

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –
филиал Сибирского федерального университета

Высшей математики, информатики и естествознания
кафедра

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

44.03.01 Педагогическое образование
код и наименование направления

**ОБУЧЕНИЕ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЮ В
ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ НА ОСНОВЕ МЕНТАЛЬНОГО
ПОДХОДА**
тема

Руководитель



подпись

Е.В. Киргизова
инициалы, фамилия

Выпускник



подпись

Д.О. Савельев
инициалы, фамилия

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –
филиал Сибирского федерального университета

Высшей математики, информатики и естествознания
кафедра

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

44.03.01 Педагогическое образование
код и наименование направления

ОБУЧЕНИЕ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЮ В
ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ НА ОСНОВЕ МЕНТАЛЬНОГО
ПОДХОДА
тема

Работа защищена « 24 » июня 2019 г. с оценкой « хорошо »

Председатель ГЭК



подпись

А.М. Гилязутдинова

Члены ГЭК

подпись

Е.В. Киргизова

подпись

Е.Н. Яковлева

подпись

А.А. Степанов

подпись

В.В. Фирер

Руководитель

подпись

Е.В. Киргизова
инициалы, фамилия

Выпускник

подпись

Д.О. Савельев
инициалы, фамилия

РЕФЕРАТ

Выпускная квалификационная работа по теме «ОБУЧЕНИЕ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЮ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ НА ОСНОВЕ МЕНТАЛЬНОГО ПОДХОДА» содержит 62 страницы текстового документа, 40 использованных источников, 2 таблицы, 2 приложения.

МЕНТАЛЬНЫЙ ПОДХОД, МЕНТАЛЬНЫЕ КАРТЫ, АЛГОРИТМИЗАЦИЯ, ПРОГРАММИРОВАНИЕ.

Актуальность и практический аспект данной проблемы связаны с необходимостью поиска новых эффективных подходов к обучению. Одним из перспективных видится ментальный подход.

Цель исследования: изучить и рассмотреть применение ментального подхода при изучении основ алгоритмизации и программирования в курсе информатики.

Объект исследования: Ментальный подход

Предмет исследования: Процесс обучения основам алгоритмизации и программирования на основе ментального подхода

Задачи исследования:

- 1) Выявить особенности и существующие проблемы изучения линии алгоритмизации и программирования
- 2) Проанализировать содержание линии алгоритмизации и программирования в школьном курсе информатики.
- 3) Рассмотреть теоретические основы ментального подхода к обучению программированию.
- 4) Разработать ментальные карты для обучения основам алгоритмизации и программирования по информатике.

В результате исследования было проанализировано содержание линии алгоритмизации и программирования, выявлены особенности и существующие проблемы, рассмотрены теоретические основы ментального подхода к обучению программированию.

СОДЕРЖАНИЕ

Введение.....	5
1 Теоретические основы обучения основам алгоритмизации и программирования в школе.....	7
1.1 Обучение основам программирования и алгоритмизации в школе и существующие проблемы.....	7
1.2 Анализ содержания линии алгоритмизации и программирования в школьном курсе информатики	11
1.3 Сущность ментального подхода к обучению программированию.....	20
2 Методическое обеспечение для изучения основ алгоритмизации и программирования в базовом курсе информатики.....	28
2.1 Пример тематического планирования по основам алгоритмизации и программирования	28
2.2 Пример применения дидактических материалов по основам алгоритмизации и программирования	29
2.3 Разработка ментальной карты по содержательной линии алгоритмизация и программирование	35
Заключение	51
Список использованных источников	56
Приложение А Ментальная карта «Основы алгоритмизации»	61
Приложение Б Ментальная карта «Начала программирования»	62

ВВЕДЕНИЕ

Раздел «Алгоритмизация и программирование» в базовом школьном курсе информатики занимает одно из центральных мест. Однако до сих пор нет достаточно определенного мнения в вопросах, касающихся более частных аспектов преподавания этого раздела. Авторы Л.Л. Босова, Н.Д. Угринович, И.Г. Семакин учебно-методических комплектов и программ по курсу «Информатика и ИКТ» по-разному подходят к изучению теоретической и практической составляющей содержательной линии «Алгоритмизация и программирование». Основопологающим в разделе «Программирование» – является алгоритмизация. Составление программ на ЭВМ начинается с построения алгоритма, важнейшим качеством профессионального программиста является развитое алгоритмическое мышление.

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.

Актуальность и практический аспект данной проблемы связаны с необходимостью поиска новых эффективных подходов к обучению. Одним из перспективных видится ментальный подход.

Цель исследования: изучить и рассмотреть применение ментального подхода при изучении основ алгоритмизации и программирования в курсе информатики.

Объект исследования: Ментальный подход

Предмет исследования: Процесс обучения основам алгоритмизации и программирования на основе ментального подхода

Задачи исследования:

- 1) Выявить особенности и существующие проблемы изучения линии алгоритмизации и программирования
- 2) Проанализировать содержание линии алгоритмизации и программирования в школьном курсе информатики.
- 3) Рассмотреть теоретические основы ментального подхода к обучению программированию.
- 4) Разработать ментальные карты для обучения основам алгоритмизации и программирования по информатике.

Методологической основой исследования являются работы Т. Бьюнзена и Д. Новака по ментальному подходу и ментальным картам.

Методы исследования: анализ учебной, учебно-методической, педагогической литературы.

