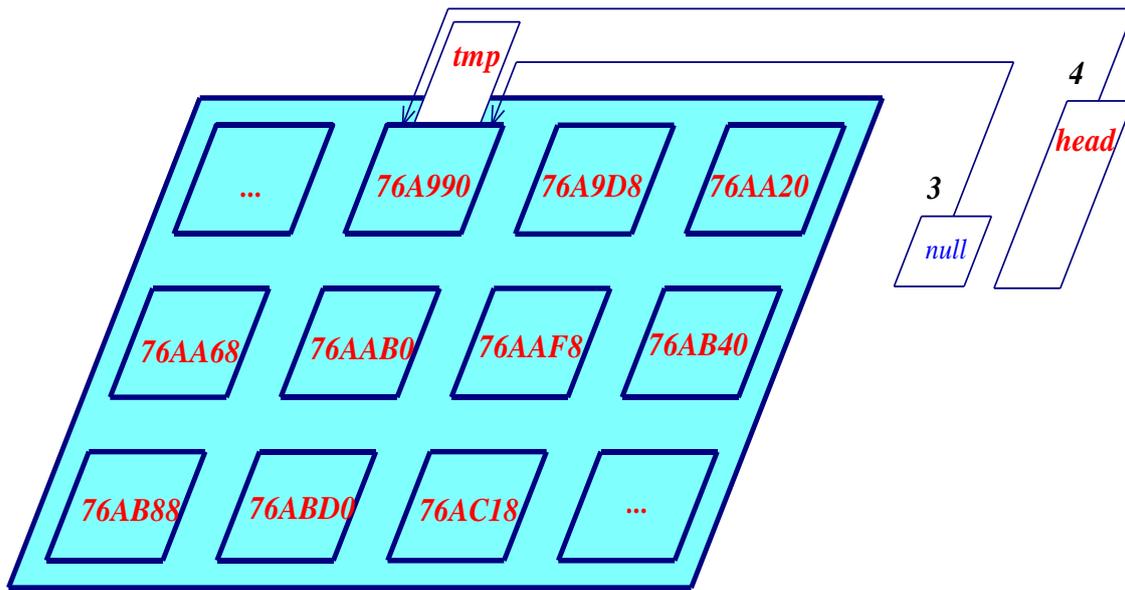


Рисунок 8. Перенаправление указателя нового звена на начало стека и направление указателя - головы списка на созданное звено

Так как процесс происходит в цикле, снова выделяется память для указателя tmp (Рис. 9):

- a) $tmp = newstek$ – указатель tmp перекладывается в новый кармашек (1);
- b) разноцветные карточки заполняются данными и помещаются в кармашек с указателем tmp (2);
- c) $tmp \rightarrow next = head$ – указатель нового звена перенаправляется на адрес указателя $head$ (3)
- d) $head = tmp$ – указатель $head$ направляется на созданное звено(4).

