

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 математики, физики и информатики

Выпускающая базовая кафедра информатики и информационных технологий
в образовании

Голубцова Анна Викторовна
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема «Обучение алгоритмизации и программированию в школьном курсе
информатики на основе ментального подхода (на примере темы
«Алгоритмическая структура «Ветвление»»)»

Направление подготовки 44.03.05 Педагогическое образование

Направленность (профиль) образовательной программы математика и
информатика

ДОПУСКАЮ К ЗАЩИТЕ
Зав. базовой кафедрой
информатики и информационных
технологий в образовании,
профессор, д.п.н.,
кандидат физико-математических наук
Н.И.Пак
« ____ » июня 2017

Руководитель
к.п.н., доцент базовой кафедрой
информатики и информационных
технологий в образовании
Т.А. Степанова

Дата защиты « ____ » июня 2017
Обучающийся Голубцова А.В.
« ____ » июня 2017 _____
Оценка _____

Красноярск

2017

Оглавление

Введение.....	4
Глава 1. Теоретические основы ментального подхода к обучению теме «Алгоритмизация и программирование» в школьном курсе информатики	7
§ 1.1. Особенности обучения теме «Алгоритмизация и программирование» в школьном курсе информатики	7
§ 1.2. Сущность ментального подхода к обучению программированию	15
Глава 2. Условия реализации ментального подхода к обучению теме «Алгоритмизация и программирование» в школьном курсе информатики ...	21
§ 2.1. Процесс формирования и развития алгоритмического стиля мышления у обучающихся.....	21
§ 2.2. Разработка алгоритмической ментальной карты по теме «Алгоритмическая структура ветвление».....	32
§ 2.3. Диагностика уровня сформированности алгоритмического стиля мышления	37
§ 2.4. Результаты апробации.....	42
Заключение	46
Список литературы	49
Приложения	53
Приложение А.....	53
Приложение Б.....	55
Приложение В	59
Приложение Г.....	61

Введение

Актуальность:

Опыт изучения программирования и обучения программированию показывает, что основные трудности у учащихся возникают не при изучении синтаксиса и основных конструкций языка программирования, а на первом этапе решения задач по программированию, на этапе алгоритмизации. И связаны эти трудности с недостаточным уровнем сформированности алгоритмического мышления обучающихся, с их неготовностью воспринимать материал достаточно высокого уровня абстракции и логики.

Следовательно, для разработки методических приемов, позволяющих повысить эффективность обучения и успешность изучения программирования, необходимо обратиться к исследованию процессов мышления.

Ментальный подход к обучению как раз и предполагает смещение целеполагания учебного процесса в сторону развития когнитивных способностей обучающихся, в частности, если рассматривать обучение программированию – то основной целью будет являться развитие алгоритмического стиля мышления. В рамках ментального подхода к обучению программированию предполагается использование методики ментальных карт как средства развития алгоритмического мышления.

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.

Цель:

Теоретически обосновать возможность и необходимость ментального подхода к обучению теме «Алгоритмизация и программирование»,

определить и разработать необходимые для его реализации методы и средства обучения.

Объект:

Обучение теме «Алгоритмизация и программирование» в школьном курсе информатики.

Предмет:

Обучение теме «Алгоритмизация и программирование» в школьном курсе информатики на основе ментального подхода.

Гипотеза:

Если в процессе обучения алгоритмизации и программированию использовать информационные ресурсы ментального типа (ментальные карты) как средства обучения и средства диагностики и контроля результатов обучения, то это будет способствовать развитию алгоритмического мышления и более успешному усвоению данной темы.

Задачи:

- 1) Выявить особенности обучения теме «Алгоритмизация и программирование» в школьном курсе информатики (ШКИ);
- 2) Рассмотреть теоретические основы ментального подхода к обучению программированию;
- 3) Уточнить понятие алгоритмического стиля мышления (АСМ), построить его информационную модель;
- 4) Проанализировать процесс формирования и развития АСМ, рассмотреть средства и методы его развития и определить способы его диагностики;
- 5) Разработать ментальные карты по теме «Алгоритмизация и программирование» ШКИ и рекомендации по их использованию в учебном процессе;
- 6) Провести апробацию разработанных материалов.

Теоретическая и практическая значимость:

Теоретическая значимость: обоснована возможность и необходимость ментального подхода к обучению теме «Алгоритмизация и программирование»;

Практическая значимость: разработаны ментальные карты как средства реализации ментального подхода к обучению теме «Алгоритмизация и программирование».

Методологический аппарат исследования:

Методологическим аппаратом исследования служат физиологические, кибернетические и психологические подходы к теории мышления, а также методика ментальных карт.

Методологическая основа исследования:

– фундаментальные труды и научные исследования в области когнитивной психологии (У. Найссер, Ж. Пиаже, Дж. Брунер, Дж. Андерсон, Р. Солсо, Ж.Ф. Ришар, Г.А. Саймон, А. Ньюэлл, Ф.Ч. Бартлетт, Ф. Джонсон-Лэйрд, Б.М. Величковский, Л.М. Веккер, М.А. Холодная, К.В. Анохин и др.);

– исследования в области моделирования разума (Н.И.Пак, Т.П.Пушкарева и др.)

– 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. Теоретические основы ментального подхода к обучению теме «Алгоритмизация и программирование» в школьном курсе информатики

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

Для того, чтобы выявить особенности обучения теме "Алгоритмизация и программирование" в школьном курсе информатики, нами был проведен анализ различных учебников по информатике.

В соответствии с общей структурой школьного образования сегодня выстраивается многоуровневая структура предмета «Информатика и ИКТ»: начальный курс (2-4-й классы), базовый курс (5-9-й классы) и профильный курс (10-11-й классы). Курс информатики в данных классах изучается как отдельный предмет, за исключением начального курса, который во многих школах изучается интегративно в рамках других учебных предметов. Возможен вариант, когда «Информатика и ИКТ» не изучается в начальной школе, тогда весь материал ученики получают в основном курсе.

В различных учебниках тема "Алгоритмизация и программирование" представлена достаточно широко, но с некоторыми вариациями в количестве часов, распределении тем по классам и содержательном аспекте. Нами был проведен анализ различных учебников по информатике на предмет того, как в программе различных авторов представлена тема "Алгоритмизация и программирование". Сравнительный анализ был проведен между образовательными программами на 3 ступенях: пропедевтический, базовый, профильный.

В начальной школе для анализа были выбраны учебный комплекс под редакцией Горячева А.В. и Бененсона Е.П.. В курсе Горячева А.В. тема "Алгоритмизация и программирование" появляется во 2 классе при изучении разделов "Алгоритмы", на изучение которого отводится 8 часов и "Шаги и

события", на изучение которого отводится так же 8 часов. При изучении данных разделов учащиеся осваивают такие понятия, как "алгоритм", "порядок шагов", "последовательность", а так же "событие" и как оно влияет на алгоритм; основные принципы ветвления и цикличности, правила записи порядка шагов, правила работы с алгоритмом для двух исполнителей, правила работы для алгоритмов, управляемых событиями, правила алфавитного кодирования. Реализация проходит в учебнике-рабочей тетради (Рис.1)[13].



Рис.1 Примеры заданий из учебника-рабочей тетради

Продолжается изучение данной темы в 3 классе в разделах "Алгоритмы" и "Алгоритмы и исполнители", на которые отводится по 8 часов. В ходе изучения данных разделов учащиеся осваивают значения терминов "линейный алгоритм", "алгоритм с ветвлением", "циклический алгоритм", изучают основные свойства алгоритмов, правила записи на языке блок-схем и учатся их строить, читать, исполнять вычислительные и событийные алгоритмы. Реализация проходит в учебнике-рабочей тетради. (Рис.2)[14].

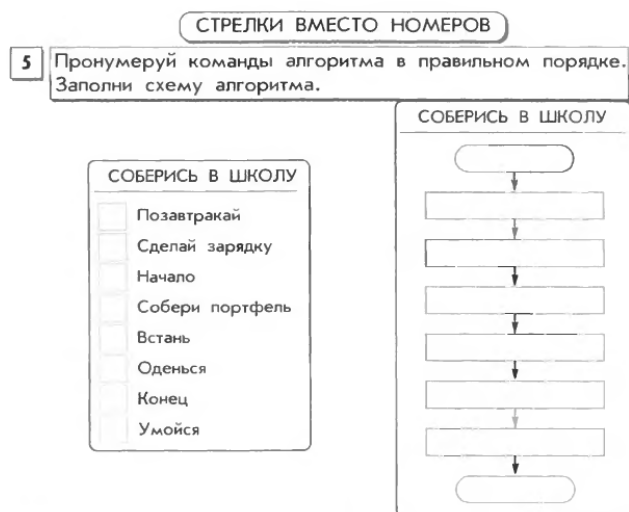


Рис.2 Примеры заданий из учебника-рабочей тетради

Учащиеся применяют полученные знания при составления простых программ в среде Kodu. (Рис 3)



Рис.3. Рабочая среда программы Kodu Game Lab

В 4 классе продолжается изучение темы "Алгоритмы и исполнители", на которую отводится 8 часов. Учащиеся продолжают работу с понятиями "линейный алгоритм", "алгоритм с ветвлением" и "циклический алгоритм". А так же вводится раздел "Программирование и управление", на который отводится 8 часов. Реализация проходит в учебнике-рабочей тетради (Рис.4) [15].



Рис.4 Примеры заданий из учебника-рабочей тетради

Учащиеся применяют полученные знания при составлении простых программ в среде Scratch. (Рис.5)



Рис.5. Рабочая среда программы Kodu Game Lab

По программе курса «Информатика и ИКТ» под редакцией Бененсона Е.П. тема «Алгоритмизация и программирование» изучается со 2 по 4 класс. Во 2 классе на тему «Алгоритмы и исполнители» выделяется 11 часов. В ходе изучения раздела происходит освоение алгоритма как пошагового описания целенаправленной деятельности, формального исполнителя алгоритма, линейных алгоритмов, их создание, способы записи и исполнение. Проходит подготовка к изучению условных алгоритмов: истинности и ложности высказываний, содержащих конструкции «не», если

... то», и слова «все», «некоторые», «не один», «каждый». Реализация проходит в рабочей тетради (Рис.6) [2,7].

86. К заданию 3 учебника.
 Выполни вместе с Энтиком алгоритм. Стрелками обозначь положение Энтика после выполнения каждой команды.

Алгоритм «Найди палку»

Начало
 ИДИ1
 ВПРАВО
 ИДИ1
 ИДИ1
Конец

Рис.6 Пример задания из рабочей тетради

Практическая деятельность осуществляется с помощью учебных исполнителей "Энтик", "Мышка-художник", "Перемещайка", "Аквариум".

Продолжается изучение алгоритмов в 3 классе в разделе «Алгоритмы и исполнители», на который так же, как во втором классе, отводится 11 часов. Учащиеся знакомятся с понятиями имя и значение переменной, присвоением значения переменной, созданием алгоритмов методом последовательной детализации и созданием и исполнением условных алгоритмов для формальных исполнителей. Данные знания сопоставляются с жизненным опытом через планирование деятельности человека с помощью условных алгоритмов типа «ветвление». Реализация проходит в рабочей тетради (Рис.7) [3,4].

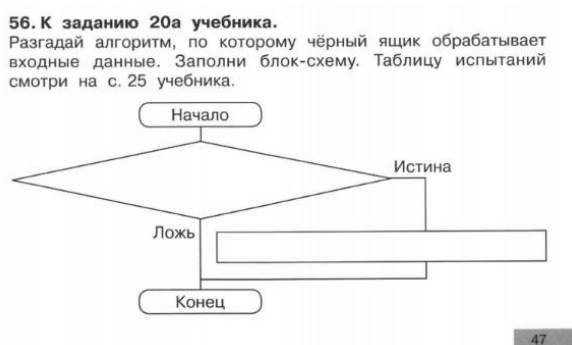


Рис.7 Пример задания из рабочей тетради

Практическая деятельность осуществляется с помощью учебных исполнителей "Считайка", "Переливайка", "Чертёжник".

Завершается изучение темы «Алгоритмизация и программирование» по программе Бененсона Е.П. в начальной школе в 4 классе в разделе «Алгоритм и исполнители», на который отводится 8 часов. В ходе изучения раздела учащиеся знакомятся с понятием «Циклический алгоритм» через циклические процессы в природе и деятельности человека, изучение циклического алгоритма с послеусловием, использование переменных в теле цикла, алгоритмов упорядочивания объектов, создание и исполнение циклических алгоритмов для формальных исполнителей, а так же знакомятся с понятием «Вспомогательный алгоритм». Реализация проходит в рабочей тетради(Рис.8) [5,6].



Рис.8 Пример задания из рабочей тетради

Практическая деятельность осуществляется с помощью учебных исполнителей "Считайка", "Лаборатория", "Путешественник", "Чертежник", "Художник".