Теоретическая значимость – состоит в том, что обоснована возможность и необходимость ментального подхода к обучению теме «Алгоритмизация и программирование»;

Практическая значимость – состоит в том, что разработаны ментальные карты как средства реализации ментального подхода к обучению теме «Алгоритмизация и программирование».

По результатам исследования на сайте infourok.ru опубликована статья на тему «Обучение алгоритмизации и программированию в школьном курсе информатики на основе ментального подхода».

Глава 1 Теоретические основы обучения основам алгоритмизации и программирования в школе

1.1 Обучение основам алгоритмизации и программирования в школе и существующие проблемы

В соответствии с примерной рабочей программой общего образования по информатике и ИКТ, понятие алгоритма является одним из важнейших понятий курса информатики и информационных технологий основной школы». Соответственно, обучение алгоритмизации и программированию, т.е. способам реализации алгоритмов на ЭВМ, является одним из основных компонентов школьного курса информатики.

При этом по оценке ряда специалистов, ситуация, сложившаяся вокруг преподавания программирования в школе, является проблемной. «Нынешнее положение курса алгоритмизации и программирования в школе можно назвать незавидным. Тенденция к его сокращению вплоть до полного вытеснения явно прослеживается не только в периодически публикуемых программах различных авторов, но и в отдельных инструктивно-методических документах» [2].

Обратим внимание и на содержательный аспект. В настоящее время существуют три наиболее распространенных подхода к преподаванию программирования:

- Преподавание программирования как теоретической дисциплины без привязки к конкретным языкам и системам;
- Преподавание на основе специально разработанного языка, ориентированного на обучение основным навыкам программирования;
- Изучение одного или нескольких языков программирования, широко используемых при решении научных и хозяйственных задач (такие языки можно назвать стандартными).

Наиболее приемлемым для общеобразовательной школы, где курс информатики преподается в 8-11 классах, является сочетание первого и третьего подходов - обучение теоретическим основам программирования на базе стандартного языка. При этом не обязательно вдаваться в глубины языка.

Учащиеся, которых он заинтересует, могут сделать это и сами. Наибольшее внимание следует уделить переходу от алгоритмических структур к их программной реализации на языке программирования [1]. При этом все же, следует, как можно больше внимания уделять именно самим алгоритмам, в этом случае ученику в будущем будет легче перейти с одного языка программирования на другой.

Федеральный базисный учебный план для образовательных учреждений РФ, разработанный на основе федерального государственного образовательного стандарта, отводит 105 часов для обязательного изучения информатики и информационных технологий на ступени основного общего образования и 70 часов на ступени полного общего образования. На алгоритмизацию и программирование отводится около 19 часов. В случае профильного изучения информатики ситуация значительно улучшается. Федеральный базисный учебный план отводит 280 часов для обязательного изучения информатики и информационных технологий на ступени среднего общего образования, то есть по 140 учебных часов на каждый год обучения. Как мы видим, на изучение раздела программирования отводится недостаточное количество времени, это приводит к тому, что изучение некоторых тем проходит поверхностно, а некоторые исключаются вовсе [14].

Следует обратить внимание, и на затруднение в освоении алгоритмизации и программирования у значительной части учащихся. Данная тема, и с точки зрения учителей, и с точки зрения учеников, является сложнейшей в рамках учебного предмета. К сожалению, необходимо отметить и низкий уровень квалификации многих учителей информатики, что не способствует качественному освоению предмета.

Перед началом обучения учитель сталкивается с проблемой выбора языка программирования для изучения. Мнения о том, какой язык программирования лучше преподавать в школе, разнятся: от того, что программирование изучать не нужно, а следует просто поднимать компьютерную грамотность и осваивать офисные программы, до того, что нужно изучать операционные системы и

несколько языков программирования с различными парадигмами. Это крайние случаи, но золотую середину найти непросто. Так же есть мнение, что не так важно, какой язык программирования взять: на уроках информатики нужно учить не языку программирования, а методам программирования и системному подходу решения задач. Нужно развивать алгоритмическое мышление и на примерах знакомиться с принципами построения современных компьютерных систем.

У каждого учителя есть свой список требований к учебному языку программирования. Набор требований может быть, например, таким: простой, интуитивный синтаксис, наличие высокоуровневых инструментов для обнаружения и недопущения ошибок и для отладки программ, наличие качественной документации с примерами, наличие дружелюбной среды разработки, кроссплатформенность (наличие версий под различные платформы) [17].

С одной стороны, школьная программа не предполагает подготовку специалистов и базовые понятия алгоритмизации можно традиционно давать на Паскале. Для школы нет особой разницы, какой именно язык учить, база у всех языков идентична, надстройку над базой дают уже в профильных учебных заведениях. Ученику достаточно знать набор базовых понятий бинарной логики, условий, циклов и т. п. а они присутствуют практически в любом языке программирования.

Проблема кроется в том, что в школе стоит задача ознакомить с тем, что такое программирование и дать примерное представление о том, как это делается. С другой стороны ИТ-технологии развиваются очень быстро, появляются новые платформы, языки программирования (Java, C#, Python...), технологии. Современному программисту приходится постоянно «держат руку на пульсе». Именно поэтому программу обучения школьников программированию нужно тщательно продумывать со взглядом в будущее.

Подходы к программированию в частности и информационным технологиям, в общем, постоянно изменяются, но что-то остается неизменным.

Эту базу необходимо передавать учащимся, в первую очередь, чтобы школьники учились самостоятельно искать нужный материал и изучать его. Нужно учить базовые вещи, которые будут востребованы всегда, независимо от технологий. В программировании есть несколько парадигм, подходов к созданию программ: модульная парадигма, процедурные парадигма, объектно-ориентированная парадигма.