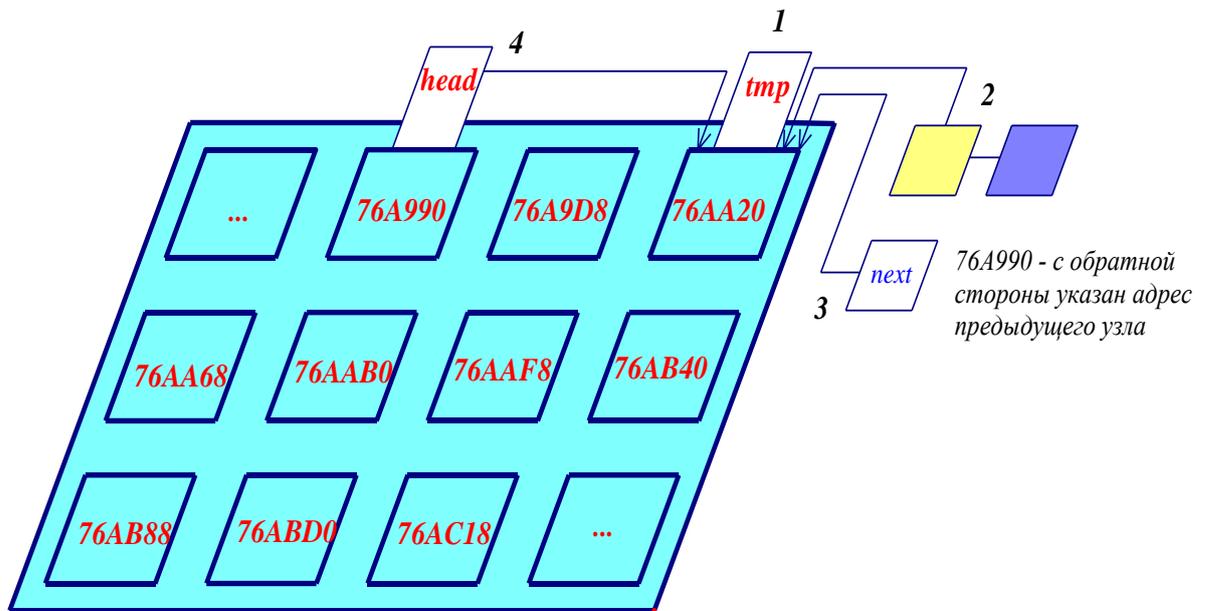


Рисунок 9. Создание нового звена стека

Когда студент сформирует и заполнит стек, указатель `head` будет находиться в кармашке с последними данными. Связь с предыдущим узлом будет только при помощи указателя `next`, помещенного в этот кармашек.

В третьем пункте может быть предложено посмотреть содержимое стека и найти его определенный элемент при помощи кода программы. Затем в четвертом пункте необходимо будет удалить стек.

Данный тренажер можно использовать и для таких динамических структур, как очереди и деки. Для этого необходимо будет просто заготовить дополнительные указатели. (Рис. 10).



Рисунок 10. Кинестетический тренажер для изучения динамического типа данных стек

Представляется, что использование в учебном процессе подобных кинестетических тренажеров позволит повысить уровень понимания и усвоения учебного материала по алгоритмизации и программированию, будет способствовать процессу развития алгоритмического стиля мышления.

Глава 2. Практические аспекты использования кинестетических тренажеров при обучении программированию.

2.1. Проектирование кинестетических тренажеров по теме «Циклы».

Тренажеры, созданные для обучения алгоритмизации и программирования, направлены выполнять основную функцию – понимание. Отрабатывая тот или иной алгоритм любой из тем, дети учатся понимать сложные вещи через игру. Чаще всего педагоги вводят на уроке понятие блок-схемы, словестно, объясняют как она работает и создают программный код задачи. Как правило, проблема заключается в том, что обучаемые не понимают, как работает блок-схема и соответственно написанная ими программа не работает.

В программировании циклы подразделяются на три типа.

Цикл с параметром «Для» используется тогда, когда известно, какое число повторений необходимо выполнить. Перед тем, как задать тело цикла, ученик должен написать число повторений, которое необходимо выполнить программе.

Цикл с условием используется, если заранее неизвестно, какое число повторений нужно выполнить. В этом случае количество повторений тела цикла зависит от истинности условия. Как только условие примет значение «ложь», выполнение цикла закончится.

Цикл с условием подразделяется на два вида: цикл с предусловием «Пока» и цикл с постусловием «До». При выполнении цикла «Пока» условие находится перед телом цикла. В цикле «До» условие находится после тела цикла. Таким образом, при решении задачи с циклом «До» тело цикла выполнится минимум один раз.

У обучающихся возникают трудности при изучении темы «Циклы»:

- учащиеся плохо представляют границы параметра, при изучении цикла с параметром;

- не могут правильно определить начальное и конечное значение параметра цикла;
- плохо представляют, как меняется параметр;
- не могут правильно составить условие, для цикла с пред условием;
- упускают из виду, что значение переменной должно меняться в теле цикла;
- иногда просто забывают заключить в операторные скобки тело цикла, если оно состоит из более чем одного оператора.

Особенно такие ошибки допускают дети с кинестетическим каналом восприятия. В силу своих особенностей они не могут воспринимать информацию, как большинство детей в классе. Поэтому учителю приходится прибегать к дополнительным средствам и методам обучения

Выявив все эти проблемы, я решила, начать знакомство с тренажером при изучении темы «Циклы».

Свой проект я решила оформить в виде составных частей блок-схемы.