Из анализа двух образовательных программ можно сделать вывод, что данные программы равноценны по своему содержанию. В каждой программе идет освоение знаний через применение жизненного опыта, что актуально для учащихся начальной школы.

Для сравнительного анализа средней школы использовались учебные программы авторов Босовой Л.Л., Угриновича Н.Д. и Семакина И.Г.. По программе Босовой Л.Л. данная тема встречается в 6 и 8 классах. В 6 классе на изучение темы "Алгоритмика" отводится 8 часов. В данном разделе учащиеся изучают формальные и неформальные исполнители, осваивают такие учебные исполнители, как Черепаха, Кузнечик, Водолей и др., их

назначение, среды, режим работы, систему команд. Знакомятся с различными формами записи алгоритмов, такими как нумерованный список, таблица, блок-схема. Составляют линейные алгоритмы, алгоритмы с ветвлениями и повторениями для управления учебными исполнителями[8,9].

Продолжение изучения данной темы проходит в 8 классе в разделе "Основы алгоритмизации", на который отводится 10 часов. Здесь происходит знакомство с учебным исполнителем Робот, в основе которого лежит алгоритм работы с величинами. Учащиеся изучают понятие простой величины, ее типы: целые, вещественные, символьные, строковые, логические, а так же строят цепочки команд с использованием данных типов. В следующем разделе "Начала программирования", на который отводится 10 часов, учащиеся начинают знакомиться с языками программирования на основе программной среды Паскаль. Изучают структуру программы, правила представления данных и правила записи основных операторов(ввод, вывод, присвоение, ветвление, цикл). Практическая деятельность заключается в решении задач по разработке и выполнению программ в среде программирования Паскаль[8,10].

По программе Угриновича Н.Д. данная тема встречается только в 9 классе в разделе "Основы алгоритмической культуры", на который отводится 15 часов. В данном разделе учащиеся изучают такие понятия, как "исполнитель", "состояние исполнителя", "система команд", "алгоритм", "свойства алгоритма", составляют линейные алгоритмы управления исполнителями и записывают их на выбранном алгоритмическом языке. Далее учащиеся создают и выполняют программы для решения несложных алгоритмических задач в выбранной среде программирования[24,25].

В программе Семакина И.Г. тема "Алгоритмизация и программирование" встречается в 9 классе в разделах "Управление и алгоритмы", на который отводится 12 часов, и "Введение в программирование", на изучение которого отводится 15 часов. В первом разделе учащиеся изучают понятие алгоритма, его основные свойства,

способы записи алгоритмов, основные алгоритмические структуры и применяют полученные знания при составлении линейных, ветвящихся и циклических алгоритмов управления одним из учебных исполнителей. Во втором разделе учащиеся знакомятся с основными видами и типами величин, назначением языков программирования, правилами представления данных и операторов при оформлении программы на Паскале, а так же работать с готовой программой[21].

Из анализа трех образовательных программ можно сделать вывод, что в курсе Босовой Л.Л. и Семакина И.Г. данная тема представлена более полно, с большим количеством часов. Этапы изучения темы "Алгоритмизация и программирование" выделены более четко, что позволяет увидеть полную картину освоения темы.

В программе среднего общего образования по дисциплине информатики для анализа были выбраны образовательные программы под руководством авторов Семакина И.Г. и Угриновича Н.Д. на базовом уровне. По программе Семакина И.Г. данная тема встречается в 10 классе. Информатика в старшей школе может изучаться в двух вариантах: 1 час в неделю или 2 часа в неделю. Отличие второго варианта заключается в том, что на теоретические и практические занятия отводится в среднем на 1 час больше, чем в первом варианте. Исключение составляют темы "Программирование циклов", на которую отводится на 2 часа больше, и "Работа с массивами", на которую отводится на 3 часа больше. На линию алгоритмизации и программирования в первом варианте отводится 18 часов, а во втором варианте - 35 часов. Она включает в себя: понятия и свойства алгоритма, основы теории алгоритмов, способы описания алгоритмов, языки программирования высокого уровня, решение задач обработки данных средствами программирования. Во втором варианте добавляются две новые темы: организация ввода-вывода с использованием файлов и комбинированный тип данных[19,20].

В программе среднего общего образования у Угриновича Н.Д. на базовом уровне тема "Алгоритмизация и программирование" не изучается. В учебнике 11 класса упоминание о данной теме встречается только в тестах для подготовки к ЕГЭ.

На основе анализа различных программ курса информатики можно сделать следующие выводы:

1) линия алгоритмизации и программирования занимает особое место на протяжении всего курса информатики;

2) с понятием "алгоритм" берёт свое начало во втором классе на основе бытовых ситуаций и развивается в ходе всего курса до использования в качестве основы для решения задач на языках программирования высокого уровня;

3) в школьном курсе информатики на базовом уровне большое внимание уделяется алгоритмизации, а на программирование отведено малое количество времени.

§ 1.2. Сущность ментального подхода к обучению программированию

Опыт изучения программирования и обучения программированию показывает, что основные трудности возникают не при изучении синтаксиса и основных конструкций языка программирования, а на первом этапе решения задач по программированию, на этапе алгоритмизации. И связаны эти трудности с недостаточным уровнем сформированности алгоритмического мышления обучающихся, с их неготовностью воспринимать материал достаточно высокого уровня абстракции и логики.

Следовательно, для разработки методических приемов, позволяющих повысить эффективность обучения и успешность изучения программирования, необходимо обратиться к исследованию процессов мышления.

К изучению мыслительных процессов подходят с разных сторон. Существует психологические, физиологические и кибернетические теории мышления.

В своих исследованиях за основу мы взяли теорию психолога Найссера о том, что мышление человека основывается на ментальных схемах [17]. Наши возможности ориентироваться в пространстве и во времени, осуществлять деятельность, говорит о том, что в нашей памяти формируются и хранятся пространственные, временные и деятельностные ментальные схемы. Попытки формализовать процессы мышления, зафиксировать существующие в мозгу человека ментальные схемы привела к понятию ментальной карты как модели ментальной схемы.

Ментальные карты – один из эффективных инструментов организации знаний, концепций и понятий, родившийся на основе психологии познания Дэвида Аулубела [26]. В 60-е годы теорию развил профессор Корнелльского университета Джозеф Новак [26]. Он разработал правила создания ментальных карт — инструмента визуализации и создания (проработки) новых идей или концепций. В основе концепции ментальных карт лежат представления о принципах работы человеческого мозга: ассоциативное (нелинейное) мышление, визуализация мысленных образов, целостное восприятие (гештальт). Дальнейшее развитие теория получила в работах психолога Тони Бьюзена. В 1974 году он опубликовал книгу «Работай головой», в которой описал метод ментальных карт. Бьюзен превратил метод в коммерческий продукт.

Не менее успешно, чем в коммерции, ментальные карты применяются в педагогике [11]. При традиционном изучении учебного материала, как правило, активизируется лишь незначительная часть огромных возможностей мозга. Традиционно тексты учебников, содержание лекций состоит из фраз, списков и цифр. При его восприятии используются принципы запоминания, рассчитанные на функции левого полушария головного мозга, отвечающего за язык, логику, составление списков и операции с числами, и совсем не

учитываются такие аспекты работы мозга, как воображение, ассоциативность, цвет, ритм и ощущения.

В ментальных картах изучаемый материал представляется в виде ключевого образа, воплощающего в себе главную тему. От этого центрального образа отходят соединительные линии, над которыми написаны или нарисованы ключевые понятия для составления образа. Эти линии в свою очередь соединены с другими, на которых расположены ключевые слова, описывающие образы или сами ключевые образы. Таким способом выстраивается многомерная, ассоциативная и образная «карта памяти» (Mind Map) всего материала. Для большей наглядности и лучшего понимания можно изменять размер и стиль используемого шрифта, использовать цвет и рисунки

Метод используется в учебном процессе в следующих вариантах: объяснение темы, подкрепление понимания и запоминания, проверка знаний и выявление неправильно понятого материала.

Современные физиологические теории мышления позволяют сделать вывод о том, что с научных позиций сознание – это инструмент мозга, они определяют биологические функции сознания, предоставляют возможность изучать сознание объективными методами, выявляют нервные и клеточные основы сознания. Например, в теории когов выявляются биологические, на уровне клеток мозга, ментальные схемы. Согласно положениям этой теории, коги – это системы нейронов, хранящие определенный образ (пространственный, временной или деятельностный) [29].

Кибернетика, занимаясь вопросами создания искусственного интеллекта, пытаясь формализовать и автоматизировать процесс мышления, предлагает три основные модели представления знаний: семантические сети, фреймы, логика [12].

В исследованиях в области искусственного интеллекта отмечается, что более других соответствует современным представлениям об организации долговременной памяти человека именно семантические сети, однако

большинство систем искусственного интеллекта использует фреймы и логику предикатов, поскольку они более формализованы и вследствие этого более эффективно поддаются компьютерной реализации.

В подавляющем большинстве работ, посвященных использованию ментальных карт в учебном процессе [11,31,35,38], ментальные карты представляются в виде простой семантической сети или даже семантического графа, поскольку в них только обозначены понятия как узлы семантической сети и установлены связи между понятиями (нарисованы стрелочки между узлами). Для того, чтобы семантический граф превратился в семантическую сеть, надо указать еще и тип связи (чтобы эти стрелочки были подписаны). А чтобы ментальная карта, представляющая собой модель ментальной схемы, не сводилось к семантической сети, еще необходимо, чтобы в ней был отражены не только понятия и связи между ними, но и некоторые знаки, символы, активизирующие чувственные образы. Тогда она будет работать не только с модельной, понятийной и абстрактной зонами памяти, но и с чувственной зоной памяти [22]. Только в этом случае, как нам кажется, ментальная карта будет действительно ментальной, будет представлять собой модель ментальной схемы и будет обеспечивать более эффективное восприятие информации и более эффективное протекание процесса мышления.

Для примера рассмотрим существующую у каждого человека ментальную схему времен года. В упрощенной версии, сведенная к семантической сети, она выглядит как схема. (Рис.9)

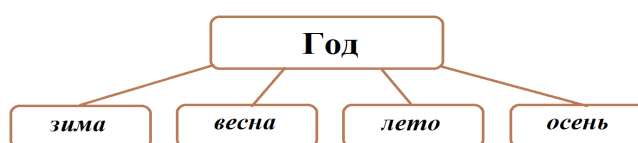


Рис.9. Времена года. Семантическая сеть

Если же добавить снежинки, капель, цветы, желтые листья, обозначить цикличность смены времен года – вот тогда эта семантическая сеть действительно превратится в ментальную карту (Рис.10), более соответствующую той ментальной схеме, которая у нас сформирована.

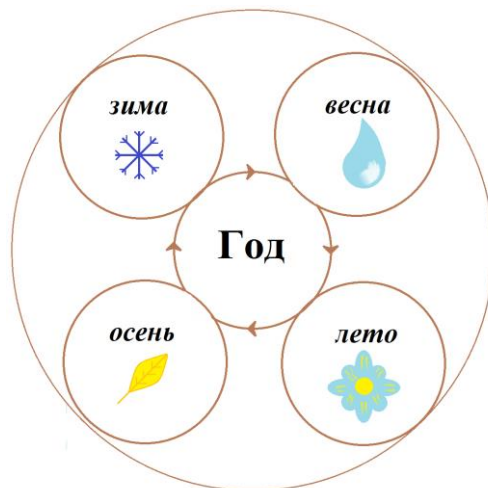


Рис.10. Времена года. Ментальная карта.

В этой связи можно сделать предположение, что к ставшим классическими моделям представления знаний современные психологические и биологические теории мышления позволяют добавить еще одну – ментальные карты как модель ментальной схемы.

Возвращаясь к проблеме использования методики ментальных карт в обучении программированию, можно сделать вывод, что в этом аспекте нас будут интересовать деятельностные ментальные схемы, поскольку ментальной картой таких схем является ни что иное, как алгоритм.

Существует классический способ формализации алгоритма, записанного на естественном языке в виде блок-схемы.

Но блок-схема алгоритма не является ментальной картой в том понимании, в котором мы пытаемся определить это понятие. Блок-схема не задействует чувственную зону памяти, интуицию, она обращается только к понятийной и абстрактной. А большой образовательный потенциал методики ментальных карт заложен именно в этих особенностях и преимуществах этой методики – использование для решения проблемы способностей обоих полушарий, и логического и интуитивного.

Опыт показывает, что при первоначальном изучении основных алгоритмических структур с учениками школы, а зачастую и со студентами младших курсов, недостаточно изобразить блок-схему, соответствующую ей, необходимо по-видимому, сделать ее более ментальной. И это, конечно же, уже будет не блок-схема в привычном понимании. Такую «оживленную» блок-схему, которая должна быть интуитивно понятна человеку, имеющему недостаточно высокий уровень алгоритмического мышления, мы предлагаем назвать ментальной алгоритмической картой.