В школах обычно преподают процедурный подход к программированию, в то время как ООП (объектно-ориентированное программирование) – очень редко или никогда. Выбор языка и системы программирования имеет принципиальное значение, т. к. от этого во многом зависит методика изучения курса, содержание и последовательность предъявления учебного материала, система учебных заданий и, главное, вся дальнейшая работа по овладению программированием для решения реальных практических задач на компьютере. От этого выбора напрямую зависит доступность восприятия, изучения и овладения учащимся приемами и методами программирования.

Какие из современных используемых на практике языков годятся на роль первого языка? Наверное, это императивный, строго типизированный, со статической типизацией и строгой семантикой язык. В последнее время наметилась явная тенденция перехода обучения от алгоритмических к объектно-ориентированным языкам с использованием визуальных систем программирования (Delphi, Visual Basic, Visual C++ и т. д.) [17].

Учитывая приведенные выше факторы, представляется актуальной задача совершенствования методики преподавания программирования в средней школе.

Важную роль в методике обучения программированию, следует отводить самостоятельной работе учеников, так как только самостоятельная разработка алгоритмов и программ, должным образом способствует развитию алгоритмического мышлению и закреплению необходимых навыков.

1.2 Анализ содержания линии алгоритмизации и программирования в школьном курсе информатики

В федеральном образовательном стандарте по информатике тема алгоритмизации присутствует в разделе «Алгоритмы и начала программирования»: «Алгоритм, свойства алгоритмов; блок-схемы. Алгоритмические конструкции. Логические значения, операции, выражения. Разбиение задачи на подзадачи, вспомогательный алгоритм». Программирование упоминается лишь одной фразой: «Представление о программировании» в подразделе компьютер.

В примерной программе разделительной линии «Алгоритмы и исполнители» выделяется 19 часов. Содержание темы расписано более подробно, чем в стандарте: Алгоритм. Свойства алгоритма. Способы записи алгоритмов; блок-схемы. Возможность автоматизации деятельности человека. Исполнители алгоритмов (назначение, среда, режим работы, система команд). Компьютер как формальный исполнитель алгоритмов (программ). Алгоритмические конструкции: следование, ветвление, повторение. Разбиение задачи на подзадачи, вспомогательный алгоритм. Алгоритмы работы с величинами: типы данных, ввод и вывод данных».

В этом же разделе говорится о программировании: «Языки программирования, их классификация. Правила представления данных. Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла. Правила записи программы. Этапы разработки программы: алгоритмизация – кодирование – отладка – тестирование» [1]. Далее рассмотрим как разные авторы учебников по информатике подходят к изучению алгоритмизации и программирования.

В учебнике Л.Л. Босовой [3] разделы, относящиеся к алгоритмизации и программированию, имеют следующее содержание:

2. Основы алгоритмизации

2.5. Алгоритмы и исполнители

2.5.1. Понятие алгоритма

- 2.5.2. Исполнитель алгоритма
- 2.5.3. Свойства алгоритма
- 2.5.4. Возможность автоматизации деятельности человека
- 2.6. Способы записи алгоритмов
 - 2.6.1. Словесные способы записи алгоритма
 - 2.6.2. Блок-схемы
 - 2.6.3. Алгоритмические языки
- 2.7. Объекты алгоритмов
 - 2.7.1. Величины
 - 2.7.2. Выражения
 - 2.7.3. Команда присваивания
 - 2.7.4. Табличные величины
- 2.8. Основные алгоритмические конструкции
 - 2.8.1. Следование
 - 2.8.2. Ветвление
 - 2.8.3. Повторение
- 3. Начала программирования**
 - 3.5. Общие сведения о языке программирования Паскаль
 - 3.5.1. Алфавит и словарь языка
 - 3.5.2. Типы данных, используемые в языке Паскаль
 - 3.5.3. Структура программы на языке Паскаль
 - 3.5.4. Оператор присваивания
 - 3.6. Организация ввода и вывода данных
 - 3.6.1. Вывод данных
 - 3.6.2. Первая программа на языке Паскаль
 - 3.6.3. Ввод данных с клавиатуры
 - 3.7. Программирование линейных алгоритмов
 - 3.7.1. Числовые типы данных
 - 3.7.2. Целочисленный тип данных
 - 3.7.3. Символьный и строковый типы данных

- 3.7.4. Логический тип данных
- 3.8. Программирование разветвляющихся алгоритмов
 - 3.8.1. Условный оператор
 - 3.8.2. Составной оператор
 - 3.8.3. Многообразие способов записи ветвлений
- 3.9. Программирование циклических алгоритмов
 - 3.9.1. Программирование циклов с заданным условием продолжения работы
 - 3.9.2. Программирование циклов с заданным условием окончания работы
 - 3.9.3. Программирование циклов с заданным числом повторений
 - 3.9.4. Различные варианты программирования циклического алгоритма

Раздел относящийся к Алгоритмизации по содержанию почти совпадает с первой частью вышеописанной примерной программы, за исключением двух частей: компьютер как формальный исполнитель алгоритмов и разбиение задачи на подзадачи.

В разделе связанном с программированием не рассматриваются различные языки программирования и их классификация, данный раздел полностью относится только к языку программирования Паскаль, собственно в нем и рассматриваются соответствующие части из примерной программы и т.д.