(Рис. 11)

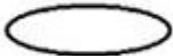
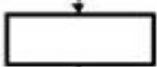
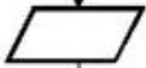
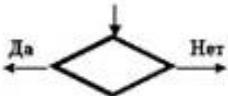
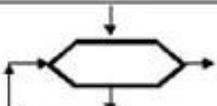
№	фигура	Что обозначает
1		Начало Конец
2		Действия
3		Ввод и вывод
4		Условие (если, циклы с условием)
5		Цикл с параметром (для)

Рисунок 11. Элементы блок-схемы и обозначение каждого элемента

Работает этот тренажер довольно просто, на маркерной доске, находятся все элементы блок-схемы с кармашками на каждом элементе. Элементы фиксируются магнитами. Кармашки нужны для того, можно было пронести переменную через весь цикл и проследить, что с ней станет. На оборотной стороне каждого элемента указано, что он обозначает, для того, чтобы обучающему было понятно стоит брать этот элемент или нет. Для тренажера следует создать инструкцию, придерживаясь, которой можно с легкостью составить алгоритм.

Итак, учащийся должен для начала определиться с типом цикла. В коробочке из-под тренажера уже заготовлена карточка с характеристиками и примерами, всех трех типов цикла: цикл с постусловием, с предусловием, с параметром. (Рис. 12)

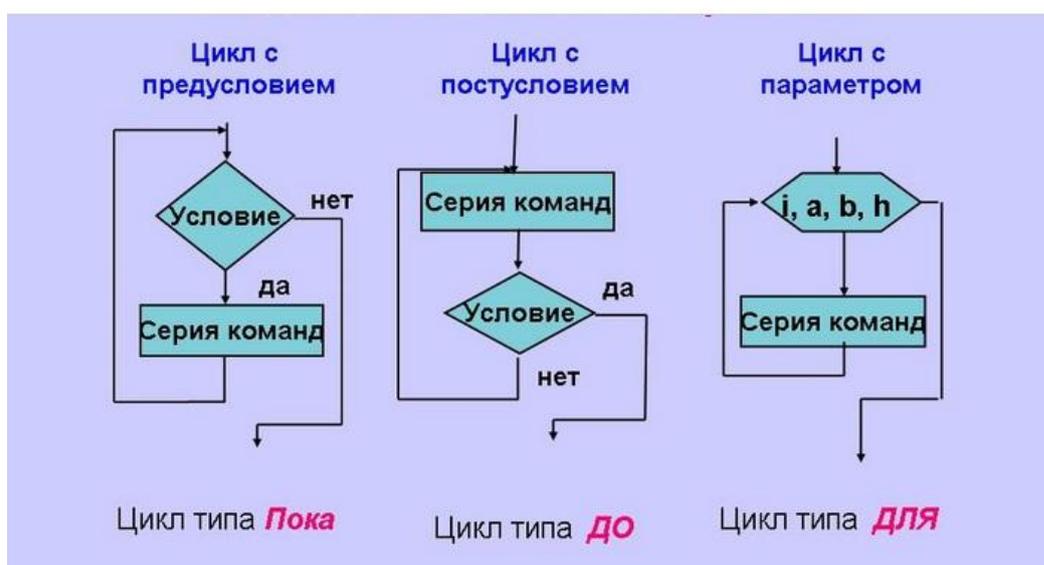


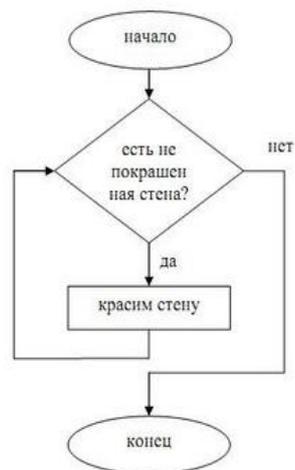
Рисунок 12. Формат циклов в программировании

Так же в коробочке имеются примеры всех трех типов цикла. (Рис. 13)