Предполагаем, что использование подобных ментальных карт в процессе объяснения школьникам основных алгоритмических структур позволит повысить эффективность обучения программированию, сделает изложение учебного материала по этим темам более живым и наглядным, будет способствовать более успешному формированию у них алгоритмического мышления.

Глава 2. Условия реализации ментального подхода к обучению теме «Алгоритмизация и программирование» в школьном курсе информатики

§ 2.1. Процесс формирования и развития алгоритмического стиля мышления у обучающихся

Современное информационное, наукоёмкое общество нуждается в людях с высоким уровнем развития абстрактного мышления вообще и алгоритмического мышления в частности, поскольку осуществление абсолютно любой деятельности – не только профессиональной, где это является необходимым условием в настоящее время практически во всех профессиях, но и бытовой, повседневной, предполагает предварительное моделирование и планирование этой деятельности, накопление и анализ информации, необходимой для её осуществления, т.е. составление алгоритма действий.

Способность человека к составлению таких алгоритмов составляет основу алгоритмического способа мышления.

Алгоритмический способ мышления позволяет принимать оптимальные решения в любой сфере человеческой деятельности и сам по себе никак не связан с программированием и вычислительной техникой. В таком неявном виде он существовал всегда, то есть изначально присущ человеческому мышлению. Появление вычислительной техники и профессии программиста привело лишь к тому, что необходимость алгоритмического способа мышления стала явной по крайней мере для определенного круга специалистов. Для решения программистских задач алгоритмический способ является единственно возможным. В программистской практике общий алгоритмический подход углубляется и детализируется: структура предметной области становится формализованной информационной структурой, в ней вычленяются количественные взаимосвязи, образующие

математическую модель предметной области, алгоритм превращается в компьютерную программу. Т.е. у людей, профессионально занимающихся программированием, алгоритмическое мышление должно быть развито уже не на житейском, повседневном уровне, а на более высоком, профессиональном [16].

Формирование алгоритмического мышления начинается в возрасте 5-6 лет и продолжается формироваться в школе.

Учащиеся могут мыслить по-разному, существуют разные стили мышления. Математика развивает математический стиль мышления, то есть умение рассуждать, пользоваться математическими формулами в своих рассуждениях. Литература и история используют гуманитарный стиль мышления. При изучении школьного курса информатики и других естественно-научных и точных дисциплин, происходит формирование и развитие алгоритмического стиля мышления .

В современной методической литературе, посвященной обучению информатики, сложно найти материалы, в которых не упоминается об алгоритмическом стиле мышления. Но еще сложнее найти публикации, в которых определяется это понятие. В лучшем случае этот термин объясняется на эмпирическом уровне.

В известной статье Дональда Кнута «Алгоритмическое мышление и математическое мышления» проводится сравнительный анализ этих стилей мышления.

В работах М.В.Беляева понятие алгоритмического мышления вводится на основе дихотомии, разделения способов мышления на ассоциативное и алгоритмическое. [28]

Вот такое определение дается Копаевым А.В.: «Алгоритмический стиль мышления – это система мыслительных способов действий, приемов, методов и соответствующих им мыслительных стратегий, которые направлены на решение как теоретических, так и практических задач, и

результатом которых являются алгоритмы как специфические продукты человеческой деятельности». [33]

В работах Пака Н.И. и Степановой Т.А. описывается информационная модель алгоритмического мышления, основанная на информационной модели разума. [18,23]

Согласно этой модели, разум человека представляет собой динамически меняющуюся во времени нейронную сеть, нейроны которой «помнят» свое состояние в каждый момент времени [18].

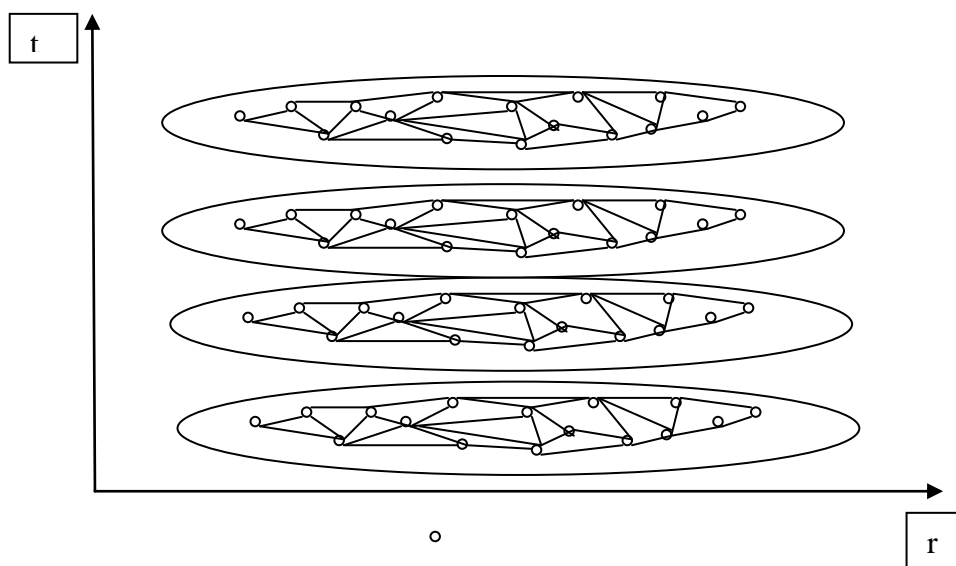


Рис.11. Информационная модель разума

Память в каждый момент времени может быть условно разделена на 4 области:

- чувственную область, в которой запечатляются отражения объектов внешнего мира, воспринятые при помощи органов чувств – зрения, слуха, осязания, обоняния;
- модельную область, в которой хранятся модели, образы, замещающие реальные объекты;
- понятийную область, в которой хранятся понятия, связанные с воспринимаемыми объектами, определения этих объектов, их свойств;
- абстрактную область, в которой хранятся абстракции, обобщенные в значительной степени образы объектов.

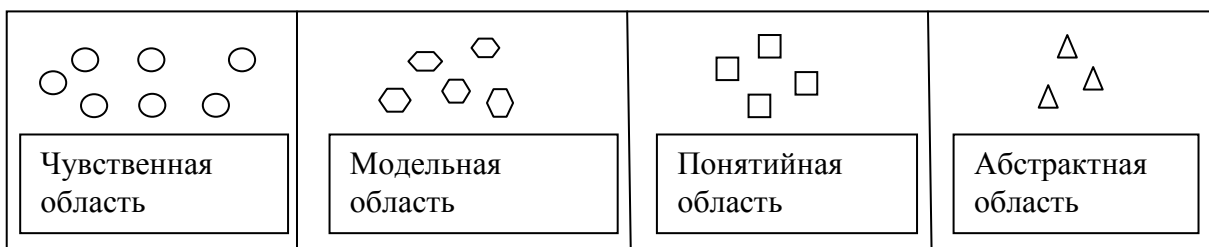


Рис. 12. Информационная модель памяти

Согласно этой модели:

- мысль – это образ в памяти,
- мыслительный процесс – выстраивание цепочки образов,
- мышление – способ выстраивания таких цепочек.

Цепочки могут выстраиваться различным способом – используя образы одной области памяти или нескольких, комбинируя их в различных сочетаниях.

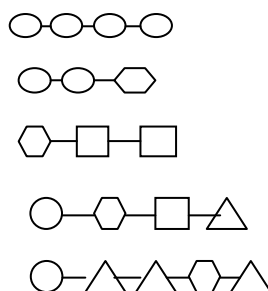


Рис. 13. Различные способы мышления

Это определение не противоречит известным в психологии определениям мышления.

Кроме того, оно очень хорошо иллюстрирует классификацию типов мышления. К примеру, мы будем говорить о конкретном мышлении, если эти цепочки выстраиваются только лишь из образов в чувственной области, образном – если они продляются в модельную, гуманитарном – если построенные цепочки образов состоят в основном из образов понятий, хранящихся в понятийной области, абстрактном мышлении – если они состоят в основном из обобщенных образов, хранящихся в абстрактной области.

Все объекты окружающего мира обладают набором свойств, в общем случае, изменяющихся во времени, и, кроме того, они способны совершать действия, либо над ними, либо с их помощью могут совершаться действия.

Т.е. с любым объектом связано как минимум две функции: $S(t)$ – функция свойств объекта, и $D(t)$ – функция действий объекта (с помощью объекта, над объектом). Если рассматривать модель памяти, представленную на Рис.12, с точки зрения свойств объекта, учитывать только функции. $S(t)$ – это будет мышление созерцателя. Алгоритмическое мышление всегда связано с деятельностью. Если рассматривать модель памяти с точки зрения действий, функции $D(t)$, то тогда получим модель алгоритмического мышления, представленную на Рис.14 [23]

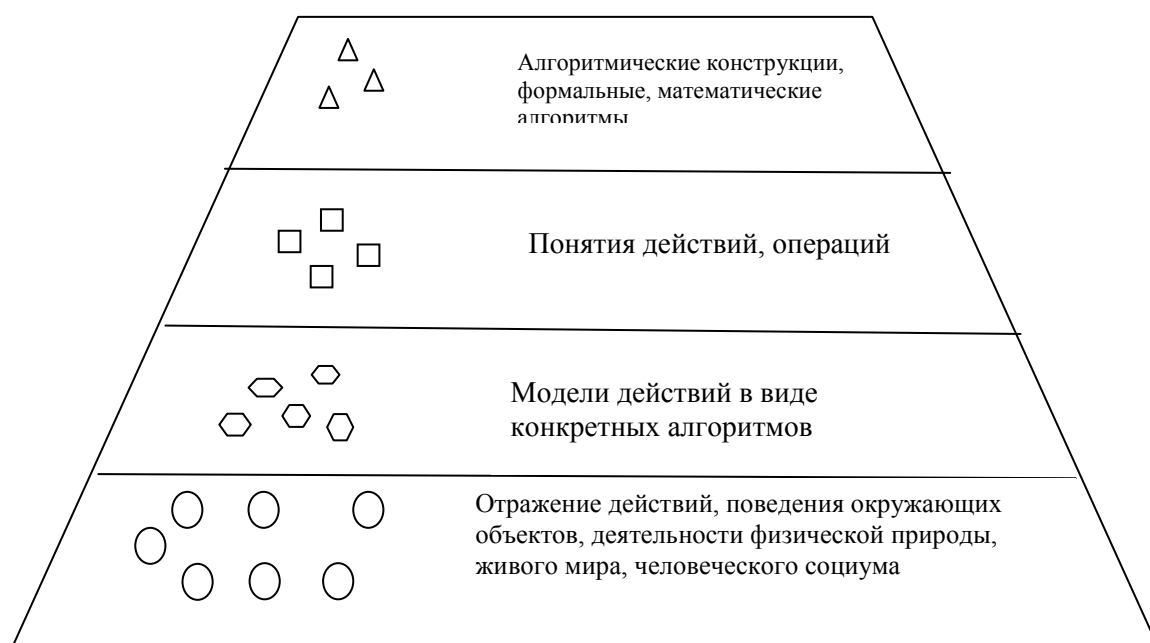


Рис.14. Информационная модель алгоритмического мышления.

В разрезе этой модели, ассоциативное, интуитивное мышление – это выстраивание цепочек только из образов нижнего, чувственного уровня. Чем больше образов задействовано с верхних уровней – тем выше уровень развития алгоритмического мышления.

Процесс формирования мышления, как и процесс формирования памяти, происходит от простого к сложному, от сложного – к еще более сложному, т.е. области памяти заполняются постепенно.

В процессе формирования и развития мышление ребенка проходит все исторические стадии формирования и развития человеческого мышления вообще, так же как физически зародыш проходит все стадии возникновения

и развития живых организмов на земле. До определенного возраста ребенок воспринимает только конкретные чувственные образы, и, следовательно, мыслит только конкретными образами, отражениями действий, поведения объектов окружающего мира. В этом отношении его сознание напоминает сознание первобытного человека. Позднее он научается сопоставлять конкретным образам их простейшие модели, допустим, узнавать маму на фотографии, соотносить изображение собачки на картинке с живой собачкой, тем в его мышлении появляются образы – модели, и модели действий.

Проводя аналогии с развитием человечества, можно рассматривать наскальные изображения как модель удачной охоты. Человечество развивалось, ребенок взрослеет, с конкретными образами и их простейшими моделями связываются понятия, определения, сами модели становятся все более абстрактными, повседневно осуществляющаяся деятельность все более и более сложной. Изучение фундаментальных дисциплин школьной программы способствует все более и более высокому уровню абстрактного мышления. Оперирование понятиями действий, изучение алгоритмов решения математических, физических задач, задач по химии, биологии, выполнение заданий по химии, биологии и другим школьными предметам, осуществление усложняющейся повседневной деятельности, ее планирование формирует более высокий уровень абстрактного мышления в его повседневном понимании.

Далее, изучая в школьном курсе информатики основы алгоритмизации и программирования, у школьников начинает формироваться профессиональное алгоритмическое мышление.