В учебнике Н.Д. Угриновича [4] раздел, относящийся к алгоритмизации и программированию, имеет следующее содержание:

1. Основы алгоритмизации и объектно-ориентированного программирования

- 1.1. Алгоритм и его формальное исполнение
- 1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования
- 1.3. Переменные: тип, имя, значение
- 1.4. Арифметические, строковые и логические выражения

1.5. Функции в языках объектно-ориентированного и процедурного программирования

1.6. Графические возможности объектно-ориентированного языка программирования Visual Basic

Н.Д. Угринович в своем учебнике совмещает алгоритмизацию и программирование в одну главу, но ее содержание сильно отличается от ранее упомянутой примерной программы. В начале, конечно, говорится об алгоритмах, но довольно кратко. По большей части в этой главе говорится об объектно-ориентированном программировании и языке Visual Basic.

В учебнике И.Г. Семакина [5] разделы, относящиеся к алгоритмизации и программированию, имеют следующее содержание:

1. Управление и алгоритмы

1.1. Управление и кибернетика

1.2. Управление с обратной связью

1.3. Определение и свойства алгоритма

1.4. Графический учебный исполнитель

1.5. Вспомогательные алгоритмы и подпрограммы

1.6. Циклические алгоритмы

1.7. Ветвление и последовательная детализация алгоритма

2. Введение в программирование

2.1. Что такое программирование

2.2. Алгоритмы работы с величинами

2.3. Линейные вычислительные алгоритмы

2.4. Знакомство с языком Паскаль

2.5. Алгоритмы с ветвящейся структурой

2.6. Программирование ветвлений на Паскале

2.7. Программирование диалога с компьютером

2.8. Программирование циклов

2.9. Алгоритм Евклида

2.10. Таблицы и массивы

2.11. Массивы в Паскале

2.12. Одна задача обработки массива

2.13. Поиск наибольшего и наименьшего элементов массива

2.14. Сортировка массива

Большая часть учебника И.Г. Семакина [5] посвящена алгоритмизации и программированию. Раздел алгоритмизации отличается от программы в связи с тем что алгоритм там рассматривается с точки зрения кибернетики, а раздел программирования, так же как и у Л.Л. Босовой полностью посвящен языку программирования Паскаль и работе в нем.

Изучение алгоритмизации в школьной информатике может иметь два целевых аспекта: первый – развивающий аспект, под которым подразумевается развитие алгоритмического (еще говорят – операционного) мышления учащихся; второй – программистский аспект. Составление программы для ЭВМ начинается с построения алгоритма, важнейшим качеством профессионального программиста является развитое алгоритмическое мышление. Если в первом школьном учебном пособии по информатике в изучении алгоритмизации превалировал второй – программистский аспект, то в дальнейшем стала больше подчеркиваться развивающая роль данной темы.

Вопрос о месте и объеме темы программирования в базовом курсе был и остается дискуссионным. Здесь так же можно говорить о двух целевых аспектах, с которыми связано изучение программирования в школе.

Первый аспект связан с усилением фундаментальной компоненты курса информатики: раскрывается сущность программного управления работой компьютера. Ученикам дается представление о том, что такое языки программирования, что представляет собой программа на языках программирования высокого уровня, как создается программа в среде современной системы программирования.

Второй аспект носит профориентационный характер. Профессия программиста в наше время является достаточно распространенной и престижной. Изучение программирования в рамках школьного курса позволяет

ученика испытать свои способности к такого рода деятельности. Безусловно, в большей степени эту задачу может решать профильный курс информатики в старших классах [1].

На рисунке 1 представлена структура основных понятий содержательной линии «Алгоритмизация и программирование», которая разделена на две ветви: ветвь алгоритмизации и ветвь программирования. Эти ветви имеют общую часть, которая начинается с блока «Алгоритмы работы с величинами». Из схемы, в частности, что основой методики обучения алгоритмизации и программированию является методика структурного программирования.

Структура ветви программирования носит характер обобщенной методической схемы, которая применима при любом уровне изучения программирования. На разных уровнях изучения может отличаться глубина и степень подробности раскрытия различных разделов схемы.