**ПРИМЕР ЦИКЛА С ПОСТУСЛОВИЕМ.
СТИРКА.**



**ПРИМЕР ЦИКЛА С ПРЕДУСЛОВИЕМ.
РЕМОНТ**



**ПРИМЕР ЦИКЛА С ПАРАМЕТРОМ:
ПРИШИВАЕМ ПУГОВИЦЫ**

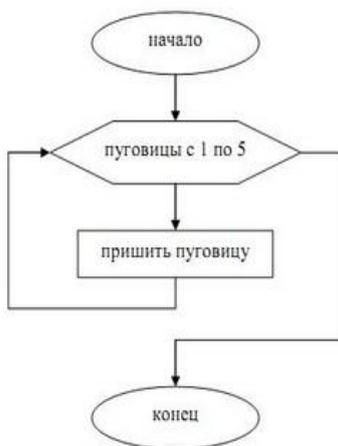


Рисунок 13. Примеры циклов: с параметром, предусловием, постусловием

После выбора типа, учащийся задает начальные значения переменным, и, конечно, же тип, следим чтобы значения были не сильно большими, чтобы можно было устно проверить правильность блок-схемы, это будет проще как

для вас, так и для обучающегося. Далее идет расстановка всех элементов цикла по примеру на карточке, и дорисовываются стрелочки на доске маркером. После чего учащийся пишет для каждого элемента, у которого есть карман, какое действие там должно выполняться, писать лучше маркером слева или справа от элемента, чтобы элемент можно было использовать много раз. После чего учащийся прогоняет свою переменную через каждый карман, меняя ее, если требуется. Когда значение меняется, обучающийся берет чистый кружочек и подписывает уже новое значение и ведет переменную дальше, до самого конца, проходя стрелочки нужно количество раз, если цикл выполняется верно и все вычисления удовлетворяют условию задачи, то можно считать, что задача решена, остается перевести алгоритм на язык программирования. (Рис. 14)