Основными компонентами алгоритмического мышления, согласно исследованиям Афанасовой Л.В. являются:

- *структурный анализ задачи;*
- *разбиение большой задачи на малые;*
- *сведение нерешенной задачи к решенным;*
- *планирование возможных ситуаций и реакций на них;*

- понимание и использование формальных способов записи решения. [36]

Исходя из основных компонентов алгоритмического мышления цепочку мыслительных операции при создании алгоритмов можно схематично представить в следующем виде(Рис.15):



Рис.15. Схема мыслительных операций при составлении алгоритма

Главное в обучении по данной схеме – составление и анализ алгоритмов.

Основные трудности в записи алгоритма возникают по трем причинам.

Во-первых, алгоритм должен быть сразу продуман во всех деталях, ничего нельзя отложить на потом - ведь выполнять алгоритм будет не ученик, а компьютер.

Во-вторых, учащиеся должны не только всё продумать «наперёд» во всех вариантах, но и записать это без выражений типа «и т. д.».

В-третьих, выполнять алгоритм будет компьютер - техническое устройство. Оно не может догадаться, что учащиеся «имели в виду», - все должно быть описано точно и на понятном компьютеру языке.

Школьник может прекрасно представлять себе, какие действия необходимо совершить для получения результата, но не всегда в состоянии записать эту последовательность команд в виде программы.

И вот тот стиль мышления, методы и способы, которые необходимо для перехода от непосредственного управления, от умения сделать к умению записать алгоритм, называется алгоритмическим стилем мышления. [36]

Формирование алгоритмического стиля мышления начинается уже со второго класса. В школьном курсе информатики задания, связанные с алгоритмизацией в начальной школе, направлены на составление линейных планов действий, поиск ошибок в последовательности действий, изучение правил составления блок-схем, знакомство со способами записи алгоритмов, сопоставление действий алгоритма действиям в быту, в сказках, составление линейных алгоритмов, алгоритмов с ветвлением и циклических алгоритмов, выполнение действий по алгоритмам, составление алгоритмов для конкретного исполнителя. Соответственно все они ориентированы на жизненный опыт учащихся[1].

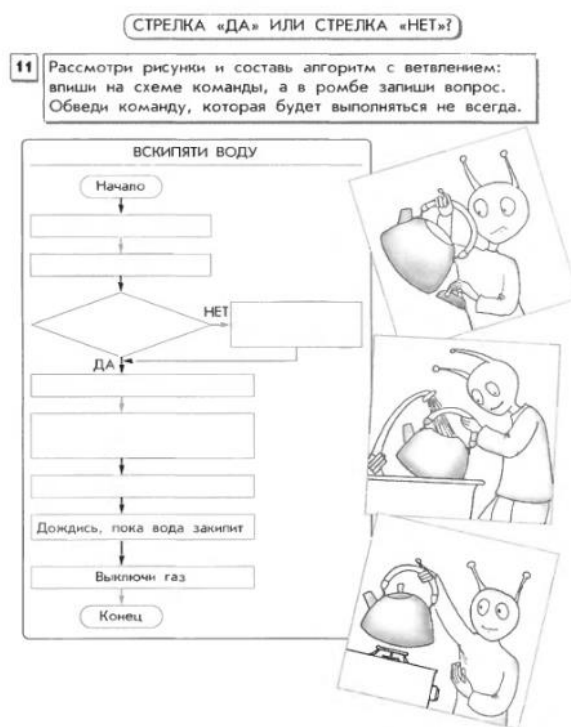


Рис.16. Пример задания для начальной школы по теме "алгоритмизация"

В средней школе продолжается изучение темы "Алгоритмизация", следовательно, продолжается развитие алгоритмического мышления. Рассматриваются свойства алгоритмов и основные алгоритмические структуры в более сложной форме. Новым этапом в развитии алгоритмического мышления является раздел "Начала программирования", в котором учащиеся начинают знакомиться с языками программирования высокого уровня, изучают структуру программы, правила представления данных и правила записи основных операторов (ввод, вывод, присвоение, ветвление, цикл). Практическая деятельность заключается в решении задач по разработке и выполнению программ в выбранной среде программирования[8].

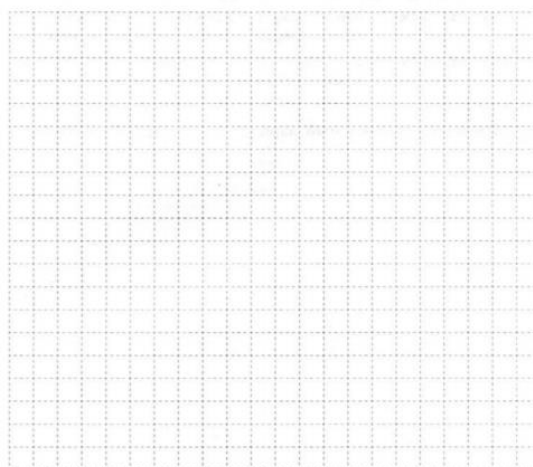
184. Дана программа на языке Паскаль:

```

program a3;
var x, y, z: real;
begin
  writeln ('Введите три числа');
  readln (x, y, z);
  if (x<=y) and (y<=z)
  then
    begin
      x:=2*x;
      y:=2*y;
      z:=2*z
    end
  else
    begin
      x:=abs(x);
      y:=abs(y);
      z:=abs(z)
    end
  writeln (x, ' ', y, ' ', z)
end.

```

Составьте блок-схему, соответствующую программе.



Что является результатом работы программы?

.....

.....

.....

.....

Рис.17. Пример задания для средней школы по теме "программирование"

В старшей школе идет более углубленное изучение программирования. Учебный процесс строится на практической деятельности, которая заключается в решении задач по разработке и выполнению программ с вложенными циклами, вспомогательными алгоритмами и подпрограммами, массивами, символьным, строковым и комбинированным типами данных в выбранной среде программирования[19].

7. Напишите линейную программу перевода любого целого четырехзначного двоичного числа в десятичную систему счисления. Например, дано число в двоичной системе счисления: 1101_2 . Перевод в десятичную систему выполняется так: $1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 = 13$.

Рис.18. Пример задания для старшей школы по теме "программирование"

Поскольку речь идет о развитии алгоритмического стиля мышления, то компьютеры, которые обычно являются целью обучения в информатике, превращаются в средство. Компьютеры нужны для того, чтобы развивать алгоритмический стиль мышления. С ними учить интереснее и эффективнее. Школьники больше успевают усвоить, если у них есть компьютеры с соответствующим программным обеспечением. [36]

Проведенный обзор процесса формирования и развития алгоритмического мышления школьников 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.

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 высоком – если сразу код программы, каким бы несложным и коротким он казался с первого взгляда. Поэтому возникает знакомая для многих преподавателей, ведущих программирование и связанные с ним дисциплины, ситуация когда в полной мере учебный материал воспринимается 20-30% учебной аудитории (класса, группы), они способны после объяснений преподавателя самостоятельно решать новые задачи с использованием изложенного теоретического материала, 40-60% воспринимают и понимают его частично, после

объяснения новой темы они готовы лишь повторить решение демонстрационной задачи, решая подобные задачи, с небольшими вариациями в исходных данных, и оставшиеся 10-40% не способны даже повторить самостоятельно решение демонстрационной задачи, на которой учитель объяснял применение новых алгоритмических структур или операторов языка программирования.

Эта ситуация связана с тем, что наш взгляд, эта оставшаяся часть аудитории, имея недостаточный для восприятия уровень сформированности алгоритмического мышления, нуждается в некоторых промежуточных между постановкой задачи и представлением ее решения в крайне абстрактном, формализованном виде этапах, которые будут постепенно понижать уровень абстракции представления алгоритма ее решения[32].

Нельзя сказать, что преподавателями ничего не делается в данном направлении. Как правило, преподаватели стараются использовать практико-ориентированные задачи, постановка которых была бы понятна учащимся, написание программного кода сопровождается подробными комментариями.

Уже традиционно многие учителя информатики используют при объяснении компьютерные презентации, обеспечивающие различную степень визуализации учебного материала.

Существуют обучающие видеоролики, иллюстрирующие выполнения различных алгоритмов, алгоритмов сортировки, например.

Эффективными средствами изучения алгоритмизации являются программа «Комплект Учебных МИРов» в котором используются встроенные исполнители – «Кузнечик», «Черепашка», «Водолей», «Робот» и другие.

Все эти средства облегчают понимание учебного материала по теме «Алгоритмизация и программирование», они собственно являются теми недостающими звеньями перехода от житейского, повседневного алгоритмического мышления, имеющегося у каждого человека к абстракциям блок-схем и программных кодов.[23]

Но наибольшим потенциалом, с точки зрения ментального подхода обучению программированию, в этом отношении обладает методика ментальных карт.

§ 2.2. Разработка алгоритмической ментальной карты по теме «Алгоритмическая структура ветвление»

Существуют два способа создания ментальных карт: ручной и программный. Для ручного способа достаточно взять: бумажный лист (лучше альбомный) ручки, карандаши, фломастеры (лучше поярче).

Программный способ — это использование компьютерных программ. У каждого способа есть свои достоинства и недостатки. Используя компьютерную программу, проще редактировать построенную ментальную карту, для того, чтобы что-то в ней изменить не придется перерисовывать ее полностью. Также гораздо удобнее носить ментальную карту на электронном носителе, чем на альбомном листе. Минусом работы в программе является ее шаблонность, ограничения в рисовании и визуальном выражении своих мыслей.

Для создания ментальных карт подходят такие программы, как XMind, Adobe Flash Professional (для «оживленных» ментальных карт), Prezi и т.д. Кратко охарактеризуем эти программы.

XMind - это программа с простым и понятным интерфейсом. В свое время она явилась революцией в создании и редактировании ментальных схем. К достоинствам программы следует отнести тот факт, что она является свободно распространяемым программным продуктом.

Работая с программой, пользователи имеют возможность легко изобразить совершенно разные идеи, создавая их структуру в удобном менеджере ментальных схем.

Программа XMind была разработана в качестве ведущего инструмента визуального создания ментальных схем, с простым и дружелюбным

пользовательским интерфейсом[27] (Рис.19). К недостаткам этой программы стоит отнести ограниченность ее возможностей.



Рис.19. Программа XMind

Adobe Flash Professional - программа для создания профессиональных анимационных Flash-файлов. Она обладает расширенным набором инструментов для работы с Flash технологией. Программа интегрируется с другими графическими приложениями компании Adobe. Тут присутствует возможность импорта файлов из Adobe Photoshop и Adobe Illustrator, удобная временная шкала, позволяющая наиболее точно накладывать эффекты движения и расширенные средства для создания и редактирования основных геометрических форм, с возможностью генерации своих фигур с помощью встроенного модуля JavaScript API. Кроме того, данная программа позволяет легко добавлять в Flash-файл музыку в формате MP3 и включать в него видео для получения файлов FLV формата[37] (Рис.20).

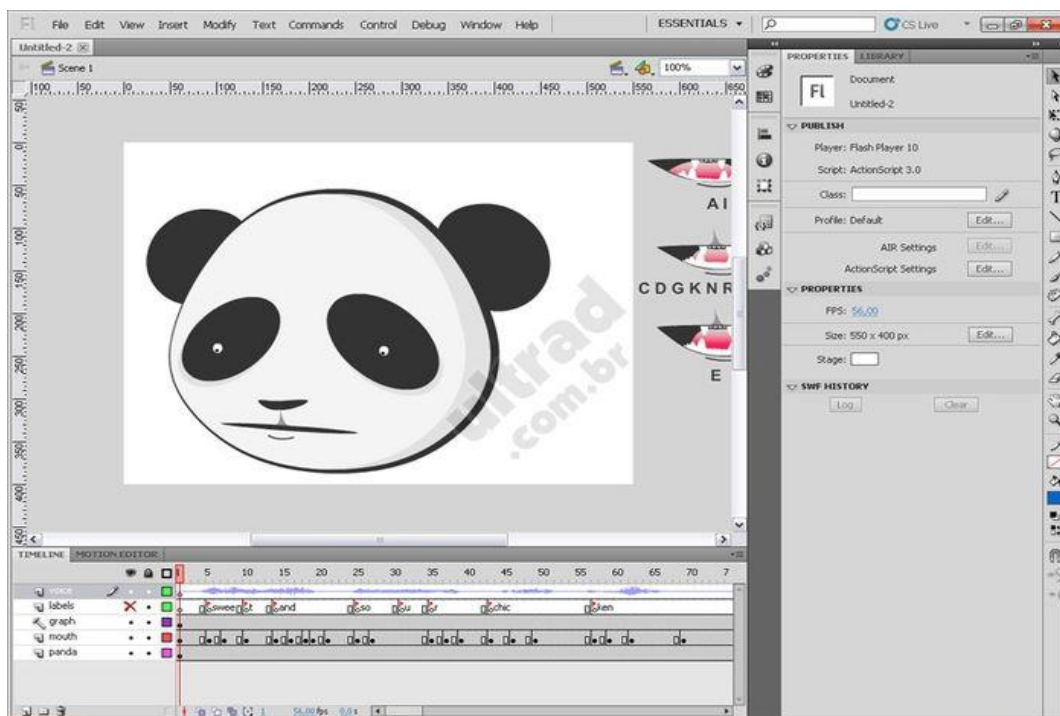


Рис.20. Программа Adobe Flash Professional

В своей работе для построения ментальной схемы мы пользовались облачной технологией Prezi. Prezi представляет собой весьма удобный сервис. Возможности программы довольно широки и с помощью этого инструмента можно создать очень эффектный и красивый проект.