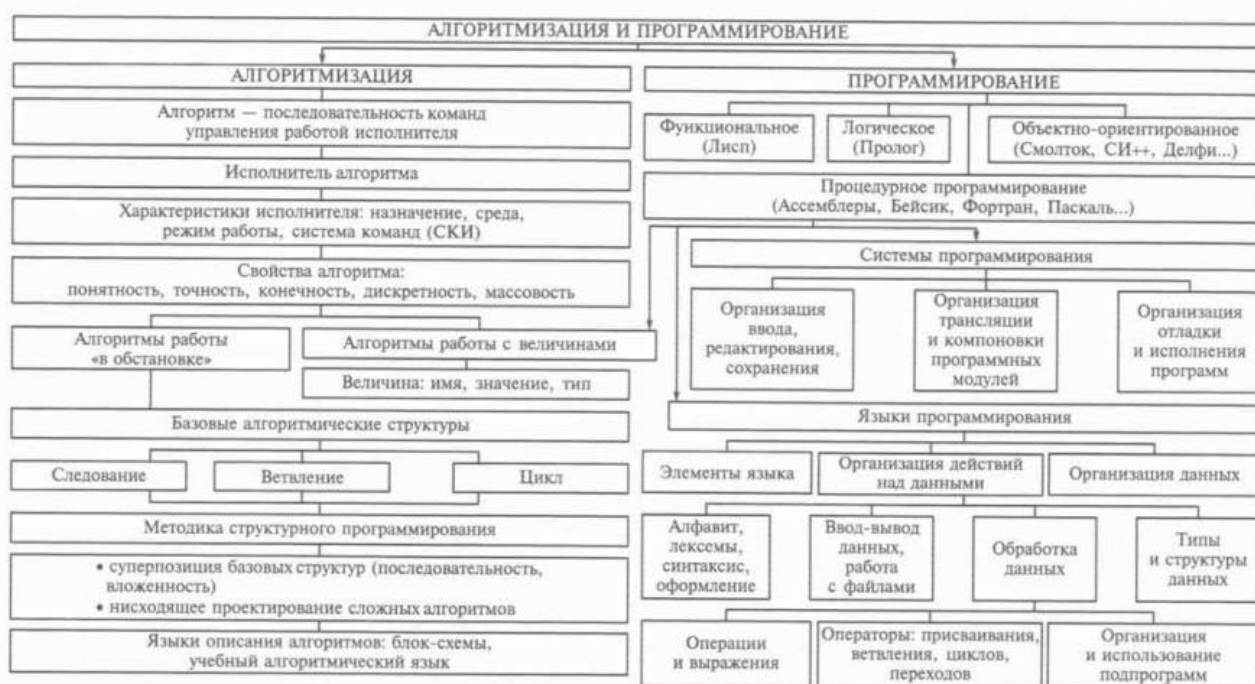


Рисунок – 1 Структура содержательной линии «Алгоритмизация и программирование»

Понятие алгоритма является центральным в первом школьном учебнике информатики [8]. Там приводится следующее определение: «Под алгоритмом понимают понятное и точное предписание (указание) исполнителю совершить

последовательность действий, направленных на достижение указанной цели или на решение поставленной задачи». Указание на выполнение каждого отдельного действия названо командой, а «совокупность команд, которые могут быть выполнены исполнителем, называется системой команд исполнителя». В качестве основного свойства алгоритма подчеркивается формальный характер работы исполнителя при его выполнении. Отсюда делается вывод о том, что исполнителем алгоритма может быть автомат (машина, робот). На этой идее основан принцип программного управления работой компьютера, поскольку программа — это и есть алгоритм, представленный на языке, «понятном» компьютеру — на языке программирования.

Сформулированные в учебнике [8] понятия явились дидактической основой для раскрытия темы алгоритмизации во всех последующих учебниках информатики.

В учебнике И.Г. Семакина применен кибернетический подход к теме алгоритмизации. Алгоритм трактуется как информационный компоненты системы управления. Такой подход дает возможность ввести в содержание базового курса новую содержательную линию – линию управления. Это многоплановая линия, которая позволяет затронуть следующие вопросы: элементы теоретической кибернетики – кибернетическая модель управления с обратной связью; элементы прикладной кибернетики – структура компьютерных систем автоматического управления (систем с программным управлением); назначение автоматизированных систем управления; основы теории алгоритмов.

Учебник Н.Д. Угриновича «Информатика. Базовый курс. 9 класс» трактует понятие «Алгоритм» как конечную последовательность действий, которая описывает процесс преобразования объекта из начального состояния в конечное, и записана с помощью точных и понятных исполнителю команд».

Для того чтобы соблюсти принцип инвариантности содержания по отношению к конкретным версиям программного обеспечения, в учебнике

описывается гипотетический учебный исполнитель, которому дано имя ГРИС – графический исполнитель. Это исполнитель, работающий в обстановке (то есть без использования величин). На примере ГРИС вводятся основные понятия алгоритмизации. Предлагаемая последовательность заданий способствует эффективному достижению основной цели раздела — освоения структурной методики построения алгоритмов.

Изучение алгоритмизации и программирования в базовом курсе Л.Л. Босовой поддерживается учебником [3] и учебным пособием [9].

В учебнике алгоритм позиционируется как один из этапов решения задач на компьютере и затрагивается лишь поверхностно, освещено лишь понятие алгоритма, его свойства и структуры не затрагиваются. Это связано с тем что в 8 классе большинство учебного времени отведено как раз таки на изучение алгоритмизации и программирования, соответственно именно там и раскрываются основные понятия этого раздела, а тот материал что присутствует в учебнике 9 класса больше нацелен на повторение и дополнение к ранее изученному.

Все понятия связанные с алгоритмизацией и программированием иллюстрируются как на примерах бытового характера, так и на примере формализованного исполнителя «Робот».

Материал для практической работы содержится в учебном пособии [9] в разделе «Алгоритмизация и программирование». Практическая работа учащихся заключается в решении задач на языке программирования Паскаль. Порядок изучения определяется последовательностью усложнения структур алгоритмов: линейные, циклические, ветвящиеся, включающие массивы, включающие процедуры.

Согласно тематическому плану, приведенному в методическом пособии для учителя [9] на изучение раздела «Алгоритмизация и программирование» в 8 и 9 классах отводится 29 часов.

На изучение раздела «Основы алгоритмизации и объектно-ориентированного программирования» по учебнику Н.Д. Угриновича [4]

отводится 14 часов в 9 классе, что согласуется с примерной программой по информатике и федеральным базисным учебным планом.

Общее определение и свойства алгоритма кратко описываются в подразделе 2.1. Учебные среды для освоения алгоритмизации не применяются. Об алгоритмических структурах и структурах данных рассказывается параллельно с описанием программирования на языке Visual Basic. Для описания алгоритмов используются блок-схемы, но не используется учебный алгоритмический язык.