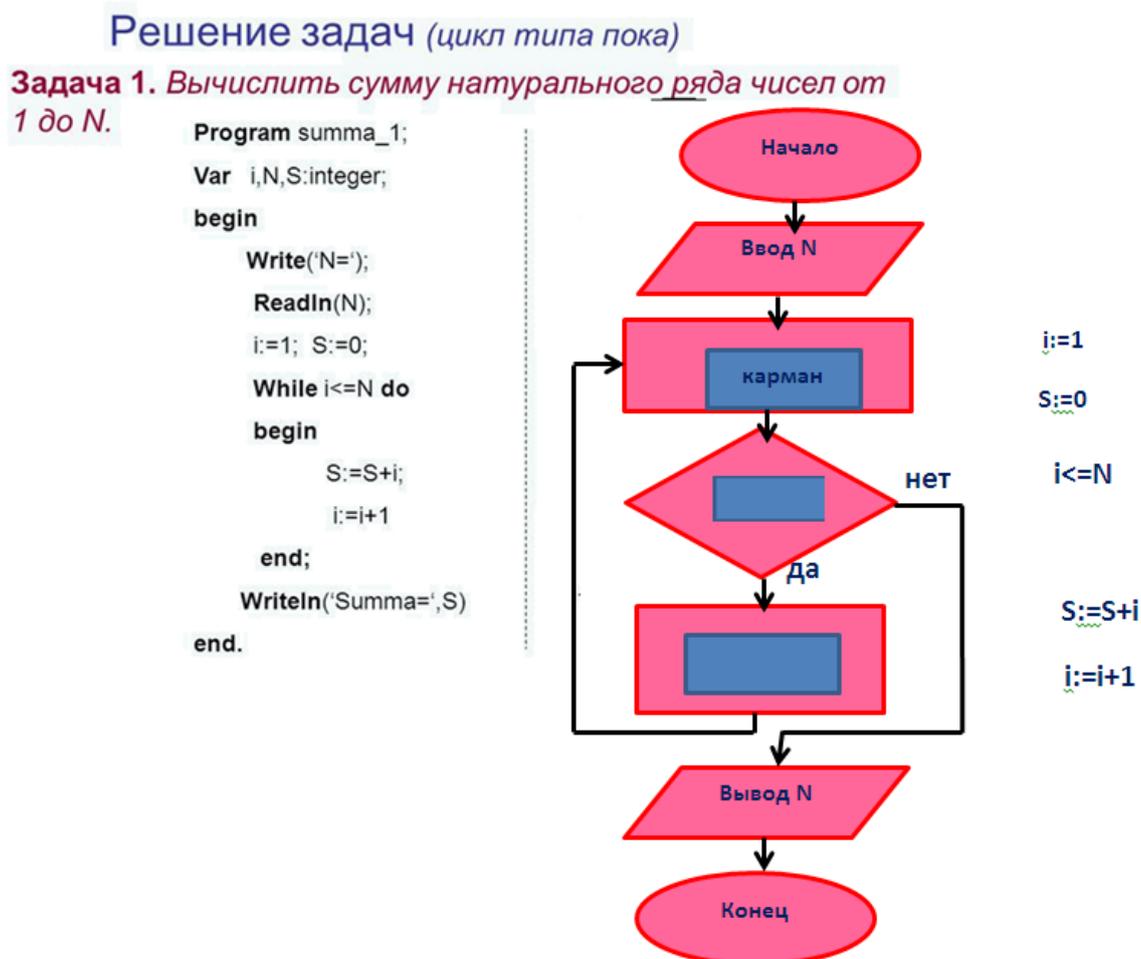


Рисунок 14. Пример решения задачи на маркерной доске с написанием кода программы.

Работать с тренажером можно в группах. Тогда стоит в группе четко придерживаться инструкции и проводить самооценку группы на каждом этапе, для того чтобы можно было отследить какой этап работы самый трудный для понимания.

2.2. Методические рекомендации по использованию кинестетических тренажеров при обучении теме «Циклы».

Язык программирования Паскаль очень трудно объяснять детям, так как это сложно для них. Следовательно, стоит внимательно разобрать с детьми циклы всех трех типов. В этом случае вам очень поможет кинестетический тренажер. Стоит внимательно отнестись ко всем действиям учащегося, задавайте наводящие вопросы, которые потом интуитивно он будет задавать себе сам. Например, какой тип данных мы будем использовать? Как его задать? Какой цикл мы выберем? Почему? Как выглядит этот цикл? Что является телом цикла? Просить вводить переменные отличные от заданного типа, пробовать вводить разные числа: положительные, отрицательные, ноль и т.д.

Все эти вопросы помогут ему не только правильно составить блок-схему, но и правильно оформить цикл, увидеть его границы (begin, end). Не стоит исправлять учащегося, если он неправильно расставил элементы блок-схемы, когда он начнет прогонять переменную по всем кармашком, он обязательно увидит ошибку и исправит ее. Проходя каждый этап просите учащегося комментировать себе или группе почему он понял значение и каким оно стало, в ходе этих рассуждений можно понять где учащийся ошибается. Лучше всего эталон работы с тренажером показывать на весь класс фронтально, после можно разбить учащийся на группы и дать задание

составить блок-схему группой, группа должна четко следовать инструкции, придерживаться эталону, задавать вопросы друг другу, спорить, комментировать. Группе следует выставлять себе оценку на каждом этапе, чтобы увидеть какой этап был наиболее сложным, где что не получилось. Преподавателю стоит отследить оценки учащегося на каждом этапе и определить какому этапу стоит уделить большее внимание.

Конечно же, при написании кода программы обратите внимание на пунктуацию, в Паскале программа не будет работать, если учащийся упустит хотя бы один знак. После окончания работы следует обговорить каждый знак препинания, и ответить на вопрос с какой целью он поставлен.

2.3.Результаты апробации

За время работы в МБОУ «Кириковская средняя школа» были апробированы разработанные материалы, а именно, конспект урока на закрепление знаний по теме "Циклы" [Приложение А] и разработан кинестетический тренажёр по теме «Циклы».

Урок информатики по теме «Циклы» был проведен в 9 классе в 2015-2016 и 2016-2017 учебном году. В 2015-2016 учебном году на уроке информатике в 9 классе не применялись тренажеры, оценки учащихся снижались по причине того что, обучающиеся не могли в полной мере правильно создать программный код задачи. В ходе работы часто у учащихся были ошибки в знаках препинания, и появлялись серьезные ошибки, когда программа зацикливалась.

В 2016-2017 учебном году был сделан вывод по работе прошлого года и решено ввести кинестетические тренажеры, пока что при изучении темы «циклы».

Выявлено, что после применения кинестетического тренажера на уроке информатики в 9 классе, у обучающихся значительно повысился уровень успеваемости. (Табл.1) (Табл.2)

Таблица 1. Успеваемость обучающихся по информатике в 9 классе за 2015-2016 учебный год.