Использование zooming-эффекта позволяет расположить слайды на рабочем столе и переключаться между ними, что очень удобно. Кроме того, в работе над проектом могут участвовать сразу несколько человек (режим реального времени). В презентацию можно включать видеоролики, анимированные и векторные изображения, аудиозаписи, текстовые блоки. Инструмент предлагает широкий выбор шаблонов и тем. Доступен практически на всех платформах. К минусам программы можно отнести интерфейс на английском языке и сложность для новичков[34] (Рис.21).

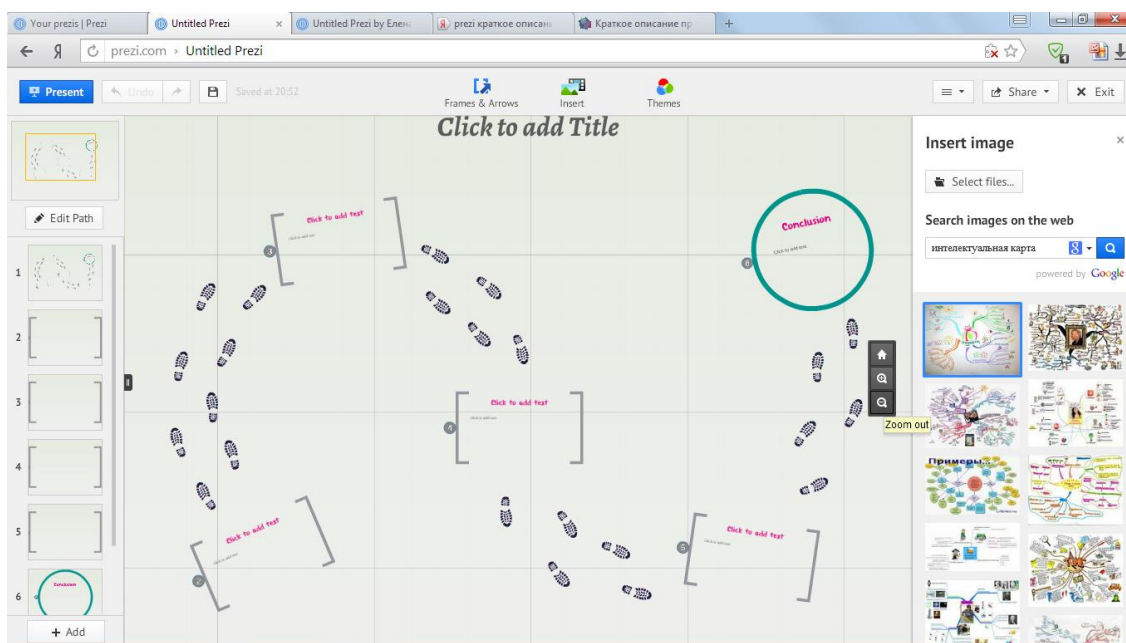


Рис.21. Программа Prezi

Технологический процесс создания ментальной карты с помощью Prezi строится как презентация. Для построения понятной карты нам необходимо понять алгоритм построения задачи. Для этого необходимо простроить все связи между составными элементами, записать их в алгоритмическую цепочку. Затем данную задачу следует прописать в программе, исходя из языка программирования (в нашем случае это Free Pascal). Только после этих действий мы сможем построить ментальную карту.

При изучении темы «Ветвление» и условных операторов возникают проблемы, связанные с неумением правильно составить логические выражение, являющееся условием выбора той или иной ветви программы, с непониманием того факта, что в зависимости от истинности этого условия будет выполняться ИЛИ одна ИЛИ другая ветвь программы, но никогда не обе, что итог выполнения программы будет различным.

Ментальная карта имеет 5 уровней абстракции. Рассмотрим подробно каждый из них.

Первый житейский уровень – это «оживленная» ситуация про сказочного богатыря – картинка, на который богатырь стоит перед камнем с указателями. Ему нужно сделать выбор, от которого зависит дальнейший

исход: идти налево или направо. От выбора направления будет зависеть итог: либо богатырь будет богат, либо он будет женат. Оба этих исхода представлены на последующих кадрах, причем каждый исход завершается финальным кадром – конец сказки [Приложение А, Рис.1].

Затем переходим от житейского представления задачи к более абстрактному. На втором уровне на ту же картинку наложены элементы блок-схемы. Богатырь является переменной 'А', которой присвоится значение – налево 'L' или направо 'R'. В центре камня появляется блок условия 'A=L', которое если переменная 'А' будет иметь значение 'L' будет истинным, если же переменная 'А' будет иметь значение 'R' – ложным. В зависимости от того, какое значение примет переменная 'А', осуществляется выбор дальнейших действий [Приложение А, Рис.2].

На третьем этапе мы переписали элементы блок-схемы в виде программных кодов. Это действие поможет ученику понять, что обозначает и что делает каждое кодовое слово в программе и это впоследствии поможет ему самостоятельно писать программы [Приложение А, Рис.3].

На четвертом шаге картинка, которая была фоном программы, уходит, и у нас остается обычная блок-схема, которая соответствует поставленной задаче. На данном этапе для ученика очень важно понимание алгоритма [Приложение А, Рис.4].

На заключительном пятом этапе при помощи опорной блок-схемы несложно написать саму программу [Приложение А, Рис.5].

К плюсам такой ментальной схемы относится то, что, если в определенный момент ученику непонятен какой-либо этап, то он всегда может к нему вернуться и разобрать то, что ему не понятно.

Таким образом, мы получили ментальную схему, благодаря которой ученик сможет разобраться в теме "ветвление"; понять структуру программы, включающую в себя условный оператор; научиться применять полученные знания на практике.

§ 2.3. Диагностика уровня сформированности алгоритмического стиля мышления

Обязательным компонентом процесса обучения, его завершающим этапом, является контроль (проверка результатов обучения).

Функция контроля состоит во всесторонней проверке результатов обучения, таких как:

- когнитивная сфера - овладение знаниями и способами их применения;
- психологическая сфера - развитие личности;
- социальная сфера - социальная адаптация.

Именно поэтому многими современными педагогами принято положение о том, что цели обучения обязательно должны быть диагностичны.

Таким образом, диагностика знаний по информатике выполняет двуединую задачу:

1. позволяет осуществить сопоставление наличных знаний с требованиями государственного стандарта;
2. способствует личностному росту субъекта учебной деятельности.

Важно также сознавать, что именно контролируется – результат или способ действия. Проще контролируется результат, но за правильным результатом может скрываться ошибка в способе действия. Контроль последнего, конечно, важнее.

К основным методам диагностики результатов обучения относятся:

- *Наблюдение*, которое заключается в непосредственном восприятии и прямой регистрации фактов. Наблюдение за учащимися в ходе учебного процесса помогает учителю составить правильное представление об отношении ученика к занятиям, степени его познавательной активности, самостоятельности, доступности и посильности для него учебного материала и на этой основе планировать форму и метод проверки знаний.

- *Методы устной диагностики:* рассказ, объяснение, сообщение ученика, чтение им текста, схемы, карты, беседа и пр. Данные методы используются с целью выявления уровня знаний, умений и навыков отдельных учащихся, уровня речевого развития, навыков общения, умения выступать и дискутировать.

- *Методы письменной диагностики:* контрольные работы, диктанты, изложения, сочинения, письменные ответы на вопросы. Цели их применения: выявление уровня овладения теоретическими знаниями; умений их применять для решения конкретных задач; степени владения письменной речью.

- *Тестирование* - опросник с множественным выбором ответов и примитивным подсчетом очков. Между тем дидактический тест это набор стандартизированных заданий по определенному материалу, устанавливающий степень усвоения его учащимися.

- *Практические методы диагностики:* лабораторные, измерительные работы, создание изделий и пр. применяются с целью выявления уровня сформированности тех или иных умений и навыков практической работы, или двигательных навыков, но, к сожалению, доля применения этих методов в школе невелика[30].

Как видно из перечисленных методов диагностики основная их направленность заключается в проверке предметных знаний, то есть, что ребенок знает по предмету. В нашем исследовании нам необходимо диагностировать уровень сформированности алгоритмического стиля мышления. Не один из перечисленных методов не позволяет в полной мере произвести данную диагностику. Следовательно, здесь необходимы другие методы.

В рамках ментального подхода мы предлагаем использовать для диагностики уровня сформированности алгоритмического мышления ментальные карты. Как было сказано ранее, ментальные карты используются не только для того, чтобы объяснять новый материал, но и эффективно

используются для диагностики. Ментальная карта - это отражение ментальной схемы. Когда ребенок рисует ментальную карту, он отражает свои мыслительные процессы с её помощью.

Чтобы внедрить разработанную ментальную карту в учебный процесс, были разработаны конспекты уроков по теме "Алгоритмическая структура ветвление" в трех вариациях: для урока введения новых знаний по данной теме с помощью ментальной карты, для закрепления пройденного материала, а так же для диагностики уровня сформированности алгоритмического стиля мышления с помощью методики ментальных карт.

Ментальная карта на уроке введения новых знаний применяется на этапе объяснения нового материала. Особое внимание нужно уделить второму и третьему этапу ментальной карты, когда на картинке появляются элементы блок-схемы и элементы программного кода. [Приложение А, Рис.2, Рис. 3] Это можно осуществить посредством следующей системы вопросов:

- "... - Чем в схеме является богатырь?*
- Что представляет собой камень?*
- Какое условие представлено на карте?*
- Как вы думаете, что обозначает L?*
- Могло бы быть другое условие?*
- Для представленного на картинке условия, какое из возможных исходов верное?*

- Следовательно, что должно быть написано на левой стрелке?*

(Истина)

- А тогда на правой?*
- Чем отличается эта картинка от предыдущей?*
- Какие значения может принимать переменная a?*
- А какой тип данных предусматривает введение символов?*
- С какого служебного слова начинается конструкция ветвления?*
- Какое служебное слово прописывается после условия?*
- Что оно обозначает?*

- А если условие ложно, то какое служебное слово напишем? ..."

[Приложение Б]

На основе приведённого разбора задач на первых трех этапах учитель демонстрирует блок-схему, [Приложение А, Рис. 4] по которой предлагает учащимся составить программу решения данной задачи.

Полный конспект урока введения новых знаний приведён в Приложении Б.

Ментальная карта на уроке закрепления знания применяется на этапе закрепления изученного материала. На её основе учитель задаёт систему вопросов для повторения основных понятий.

"... - Если условие истинно, то выводится результат и программа завершена.

- Если условие ложно, то выводится соответствующий результат и программа завершается. ..." [Приложение В]

При диагностике знаний и уровня сформированности алгоритмического мышления мы решили отойти от стандартной диагностики знаний и не задаем детям вопросов "Что такое ветвление?", а предлагаем учащимся нарисовать свою ментальную карту, где в жизненной ситуации встречается структура ветвление, используя в качестве образца для выполнения учащимися практической работы разработанную нами ментальную карту.

Полный конспект урока закрепления и проверки знаний приведён в Приложении В.

Вся сложность в проверке знаний заключается в том, что, например, тест легко проверить на правильность - неправильность ответа, а в данном случае проверка знаний заключается только в экспертной оценке, анализе учителя в области ментальных карт и в области программирования.

Для экспертной оценки ментальной карты, необходимы определенные критерии, опираясь на которые, учитель сможет оценить уровень развития

алгоритмического стиля мышления. Чтобы оценить ментальные карты, можно предложить следующие критерии:

- соответствие выбранной задачи данной теме;
- проверка того, является ли данная последовательность алгоритмом;
- соблюдение структуры ментальной карты;
- соответствие элементов блок-схемы объектам данной задачи;
- правильность замены элементов блок-схемы программным кодом;
- проверка составленной программы.

Рассмотрим каждый критерий более подробно.

Соответствие выбранной задачи данной теме: изначально, очень важно выбрать задачу, соответствующую данной теме, так как задача является основой для разработки ментальной карты. Например, для темы "Алгоритмическая структура "Ветвление"" будет ошибочным выбрать задачу про обед в столовой, в которой ученику необходимо сделать выбор между первым, вторым и третьим блюдом. Так как он может выбрать и первое, и второе, и третье блюда, а исход должен быть только один, то данная задача не подходит заданной теме.