В подходах к определению алгоритма в разных учебниках имеются заметные различия. В учебнике [3] при раскрытии понятия алгоритма рассматриваются этапы решения задач на компьютере, алгоритм представляется как один из шести этапов. Алгоритм это четкая инструкция, задающая необходимую последовательность действий для решения задачи.

В учебнике [5] при раскрытии понятия алгоритма рассматривается кибернетическая модель системы управления, в которой управляющий объект и объект управления связаны между собой каналами прямой и обратной связи. Алгоритм есть последовательность команд, которая передается от управляющего через канал прямой связи к объекту управления. Исполнителем алгоритма является объект управления, поскольку именно он осуществляет действия, предписанные командами алгоритма.

В учебнике [4] сказано: «Алгоритмы широко используются в технике в системах управления объектами. В любой системе управления существует управляющий объект, который является исполнителем алгоритма управления». Утверждение противоположно тому, что дается в [5]. Приводится пример управления обогревателем воздуха со стороны человека, наблюдающего за температурой в помещении. Делается утверждение, что исполнителем алгоритма является человек.

Подводя итог обзору особенностей изложения алгоритмизации и программирования в данных учебниках, можно отметить, что у каждого автора свой подход к изучению алгоритмизации и программирования и не все из них

придерживаются примерной программы, в их учебниках отразились результаты большой исследовательской работы в построении методической системы обучения алгоритмизации. Это одно из значительных научно-методических достижений в области теории и методики обучения информатике.

1.3. Сущность ментального подхода к обучению программированию

Прежде чем говорить о ментальном подходе к обучению, необходимо выделить исходное определение понятия «подход» как некоторой точки зрения, обуславливающей проектирование, организацию образовательного процесса. Подход в толковом словаре русского языка [18] означает «идти под низ чего-то», т.е. находиться в основе чего-то.

По своему определению термин «подход к обучению» многозначен – это:

- а) мировоззренческая категория, в которой отражаются социальные установки субъектов обучения как носителей общественного сознания;
- б) глобальная и системная организация и самоорганизация образовательного процесса, включающая все его компоненты и, прежде всего, самих объектов педагогического взаимодействия: учителя и ученика [19].

Ментальный подход к обучению предполагает смещение целеполагания учебного процесса в сторону развития когнитивных способностей обучаемых, в частности, если рассматривать обучение программированию – то основной целью будет являться развитие алгоритмического стиля мышления.

Алгоритмический способ мышления позволяет принимать оптимальные решения в любой сфере человеческой деятельности, и сам по себе никак не взаимодействует с программированием и вычислительной техникой. В таком некоем виде он существовал постоянно, то есть изначально принадлежал человеческому мышлению. Благодаря появлению вычислительной техники и профессии программиста привело к тому, что определенный круг специалистов нуждается в необходимости алгоритмического способа мышления.

Алгоритмический способ для выполнения программистских задач является единственно возможным.

Общий алгоритмический подход в программистской практике углубляется и детализируется: структура предметной области становится формализованной информационной структурой, в ней вычленяются количественные взаимосвязи, образующие математическую модель предметной области, превращение алгоритма в компьютерную программу. Т. е. у людей программистов, алгоритмическое мышление развито на более высоком, профессиональном уровне. Для того, чтобы сформировать алгоритмическое мышление у школьников, на повседневном уровне, у самого учителя оно должно быть развито на профессиональном уровне. Кроме того, современные подходы к совершенствованию системы среднего образования предъявляют высокие требования к уровню предметной подготовки учителя. Несомненно, его педагогический кругозор должен намного превышать рамки школьной программы. Следовательно, учитель информатики должен владеть всеми современными технологиями программирования.

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.

Наши возможности ориентироваться в пространстве и во времени, осуществлять деятельность, говорят 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 [21]. В 60-е годы теорию развил профессор Корнелльского университета Джозеф Новак [21]. 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.

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 [22].

Метод ментальных карт представляет собой особую технику визуализации мышления, построенную на создании эффективных альтернативных записей. Существуют и другие названия данного метода: «карты ума», «интеллект-карты», «карты разума», «карты памяти».

Тони Бьюзен исследовал мыслительные системы, которые были присущи людям в эпоху античности и Ренессанса. Поскольку в эти времена было создано множество культурных объектов, литературных произведений. Т. Бьюзен заметил, что, создавая свои записи, они использовали фантазию и следовали ассоциативным связям.

Возможно, именно поэтому их записи были способны донести информацию не только их создателю, но и любому человеку, даже по

истечении нескольких столетий. Также психолог оценил значимость рисунка оформления собственных мыслей.

Основной смысл методики ментальных карт заключается в получение рисунка, где в центре выделено основное понятие, от которого потом ответвляются задачи, идеи, отдельные мысли и шаги, необходимые для реализации конкретного проекта или задумки. Как и основная ветвь ассоциативных связей, мелкие ветки могут делиться еще на несколько более мелких. Таким образом, ментальная карта отображает все ассоциативные связи в мыслительном процессе ее создателя.

Данная методика базируется на принципе «радиантного мышления», которое связано с ассоциативными мыслительными процессами.

Отправная точка - это центральный объект (мысль, идея, задача). Радиант - это точка небесной сферы, от которой как будто отходят видимые пути тел, движущихся с одинаково направленными скоростями. Отсюда можно сделать вывод, что «радиантное мышление» отражает бесконечное множество всех возможных ассоциаций, а ментальные карты позволяют зафиксировать их на различных носителях.