№	Фамилия, Имя	Оценка
1.	Вебер Алина	3
2.	Вебер Марина	4
3.	Галяутдинов Сергей	3
4.	Мамонтова Алина	3
5.	Осипов Николай	4
6.	Шилина Лилия	3
7.	Яр-мухамедов Валентин	3

Таблица 2. Успеваемость обучающихся по информатике в 9 классе за 2016-2017 учебный год.

№	Фамилия, Имя	Оценка
1.	Данилова Любовь	5
2.	Лукин Егор	4
3.	Мифтахутдинов Рамиль	3
4.	Сластикина Олеся	5
5.	Стифанов Дмитрий	3
6.	Фильков Сергей	3

В ходе уроков были проведены рефлексии, которые помогли определить отношение учащихся к новому тренажеру. У учащихся повысился интерес к программированию, обучающиеся с интересом создавали блок-схемы и код программы, активно обсуждали в парах программу, и применяли тренажеры. В ходе работы мною был виден прогресс не только в качественном исполнении работы, но и в умении учащимися выстраивать ход работы, рассуждать, задавать себе и другим вопросы, приходить к единой мысли. По итогам данной работы можно смело судить, что у обучающихся повысился уровень алгоритмического мышления.

Проведенная апробация позволила сделать вывод о целесообразности применения кинестетических тренажеров к обучению теме «Алгоритмизация и программирование» в школьном курсе информатики.

Заключение

В ходе проведенной работы цель была достигнута: мы теоретически обосновали возможность и необходимость использования кинестетических тренажеров при обучении программированию, спроектировали примеры таких тренажеров и методические рекомендации по их использованию в учебном процессе.

Анализ научной литературы позволил определить теоретические основы телесного подхода к обучению программированию.

Был проанализирован процесс формирования и развития алгоритмического стиля мышления школьников. Анализ позволяет сделать вывод, что развитие алгоритмического мышления школьников, в основном происходит на уроках информатики и математики, поскольку именно там учащиеся погружаются в алгоритмическую деятельность. И особенно активизируется этот процесс конечно же при изучении алгоритмизации и программирования.

Разработан кинестетический тренажер по теме «Циклы» и методика его применения для уроках введения новых знаний, закрепления пройденного материала, а так же для контроля результатов обучения.

В период прохождения педагогической практики была проведена апробация разработанных материалов и проведен анализ ее результатов, акт о внедрении имеется в приложении.

В результате проведенного исследования можно сделать следующие выводы:

- применение кинестетических тренажеров в обучении является возможным и нужным, так как они развивают мышление, используя моторику и тактильно восприятие мира.

- использование тренажеров в процессе объяснения обучающимся основных алгоритмических структур позволяет повысить эффективность обучения программированию и уровень алгоритмического мышления, делает изложение учебного материала по этим темам более живым и наглядным, способствует более успешному формированию у них алгоритмического мышления.

Список литературы

1. Алексеева Д.В., Степанова Т.А., Хайбрахманова А.Т. Кинестетические тренажеры как средства развития алгоритмического мышления // Материалы XIV Всероссийской (с международным участием) научно-практической конференции студентов, аспирантов и молодых ученых «Молодежь и наука» – Красноярск, 2014.
2. Алюшин А.Л., Князева Е.Н. Телесный подход в когнитивной науке // Философские науки, 2009. № 2. С.106-126.
3. Беляев М.В. Алгоритмическое мышление как цель современного образования //Сборник материалов Международная региональная конференция ЮНЕСКО «Экология человека: взаимодействие культуры и образования в современных условиях». ч.1, Новосибирск, 1998. С.56
4. Борисёнок И. Аудиалы, Визуалы, кинестетики и дигиталы URL: <http://vseklass.ru/>
5. Босова Л.Л., Босова А.Ю. Информатики и ИКТ, учебник 9 класса, в 2 ч., 2015 г. URL: <http://lbz.ru/books/576/7400/>
6. Гурова Л.Л. Психология мышления.– М.: ПЕР СЭ, 2005 с. 30-33
7. Если у ребенка кинестетический тип интеллекта. [Электронный ресурс] URL: <http://udoktora.net/esli-u-rebenka-kinesteticheskiy-tip-intellekta-62484/> (10.05.2017г)
8. Калитина В.В., Пушкарева Т.П., Степанова Т. А. Алгоритмические ментальные карты как эффективное средство обучения программированию // Сборник статей Международной научно-практической конференции «Фундаментальные и прикладные научные исследования», Москва, 2015. С. 179.
9. Семакин И.Г., Заголова Л.А., Русаков С.В. и др. Информатика 9 класс, 2015г. URL: <http://lbz.ru/books/577/8005/>