Проверка того, является ли данная последовательность алгоритмом: так как программирование строится на базе алгоритмизации, то последовательность решения задачи должна быть алгоритмом. Для этого, а так же для того, чтобы быть понятной программной среде, должны выполняться все свойства алгоритма, а именно:

- Дискретность - алгоритм должен состоять из конкретных действий, следующих в определенном порядке;
- Детерминированность - любое действие должно быть строго и недвусмысленно определено в каждом случае;

- Конечность - каждое действие и алгоритм в целом должны иметь возможность завершения;
- Массовость - один и тот же алгоритм можно использовать с разными исходными данными;
- Результативность - отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях.

Соблюдение структуры ментальной карты: разработанная ментальная карта должна иметь пять уровней абстракции, соответствующие ментальной карте, приведенной в качестве примера. Содержание каждого уровня должно быть соответствующим.

Соответствие элементов блок-схемы объектам данной задачи: каждый элемент блок-схемы должен быть размещен на соответствующем ему объекте. В нашем примере ментальной карты условием является надпись на камне, следовательно, блок условие изображается на нем и т.д.

Правильность замены элементов блок-схемы программным кодом: проверка умений перевода элементов блок-схемы в элементы программного кода.

Проверка составленной программы: блок-схема, составленная в ходе решения задачи, должна быть переведена на соответствующий ей программный код. От правильности составления блок-схемы будет зависеть правильность составления программы, а, соответственно, ее выполнение.

Использование ментальных карт в процессе обучения программированию делает изложение учебного материала более живым и наглядным, способствует более успешному формированию у них алгоритмического мышления.

§ 2.4. Результаты апробации

Во время прохождения педагогической практики на базе МБОУ СШ №145 были апробированы разработанные материалы, а именно, конспект урока на закрепление знаний по теме "Ветвление" [Приложение В] и разработанная ментальная карта по теме "Ветвление" [Приложение А]. Так же была проведена диагностика с использованием методики ментальных карт, путем экспертной оценки учителем ментальных карт, разработанных учениками в ходе учебного занятия.

Урок информатики по теме «Ветвление» был проведен в 9 классе. На уроке для демонстрации ментальной карты использовались технические средства, такие как проектор, интерактивная доска и компьютер. Данные средства помогали поддерживать внимание на уроке, сделали процесс обучения более "оживленным". В ходе урока у учащихся активизировались мыслительные словесно-логические процессы, образная память и воображение, которые необходимы для составления ментальных карт. Задания на уроке стимулировали творческую деятельность учащихся. В конце урока была проведена рефлексия по самоанализу учебных действий.

На этапе повторения знаний часто устанавливалась обратная связь с учащимися. Цель этого действия - активизировать учащихся с начала урока на учебную деятельность, а так же повторить знания с прошлых занятий. Уровень дисциплины был достаточно высокий. Учащиеся с интересом принимали информацию и участвовали в обсуждении.

Продуктом деятельности учащихся в ходе урока стали разработанные ими ментальные карты, которые отражают уровень усвоения знаний и сформированности алгоритмического мышления. Созданные ими карты прикреплены в приложение Г

Мы не можем сделать точный вывод о повышении алгоритмического стиля мышления учеников, так как за один урок это невозможно. Необходима система уроков с применением ментальных карт. Но мы можем судить о развитии алгоритмического мышления по правильности

разработанной школьниками ментальной карты. Для этого была проведена экспертная оценка по каждому из критериев, рассмотренных ранее.

Из учащихся 9 класса было организовано три группы по 3-4 человека, следовательно, было разработано три ментальные карты.

Экспертная оценка первой ментальной карты [Приложение Г, Рис.1, Рис.2]:

Задача первой группы заключалась в том, что человеку необходимо сделать выбор в зависимости от погодных условий: брать с собой зонт при походе в магазин или не брать. Данная задача соответствует данной теме, поэтому удовлетворяет первому критерию.

Последовательность действий в решении, как и в предыдущей задаче, удовлетворяет свойствам алгоритма: дискретность, детерминированность, конечность, массовость, результативность. Следовательно, второй критерий выполняется.

Структура ментальной карты соблюдена. Переход осуществляется от житейского уровня к блок-схемам, от блок-схем - к программному коду.

Элементы блок-схемы расположены на соответствующих объектах, поэтому четвертый критерий выполняется.

Перевод элементов блок-схемы в элементы программного кода учащиеся первой группы выполнили успешно, поэтому данный критерий считается выполненным.

По данной задаче учащиеся составили программу, которая соответствует составленной блок-схеме. При реализации данного кода в программной среде Паскаль, был выдан результат, соответствующий решению данной задачи.

Так как все критерии для экспертной оценки первая группа выполнила, то результат деятельности учащихся оценивается 5 баллов.

Экспертная оценка второй ментальной карты [Приложение Г, Рис.3, Рис.4]:

Задача второй группы заключалась в том, что мальчику необходимо сделать выбор: купить билет на поезд в Москву к бабушке или в Хабаровск к тёте. Данная задача соответствует теме "Ветвление", поэтому удовлетворяет первому критерию.

Последовательность действий в решении данной задачи удовлетворяет свойствам алгоритма, следовательно, второй критерий выполняется.

Структура ментальной карты соблюдена. Переход осуществляется от житейского уровня к блок-схемам, от блок-схем - к программному коду.

Элементы блок-схемы расположены на соответствующих объектах: блок условия располагается на окне кассы, исходы выбора расположены на соответствующих поездах, следовательно, четвертый критерий выполняется.

Перевод элементов блок-схемы в элементы программного кода учащиеся второй группы выполнили успешно.

По данной задаче учащиеся составили программу, которая соответствует составленной блок-схеме. При реализации данного кода в программной среде Паскаль, был выдан результат, соответствующий решению данной задачи.

Так как все критерии для экспертной оценки вторая группа выполнила, то результат деятельности учащихся оценивается 5 баллов.

Экспертная оценка третьей ментальной карты [Приложение Г, Рис.5, Рис.6]:

Задача третьей группы заключалась в том, ученику необходимо определить свои действия в зависимости от звонка: идти на урок или гулять на перемене. Данная задача соответствует теме урока, поэтому удовлетворяет первому критерию.

Последовательность действий в решении данной задачи удовлетворяет свойствам алгоритма: состоит из конкретных действий, следующих в определенном порядке, каждое действие строго определенное, имеет конечное число шагов, данный алгоритм может быть применен к другим

подобным задачам, приводит к правильному результату. Следовательно, второй критерий выполняется.

Структура ментальной карты соблюдена полностью, что удовлетворяет третьему критерию.

На четвертом этапе не все элементы блок-схемы были расположены на соответствующих объектах: блок условия учащиеся расположили на школьнике. При совместном анализе данной ситуации был сделан вывод, что данный элемент должен был располагаться на звонке, так как именно он отвечает за условие, а ученик является в этом случае переменной, которая делает выбор. Поэтому данный критерий выполнен частично.

Перевод элементов блок-схемы в элементы программного кода учащиеся третьей группы выполнили успешно.

По данной задаче учащиеся составили программу, которая соответствует составленной блок-схеме. При реализации данного кода в программной среде Паскаль, был выдан результат, соответствующий решению данной задачи.

Ошибка по четвертому критерию не повлияла на дальнейшее решение задачи, но позволила акцентировать внимание учащихся на то, что является в задаче условием, а что переменной. Вследствие этого данная ошибка является недочетом, поэтому результат деятельности учащихся данной группы оценивается в 5 баллов.

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

Заключение

В ходе проведенной работы цель была достигнута: из анализа учебников по информатике различных авторов, нами были выявлены особенности обучения теме «Алгоритмизация и программирование» в ШКИ.

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

Для того, чтобы уточнить понятие алгоритмического стиля мышления на основе компонентов алгоритмического мышления, нами была изучена его информационная модель, с помощью которой можно проследить цепочку мыслительных операций при создании алгоритмов.

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. Были подобраны критерии для диагностики развития алгоритмического стиля мышления.

Разработана ментальная карта по теме «Ветвление» и методика её применения для урока введения новых знаний, закрепления пройденного материала, а так же для диагностики уровня сформированности алгоритмического стиля мышления.

В период прохождения педагогической практики была проведена апробация разработанных материалов и проведен анализ ее результатов.

В результате проведенного исследования можно сделать следующие выводы:

- применение ментальных карт в обучении является возможным и нужным, так как они учитывают такие аспекты работы мозга, как воображение, ассоциативность, цвет, и ощущения, то есть чувственную зону памяти.
- использование ментальных карт в процессе объяснения обучающимся основных алгоритмических структур позволяет повысить эффективность обучения программированию, делает изложение учебного материала по этим темам более живым и наглядным,

способствует более успешному формированию у них алгоритмического мышления.

- Не менее успешно методика ментальных карт может применяться для диагностики результатов обучения, поскольку в ментальных картах отражается не только факт того, усвоены или нет знания по предмету, но и насколько глубоко они понимаются обучающимися, ментальная карта отражает процесс решения задачи, а не только результат, следовательно, позволяет оценить процесс мышления.

Список литературы

1. Аверкин Ю. А. Информатика. 2–4 классы: методическое пособие / Ю. А. Аверкин, Д. И. Павлов. — М.: БИНОМ. Лаборатория знаний, 2016. — 40 с.
2. Бененсон Е.П. Информатика и ИКТ [Текст]: 2 кл.: Учебник: В 2 ч. (Первый год обучения)/ Е.П.Бененсон, А.Г. Паутова - 3-е изд. - М.: Академкнига/Учебник, 2013. - Ч.2: 80 с.: ил.
3. Бененсон Е.П. Информатика и ИКТ [Текст]: 3 кл.: Методическое пособие (Второй год обучения)/ Е.П.Бененсон, А.Г. Паутова - М.: Академкнига/Учебник, 2012. - 248 с.
4. Бененсон Е.П. Информатика и ИКТ [Текст]: 3 класс: Учебник: В 2 ч. (Второй год обучения)/ Е.П.Бененсон, А.Г. Паутова - 3-е изд. - М.: Академкнига/Учебник, 2013. - Ч.2: 96 с.: ил.
5. Бененсон Е.П. Информатика и ИКТ [Текст]: 4 класс: Методическое пособие (Третий год обучения)/ Е.П.Бененсон, А.Г. Паутова - М.: Академкнига/Учебник, 2012. - 272 с.
6. Бененсон Е.П. Информатика и ИКТ [Текст]: 4 класс: Учебник: В 2 ч. (Третий год обучения)/ Е.П.Бененсон, А.Г. Паутова - 3-е изд. - М.: Академкнига/Учебник, 2013. - Ч.1: 96 с.: ил.
7. Бененсон Е.П., Паутова А.Г. Информатика и ИКТ [Текст]: 2 кл.: Методическое пособие/ Е.П.Бененсон, А.Г. Паутова - М.: Академкнига/Учебник, 2012. - 120 с.
8. Босова Л.Л. Информатика. Программа для основной школы: 5-6 классы. 7-9 классы / Л.Л. Босова, А.Ю. Босова. - 3 изд. - М.: БИНОМ. Лаборатория знаний, 2015. - 88 с.: ил. - (Программы и планирование).
9. Босова Л.Л. Информатика: учебник для 6 класса/ Л.Л. Босова, А.Ю. Босова. - М.: БИНОМ. Лаборатория знаний, 2013. - 213с.: ил.
10. Босова Л.Л. Информатика: учебник для 8 класса/ Л.Л. Босова, А.Ю. Босова. - 2-е изд., испр. - М.: БИНОМ. Лаборатория знаний, 2014. - 160 с.: ил.

11. Гаврилова Т.А., Хорошевский В.Ф. «Базы знаний интеллектуальных систем». — Питер, 2000
12. Голубцова А.В., Грук Е.Д., Степанова Т.А. Разработка ментальных алгоритмических карт по теме «Основные алгоритмические структуры». — Материалы II Всероссийской научно-практической конференции с международным участием «Перспективы и вызовы информационного общества», 2013
13. Горячев А.В., Горина К.И., Волкова Т.О. Информатика. 2 класс ("Информатика в играх и задачах"). Учебник в 2-х частях, часть 1. - Изд. 3-е, испр. - М.:Баласс; Школьный дом, 2011. - 64 с. : ил. (Образовательная система "Школа 2100").
14. Горячев А.В., Горина К.И., Суворова Н.И. Информатика в играх и задачах. 3-й класс: Учебник в 2-х частях, часть 1. - Изд. 2-е, испр. - М.:Баласс, 2009. - 64 с. : ил. (Образовательная система "Школа 2100").
15. Горячев А.В., Горина К.И., Суворова Н.И., Лобачёва Л.Л., Спиридонова Т.Ю. Информатика. 4 класс ("Информатика в играх и задачах"). Учебник в 2-х частях, часть 1. - Изд. 3-е, испр. - М.:Баласс; Школьный дом, 2011. - 64 с. : ил. (Образовательная система "Школа 2100").
16. Молодежь и наука: XV Международный форум студентов, аспирантов и молодых ученых: материалы научно-практической конференции. Красноярск, 19– 26 мая 2014 г. /ред. кол.; отв. ред. С.В. Бортновский; Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2014. – 514 с.
17. Найссер У. Познание и реальность. — М.: Прогресс, 1981. — 252 с.
18. Пак Н.И. Информационное моделирование. Учебное пособие. – Красноярск: РИО КГПУ, 2010. –152 с
19. Семакин И.Г. Информатика. Программа для старшей школы: 10-11 классы. Базовый уровень / И.Г. Семакин. - М.: БИНОМ. Лаборатория знаний, 2015. - 100 с. - (Программы и планирование).