Полученные результаты для наглядности обычно фиксируются на бумаге. Чтобы лучше воспринимать информацию с ментальной карты, рекомендуется оформлять все записи разными способами, например, используя различные цвета, формы, картинки. Такое наглядное оформление позволяет структурировать и группировать информацию, делая ее наглядной и более понятной [24].

Методика «ментальных карт» - это метод целостного мышления. Известно, что за логическое мышление человека отвечает левое полушарие мозга. Значит, при решении какой либо стандартной проблемы будет активизировано именно оно. В то время как правое, отвечающее за творчество и образное мышление, не будет вовлечено в рабочий процесс.

Метод ментальных карт позволяет одновременно задействовать оба, что и означает целостность мышления.

Стоит помнить о том, что у каждого человека индивидуальный способ мышления и ментальная карта должна отображать такую особенность. Нужно максимально эффективно использовать пространство ментальной карты, не оставляя пустого пространства, но и не перегружая рисунок. Наиболее оптимальным расположением является горизонтальное расположение карты.

Карта считается законченной, если она выглядит цельной. Это значит, что ее создатель произвел комплексный анализ проблемы и разобрался в ней. Если же какая-либо ветвь ментальной карты выглядит незаконченной, стоит продолжить анализ и ассоциативный ряд.

Такой способ представления информации имеет свои преимущества и недостатки (См. Таблица 1).

Таблица 1 – Преимущества и недостатки ментальных карт

Преимущества	Недостатки
Представленная информация меньше по объему, ее легко фиксировать и анализировать.	Человеку, не составляющему карту и увидевшему ее в первый раз достаточно трудно понять ее содержание
При чтении карты видны взаимосвязи в каждом блоке, их структура и логика	
При использовании метода развивается творческое и логическое мышление, воображение и память, поскольку используются оба полушария человеческого мозга	

Ментальная карта представляет собой идеальное решение для проверки знаний учащихся и помощника при планировании, выполнении и осуществлении контроля усвоенных знаний учащимися.

Примером использования ментальных карт в образовании является подготовка на их основе занятий. Занятие в форме ментальной карты гораздо легче подготовить, нежели написать его «линейный» вариант, кроме того, она

предоставляет как педагогу, так и учащемуся то большое преимущество, что все содержание занятия оказывается, как говорится, на ладони.

Графический метод представления информации увлекает учащихся, позволяет им лучше запомнить и усвоить излагаемый материал. После занятия ученикам могут быть розданы черно-белые копии ментальной карты, представленной на занятии педагогом, содержащие лишь ее остов, и предложено заполнить их по памяти и раскрасить.

Ментальные карты можно широко использовать при разработке плана проведения занятия. Бывает трудно отобрать из того наработанного материала, который имеется у педагога те вопросы, задачи, упражнения, которые подходят именно данной категории учащихся. Тогда, прорисовывая основные вопросы занятия и относя к нему все упражнения, которые позволяют обеспечить его усвоение, можно при их наглядном перечне выбрать необходимые, пометить их маркером, отметить те, которые являются запасными, требуют вводного теоретического материала или иного последующего действия. Составление ментальной карты позволяет увидеть лимит времени, видеть запас и возможные изменения хода занятия, отбирать задания и их последовательность.

Ментальная карта занятия дает четкий план действий по реализации поставленных целей. На карте легко увидеть, что уже сделано, а что еще предстоит сделать, видны моменты, требующие доработки.

Ментальные карты можно использовать для обратной связи. Достаточно эффективной технологией составления интеллект-карт представляется для проверки качества усвоения материала учащимися. На этапе рефлексии учебного занятия его участникам может быть предложено в течение короткого промежутка времени надписать ответвления от центральной темы. Количество основных ветвей выбирается преподавателем и соответствует целям лекции. Количество ответвлений второго уровня может быть определено каждым автором карты самостоятельно. Особенность интеллект-карт заключается в том, что они редко могут быть завершены полностью, они постоянно дополняются и расширяются по мере того, как возникают новые идеи.

Ментальная карта представляет собой идеальное решение для проверки знаний учащихся и играет роль помощника при планировании, выполнении, осуществлении контроля и защите проектных работ. Такие карты несут информацию о реальных результатах. Постепенно внедрение этой техники поможет разрабатывать разнообразные занятия с учетом категории учащихся, их возможностей и запросов, ведь любой новый инструмент, когда его только показываешь, вызывает чаще всего отторжение, так как мы привыкаем работать с каким-то инструментом, а выход на новый уровень связан в первую очередь с напряжением внутренних сил, с затратой времени [23].

В подавляющем большинстве работ, посвященных использованию ментальных карт в учебном процессе [22,25,26,27], 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 были 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 [28]. Только в этом случае, ментальная карта будет действительно ментальной, будет представлять собой модель ментальной схемы, и будет обеспечивать более эффективное восприятие информации и более эффективное протекание процесса мышления.

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.

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 Методическое обеспечение для изучения основ алгоритмизации и программирования в базовом курсе информатики

2.1 Пример тематического планирования по основам алгоритмизации и программирования

На основе примерной программы и изученных учебно-методических комплектов Л.Л. Босовой, Н.Д. Угриновича и И.Г. Семакина было составлено следующее тематическое планирование (См. Таблица 2):

Таблица 2 – Тематическое планирование содержательной линии
«Алгоритмизация и программирование»

№	Тема	Часы
	Основы алгоритмизации	
1	Алгоритмы и исполнители	2
2	Способы записи алгоритмов	2
3	Объекты алгоритмов	2
4	Основные алгоритмические конструкции	2
	Начала программирования	
5	Общие сведения о языке программирования Паскаль	3
6	Организация ввода и вывода данных	3
7	Программирование линейных алгоритмов	3
8	Программирование разветвляющихся алгоритмов	3
9	Программирование циклических алгоритмов	3
	ВСЕГО:	23

Тема 1. «Алгоритмы и исполнители»

Понятие алгоритма. Исполнитель алгоритма. Свойства алгоритма.