10. Степанова Т.А. Сущность алгоритмического мышления с позиции информационного подхода // Инновации в непрерывном образовании, 2012, №3. С.95.
11. Нигматулина Э.А., Степанова Т.А. Телесный подход к обучению программированию // Перспективы и вызовы информационного общества: материалы IV Всероссийской научно-практической конференции с международным участием. Красноярск, 12 ноября 2015 г. [Электронное ресурс] / КГПУ им. В.П. Астафьева. – Красноярск, 2015. С. 131-1366).
12. Тхостов А.Ш. Психология телесности. — М.: Смысл, 2002.
13. Угринович Н.Д. Информатика 9 класс. 2015г. URL: <http://lbz.ru/books/578/8026/>
14. Федеральный государственный образовательный стандарт основного образования от 17.12.2010 №1897
15. Чепикова Е.Ю. Разработка кинестетического тренажера по теме «Динамические структуры данных» курса «Языки и методы программирования» // Материалы конференции "Актуальные проблемы информатики и информационных технологий в образовании: материалы Всероссийской научно-практической конференции с международным участием в рамках XVIII международного научно-практического форума студентов, аспирантов и молодых ученых «Молодежь и наука XXI века». Красноярск, 23 мая 2017 г. [Электронный ресурс] / ред. кол.; отв. ред. П.С. Ломаско. – Электрон. дан. / Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2017.

Приложения

Приложение А

План – конспект урока по информатике (учебник Семакин И.Г. 9 класс)

Тема урока: Программирование циклов

Цель урока:

Образовательная: закрепление знаний учащихся о цикле с предусловием и его формой записи на Паскале.

Развивающая: формирование умения записи цикла с предусловием в среде программирования Паскаль; развитие логического мышления.

Воспитательная: повышение мотивации к изучению предмета информатика.

Оборудование: компьютерный класс с установленным ПО, магнитная доска, кинестетический тренажер, ватман, маркеры, проектор, экран.

Тип урока: урок закрепления знания.

Формы учебной работы учащихся: работа в группах

План урока:

1. Организационный момент (1-2 мин).
2. Актуализация деятельности учащихся (5 мин).
2. Закрепление изученного материала. (5 мин).
3. Выполнение практической работы в парах (25-30 мин).
4. Подведение итогов работы (3 мин).

Ход урока.

1. Организационный момент

2. Актуализация деятельности учащихся

Фронтальный опрос:

1. С какого служебного слова начинается цикл с предусловием?	while
2. Какое служебное слово следует после условия?	do
3. Дан фрагмент программы: n:= 0; a:= 0; while n <9 do begin n:=n+1; a:=10*n; writeln(n, ' ',a); end; - Какие значения должна принимать переменная n , чтобы цикл выполнялся?	От 0 до 8
4. При while 5<9 do чему будут равны переменные n и a в теле цикла?	n = 6, a = 60
5. При каком значении переменной n ,	При n = 9

программа завершится?	
6. Перед каким служебным словом не ставится точка с запятой?	Перед do

3. Закрепление изученного материала.

- Для выполнения сегодняшней практической работы, нам необходимо вспомнить, как мы работаем с кинестетическим тренажёром. Поэтому вам предстоит работа в парах. Вам следует придумать одну задачу, посмотрим какая пара справится с этим заданием быстрее. Не забываем, что от вас помимо блок-схемы, требуется еще и готовая программа.

4. Выполнение самостоятельной практической работы.

Сегодняшняя ваша практическая работа заключается в том, что вам нужно будет разбиться на пары и совместно придумать задачу по теме "Цикл с предусловием" и разработать для этой задачи блок-схему с помощью тренажера и написать программу.

5. Домашнее задание: Составить блок-схему для цикла с предусловием из вашей жизни.

6. Подведение итогов занятия.

Рефлексия

Ответьте на вопросы:

1. Какой этап для вас самый сложный?
2. Сможете ли вы создать код программы без кинестетического тренажера или блок-схемы?
3. Как вы считаете насколько процентов вы оцениваете понимание вами данной темы?