20. Семакин И.Г. Информатика. Базовый уровень: учебник для 10 класса / И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шеина. - 4-е изд. - М.: БИНОМ. Лаборатория знаний, 2015. - 264 с.: ил.
21. Семакин И.Г. Информатика: учебник для 9 класса/ И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова. - 5-е изд. - М.: БИНОМ. Лаборатория знаний, 2012. - 341 с.: ил.
22. Степанова Т.А. Сущность алгоритмического мышления с позиций информационного подхода. // Инновации в непрерывном образовании, 2012, №3. С. 95-100
23. Степанова Т.А. Теория алгоритмического мышления: учебное пособие для магистрантов, учителей общеобразовательных учреждений, преподавателей вузов; Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2014. – 72 с.: илл.
24. Угринович Н.Д. Информатика и ИКТ: учебник для 9 класса/ Н.Д. Угринович. - 6-е изд. - М.: БИНОМ. Лаборатория знаний, 2012. - 295 с.: ил.
25. Угринович Н.Д. Информатика. Программа для основной школы: 7-9 классы / Н.Д. Угринович, Н.Н. Самылкина. - М.: БИНОМ. Лаборатория знаний, 2012. - 53с. - (Программы и планирование).
26. Joseph D. Novak, Cornell University. The Theory Underlying Concept Maps and How To Construct Them. URL: <http://cmap.coginst.uwf.edu/info/index.html> (дата обращения: 1.04.2015)
URL: <http://www.studfiles.ru/preview/4518781/> (дата обращения: 2.06.2017)
27. XMind - Конструирование интеллектуальных карт. URL: <http://pro-spo.ru/-linux-e/1940-2010-02-17-10-45-32> (дата обращения: 1.04.2015)
28. Беляев, М.В. Алгоритмическое мышление как цель современного образования. URL: www.mbelyaev.chat.ru (дата обращения: 1.04.2015)
29. Богданова Г.Н. Интеллект-карты как средство развития интеллектуальных способностей учащихся. URL: <http://do.gendocs.ru/docs> (дата обращения: 1.04.2015)

30. Диагностика знаний по информатике: Файловый архив студентов StudFiles.ru: Интернет-сервис. 2015.
31. Жаркая М.А. Изучаем тему «Понятие алгоритма» посредством создания карт понятий. URL: Персональный сайт учителя информатики. – <http://inf429.ucoz.ru> (дата обращения: 1.04.2015)
32. Кнут, Д. Алгоритмическое мышление и математическое мышление. URL: www.philosophy.ru (дата обращения: 1.04.2015)
33. Копаев А.В. Алгоритм как модель алгоритмического процесса. URL: <http://www.rusedu.info/Article100.html> (дата обращения: 1.04.2015)
34. Краткое описание программ для создания презентаций. URL: <http://prezentacii.me/programmy-i-onlajn-servisy-dlya-sozdaniya-prezentacij-kratkoe-opisanie-ix-vozmozhnostej.html> (дата обращения: 1.04.2015)
35. Озерецковский Н.В., Королев Н.А. Использование средств структурирования содержания в разработке учебных материалов для повышения квалификации в области ИКТ. — <http://tm.ifmo.ru/tm2006/db/doc>
36. Сайт учителя информатики Афанасовой Людмилы Владимировны. Педагогический опыт по теме "Развитие алгоритмического мышления учащихся на уроках информатики" (районный банк данных, 2003 год) URL: <http://afanasova-mila.narod.ru/images/apo2003.doc> (дата обращения: 2.06.2017)
37. Сибнет. Adobe Flash Professional. URL: <http://soft.sibnet.ru/soft/10705-adobe-flash-cs3-professional> (дата обращения: 1.04.2015)
38. Сидоров С.В. Ментальные карты на лекции по педагогике. — Сайт педагога-исследователя. URL: <http://sv-sidorov.ucoz.com> (дата обращения: 1.04.2015)

Приложения

Приложение А



Рис.1

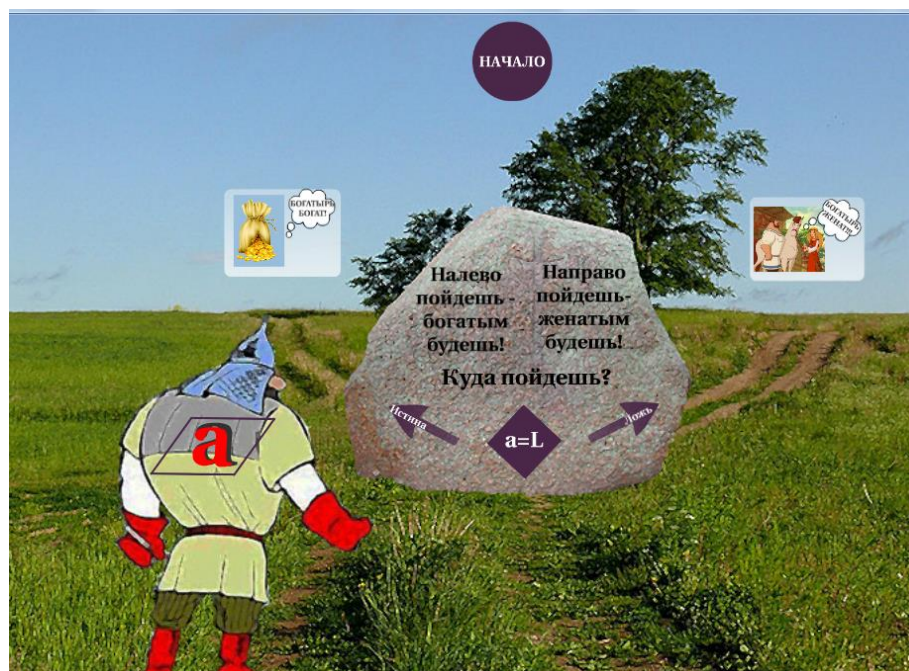


Рис.2



Рис.3

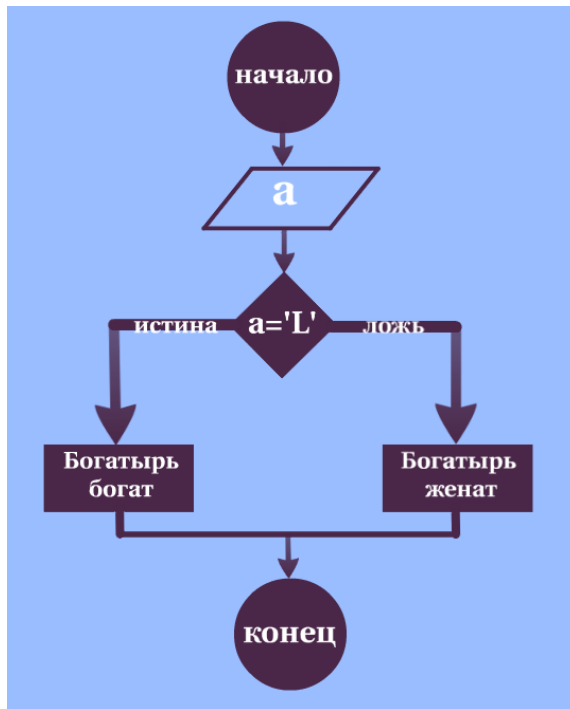


Рис.4

```
uses crt;  
var a:char;  
begin  
    writeln ('Куда пойдём? L/R');  
    readln (a);  
    if a='L'  
        then writeln ('Богат')  
        else writeln ('Женат');  
end.
```

Рис.5

Приложение Б

**План – конспект урока по информатике (учебник Боссовой Л.Л. 9 класс,
1 часть, 2012г)**

Тема урока: Условный оператор.

Цель урока:

Образовательная: ознакомление учащихся с оператором ветвления, формой и типами записи оператора ветвления на Паскале.

Развивающая: формирование умения записи оператора ветвления в среде программирования Паскаль; развитие логического мышления.

Воспитательная: повышение мотивации к изучению предмета информатика.

Оборудование: компьютерный класс с установленным ПО, ментальная карта по теме «Ветвление», 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)

3. Объяснение нового материала.

Условный оператор используется для выполнения одного из двух возможных вариантов программы.

Формат записи оператор ветвления:

if логическое_условие then оператор_1 else оператор_2;	если логическое_условие верно то выполняется оператор_1 , иначе оператор_2;
---	---

Перед else точка с запятой не ставится!

Формат полного оператора ветвления:	Формат неполного оператора ветвления:
if <логическое выражение> then <оператор 1> else <оператор 2>;	if <логическое выражение> then <оператор>

Логическое выражение:

Простой формой логического выражения является операция отношения:

- < меньше,
- > больше,
- <= меньше или равно,

- \geq больше или равно,
- $=$ равно,
- \neq не равно.

Сложные формы логических выражений составляются с использованием логических операций:

- not логическое отрицание (НЕ),
- and логическое умножение (И),
- or логическое сложение (ИЛИ),

Порядок действий:

- Отрицание, умножение, сложение, операции отношений.

Операторные скобки:

- Если после слов then или else нужно выполнить не один оператор, а несколько, то эти операторы заключают в операторные скобки: begin ... end
- Конструкция такого вида:
Begin <последовательность операторов> end
относится к *составным операторам*.
- Операторы ветвления могут быть вложены друг в друга, необходимо только следить за тем, чтобы then и else одного и того же оператора располагались друг под другом.

В качестве примера рассмотрим следующую задачу:

Известный из сказок богатырь стоит перед камнем. Ему нужно сделать выбор: пойти налево и быть богатым или пойти направо и быть женатым.

Рассмотрим решение данной задачи с помощью ментальной карты.

- (1 этап ментальной карты) Что изображено на рисунке? (Богатырь стоит перед камнем, на котором написано "Налево пойдешь - богатым будешь! Направо пойдешь - женатым будешь! Куда пойдешь?")

- Правильно! А может ли богатырь одновременно быть "богатым" и "женатым"? (Нет, потому что он не может одновременно пойти в 2 стороны)

- Хорошо! Значит при выполнении условия возможен только один исход, т.е. если богатырь пойдет налево, он будет богатым и не будет женатым. А если богатырь выберет путь направо, то он будет женатым и не будет богатым.

- (2 этап ментальной карты) Что изменилось на картинке? (Появились части блок-схемы)

- Давайте разберём, из чего она состоит. Чем в этой схеме является богатырь? (переменной a)

- А что представляет собой камень? (Условие)

- Какое условие представлено на карте? ($a=L$)

- Как вы думаете, что обозначает L ? (лево)

- Молодцы! А могло бы быть другое условие? (да, $a=R$, на право)

- Для представленного на картинке условия, какое из возможных исходов верное? (пойти налево)

- Следовательно, что должно быть написано на левой стрелке? (Истина)

- А тогда на правой? (ложь)

- (3 этап ментальной карты) Теперь рассмотрим следующий рисунок. Чем отличается эта картинка от предыдущей? (Вместо блок-схемы написана программа)

- Правильно! Только эта не вся программа, а лишь её часть. Назовем это элементами программного кода. Для начала давайте определим тип переменной. Какие значения может принимать переменная a ? (L или R)

- А какой тип данных предусматривает введение символов? ($char$)

- Тип данных определили, теперь перейдем к условию. С какого служебного слова начинается конструкция ветвления? (с if , если)

- Далее прописываем условие. Какое служебное слово прописывается после условия? ($then$)

- Что оно обозначает? (Если условие истинно, то...)

- Если условие истинно, то выводится результат и программа завершена. А если условие ложно, то какое служебное слово напишем? ($else$)

- Если условие ложно, то выводится соответствующий результат и программа завершается.

- (4 этап ментальной карты) На основе наших рассуждений несложно составить блок-схему, по которой можно написать программу. Попробуйте самостоятельно на основе схемы составить программу с помощью языка программирования Паскаль. На выполнение задания 5 минут.

- (5 этап ментальной карты) У кого получилось? Программа данной задачи представлена на доске, сверьте этот код со своим. (при необходимости обсудить ошибки с учениками)

4. Выполнение самостоятельной практической работы.

1. Дано целое число. Если оно является положительным то прибавить к нему 1, в противном случае вычесть из него два. Вывести полученное число.

2. Дано целое число. Если оно меньше 10, то умножить его на 5, в противном случае разделить на 20. Вывести полученное число.

5. Домашнее задание: стр. 202 № 5а

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

Приложение В

План – конспект урока по информатике (учебник Босовой Л.Л. 9 класс, 1 часть, 2012г.)

Тема урока: Условный оператор.

Цель урока:

Образовательная: закрепление знаний учащихся об операторе ветвления, формой и типами записи оператора ветвления на Паскале.

Развивающая: формирование умения записи оператора ветвления в среде программирования Паскаль; развитие логического мышления.

Воспитательная: повышение мотивации к изучению предмета информатика.

Оборудование: компьютерный класс с установленным ПО, ментальная карта по теме «Ветвление», проектор, экран.

Тип урока: урок закрепления 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. Закрепление изученного материала. (5 мин).
3. Выполнение практической работы (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 условный оператор?	if
2. Какое служебное слово отвечает за истинность условия?	then
3. Какое служебное слово отвечает за истинность условия?	else
4. Обязательными ли являются в конструкции условия слова then и else?	then - обязательное, else - необязательное
5. Отличие полной записи оператора условия от неполной?	В полной then и else, а в неполной только then
6. Перед какими служебными словами не ставится точка с запятой?	Перед if и then

3. Закрепление изученного материала.

- Для выполнения сегодняшней практической работы, нам необходимо вспомнить изученную на прошлом занятии ментальную карту. Давайте пройдемся по её структуре:

- (1 этап ментальной карты) В чем состоит первый этап ментальной карты? (на экране представлена картинка, отражающая суть задачи)

- (2 этап ментальной карты) Что добавляем на втором этапе? (добавили в картинку элементы блок-схемы, где богатырь является переменной, а камень - это блок условия)

- (3 этап ментальной карты) Что происходит на третьем этапе? (замена элементов блок-схемы на программные коды)

- (4-5 этап ментальной карты) Что делаем на последних двух этапах? (составляем блок-схему, по которой пишем программу)

4. Выполнение самостоятельной практической работы.

Сегодняшняя ваша практическая работа заключается в том, что вам нужно будет разбиться на 4 группы и совместно придумать задачу по теме "Ветвление" и разработать для этой задачи свою ментальную карту.

5. Домашнее задание: Оформить ментальную карту в виде плаката и подготовить её презентацию.

6. Подведение итогов занятия.

Приложение Г

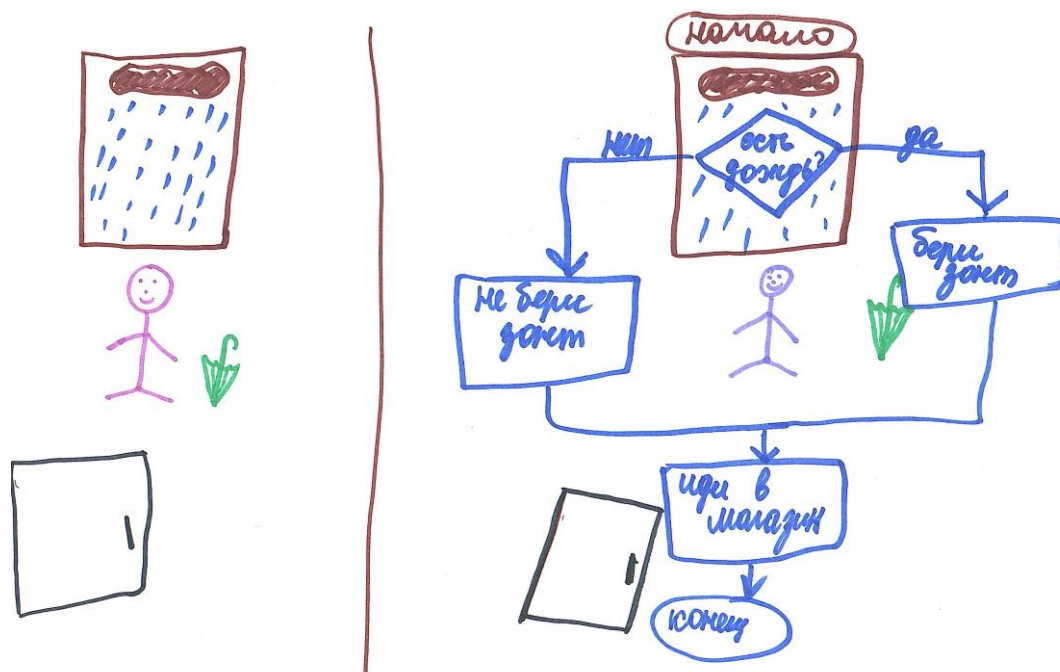


Рис.1.

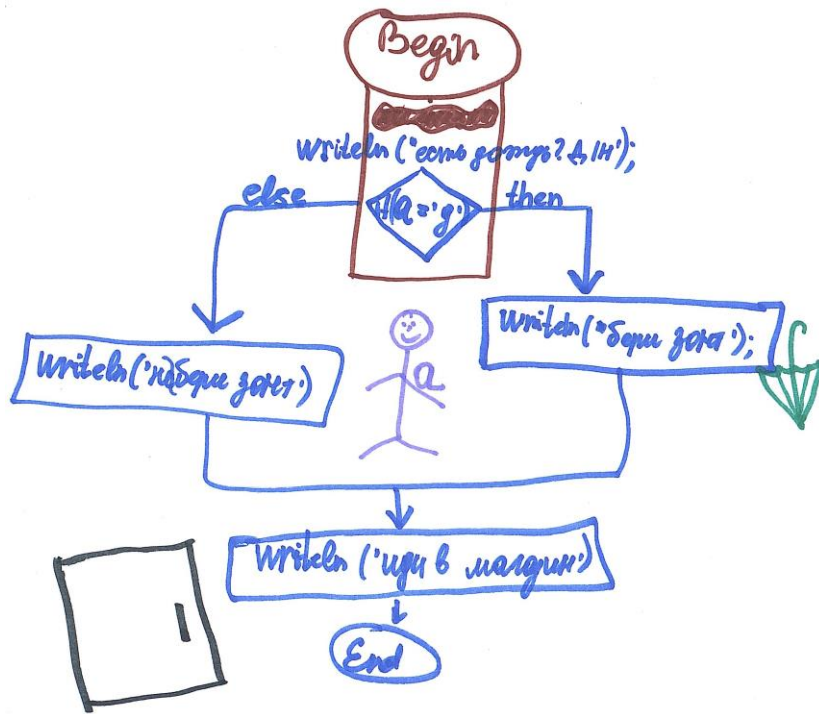


Рис.2.



Рис.3.

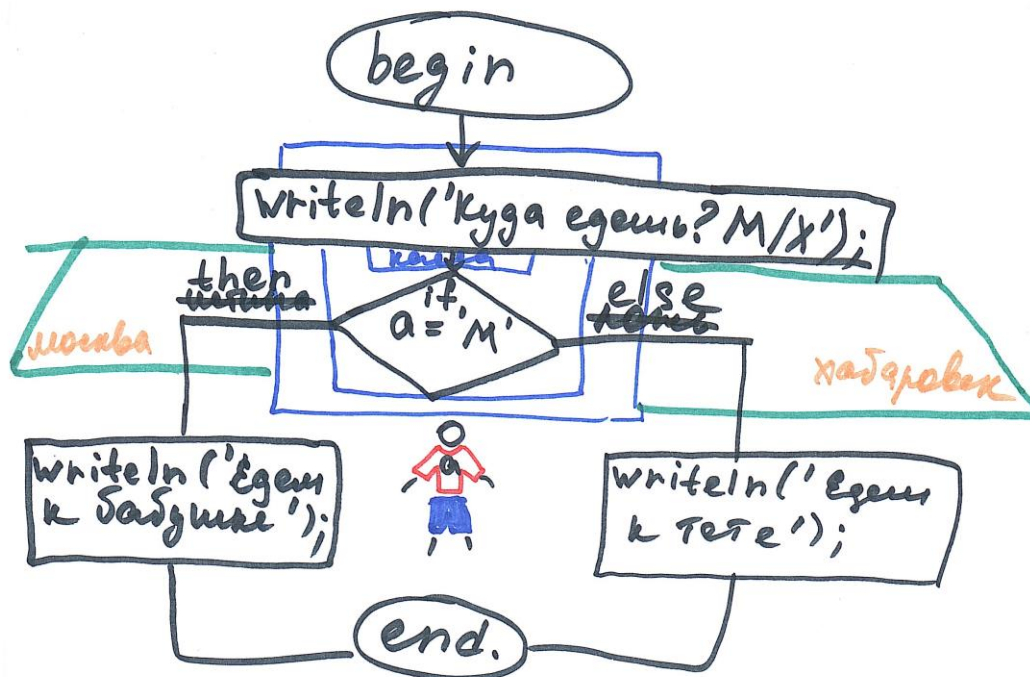


Рис.4.

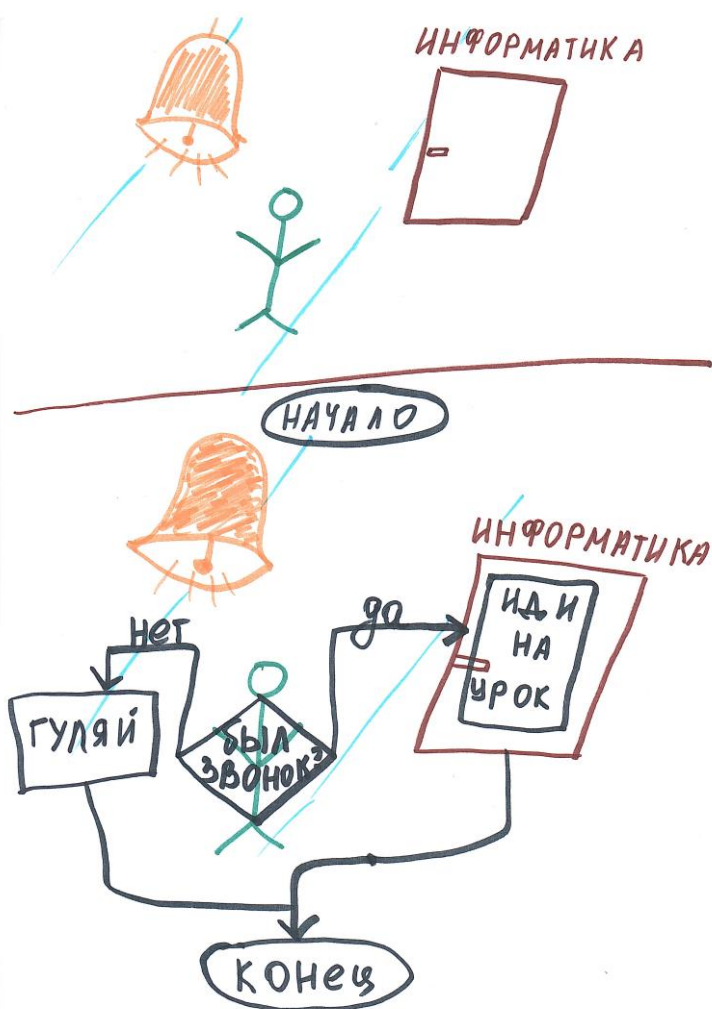


Рис.5.

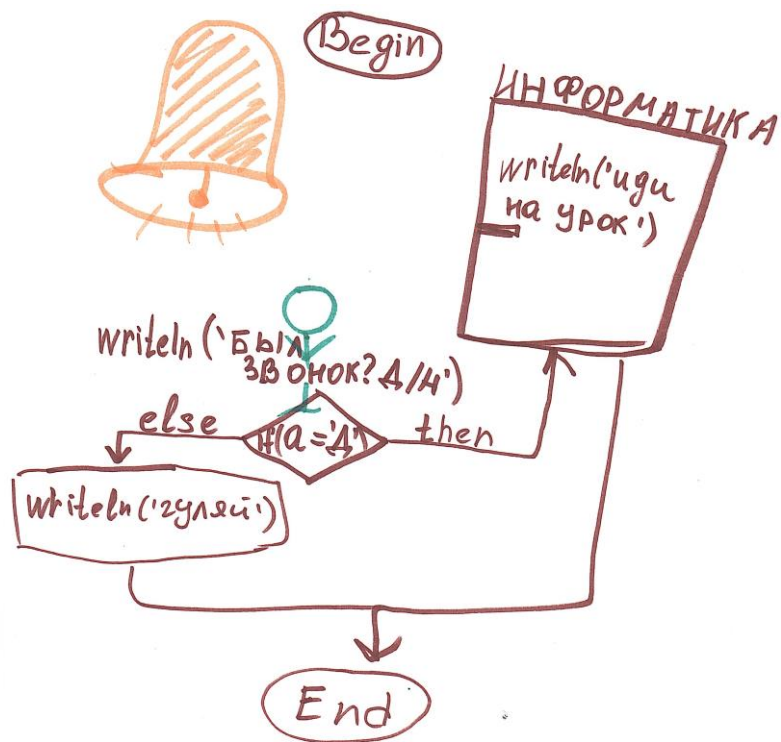


Рис. 6