План – конспект урока по информатике (учебник Семакин И.Г. 9 класс)

Тема урока: Программирование циклов

Цель урока:

Образовательная: ознакомление учащихся с циклом с предусловием и его формой записи на Паскале.

Развивающая: формирование умения записи цикла с предусловием в среде программирования Паскаль; развитие логического мышления.

Воспитательная: повышение мотивации к изучению предмета информатика.

Оборудование: компьютерный класс с установленным ПО, кинестетический тренажер, магнитная доска, маркер, проектор, экран.

Тип урока: урок введения нового знания.

Формы учебной работы учащихся: фронтальная работа, индивидуальная работа учащихся.

План урока:

1. Организационный момент (1-2 мин).
2. Актуализация деятельности учащихся (5 мин).
2. Объяснение нового материала (15-20 мин).
3. Закрепление изученного материала. Выполнение практической работы (15 мин).
4. Подведение итогов работы (3 мин).

Ход урока.

1. Организационный момент

2. Актуализация деятельности учащихся

Фронтальный опрос:

1. Как записывается заголовок программы на Паскале?	Program
2. Как записывается раздел описания переменных?	Var
3. С какими типами числовых величин работает Паскаль?	Целые и вещественные
4. Как записывается оператор присваивания?	:=
5. Как записывается оператор ввода и вывода в Паскале?	Read (readln), write (writeln)
6. С какого служебного слова начинается условный оператор?	if
7. Какое служебное слово отвечает за истинность условия?	then
8. Какое служебное слово отвечает за истинность условия?	else
9. Обязательными ли являются в структуре условия слова then и else?	then - обязательное, else - необязательное
10. Отличие полной записи оператора условия от неполной?	В полной then и else, а в неполной только then
11. Перед какими служебными словами не	Перед if и then

ставится точка с запятой?	
---------------------------	--

3. Объяснение нового материала.

Цикл с заданным условием продолжения работы (цикл - ПОКА) программируется в языке Паскаль с помощью оператора `while`.

Формат записи оператора: *(учитель достает из коробочки карточку и показывает формат записи цикла)*

while логическое_условие do оператор;	пока логическое_условие верно то выполняется оператор ,
--	--

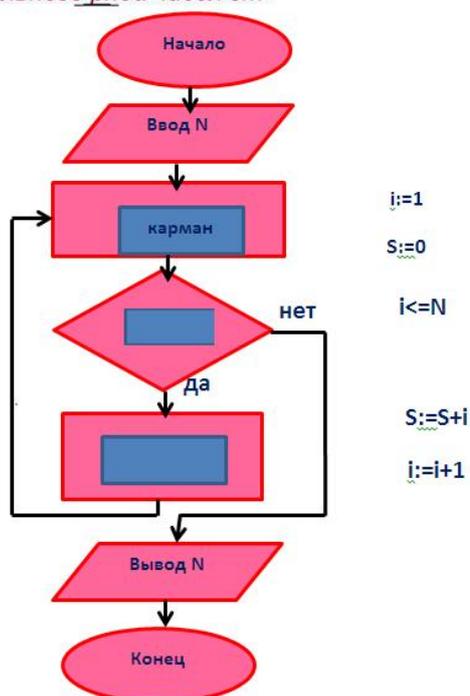
Перед `do` точка с запятой не ставится!

В качестве примера рассмотрим следующую задачу:

Решение задач (цикл типа пока)

Задача 1. Вычислить сумму натурального ряда чисел от 1 до N.

```
Program summa_1;  
Var i,N,S:integer;  
begin  
  Write('N=');  
  Readln(N);  
  i:=1; S:=0;  
  While i<=N do  
  begin  
    S:=S+i;  
    i:=i+1  
  end;  
  Writeln('Summa=',S)  
end.
```



Объясняю каждый элемент и прогоняю переменные разного рода по кармашкам.

4. Выполнение самостоятельной практической работы.

1. В первый день спортсмен пробегает 10 км, а в каждый следующий на 2 км больше, чем в предыдущий. Сколько км. пробежит спортсмен за неделю?

2. В первый день спортсмен проплывает 5 км, а в каждый следующий на 1 км больше, чем в предыдущий. Сколько км. проплывет спортсмен за 5 дней?

5. Домашнее задание: стр. 210 № 1

6. Подведение итогов занятия. (Оценить работы учащихся.)