Тема 2. «Способы записи алгоритмов»

Словесные способы записи алгоритма. Блок-схемы. Алгоритмические языки

Тема 3 «Объекты алгоритмов»

Величины. Выражения. Команда присваивания. Табличные величины.

Тема 4 «Основные алгоритмические конструкции»

Следование. Ветвление. Повторение.

Тема 5 «Общие сведения о языке программирования Паскаль»

Алфавит и словарь языка. Типы данных, используемые в языке Паскаль. Структура программы на языке Паскаль. Оператор присваивания.

Тема 6 «Организация ввода и вывода данных»

Вывод данных. Первая программа на языке Паскаль. Ввод данных с клавиатуры.

Тема 7 «Программирование линейных алгоритмов»

Числовые типы данных. Целочисленный тип данных. Символьный и строковый типы данных. Логический тип данных.

Тема 8 «Программирование разветвляющихся алгоритмов»

Условный оператор. Составной оператор. Многообразие способов записи ветвлений.

Тема 9 «Программирование циклических алгоритмов»

Программирование циклов с заданным условием продолжения работы. Программирование циклов с заданным условием окончания работы. Программирование циклов с заданным числом повторений.

Таким образом, данное тематическое планирование почти идентично примерной программе и рассчитано примерно на 23 учебных часа. За основной язык программирования выбран Паскаль.

2.2 Разработка ментальных карт по содержательной линии алгоритмизация и программирование

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

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, которую вы можете увидеть на втором рисунке была разработана в качестве ведущего инструмента визуального создания ментальных схем, с простым и дружелюбным пользовательским интерфейсом [27] . К недостаткам этой программы стоит отнести ограниченность ее возможностей.

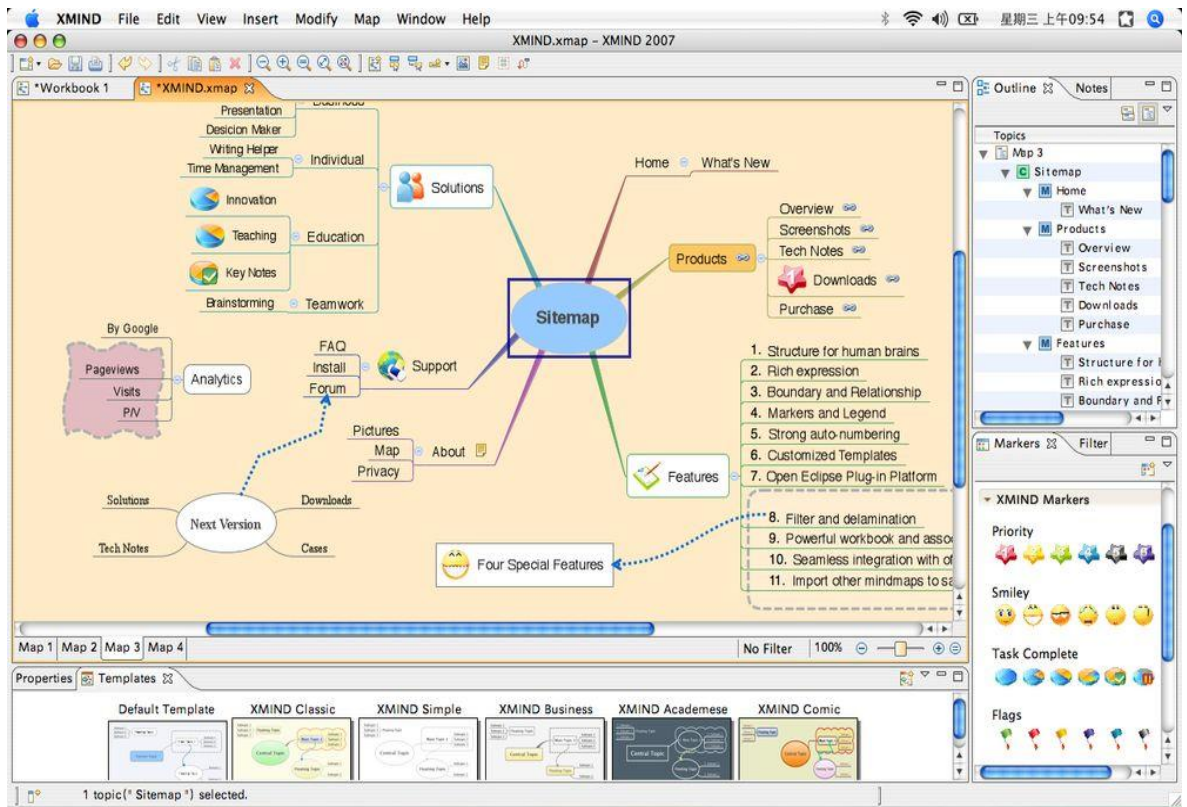


Рисунок – 2 Программа XMind

Программа Prezi, представленная на третьем рисунке, 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] (См. Рисунок 4).

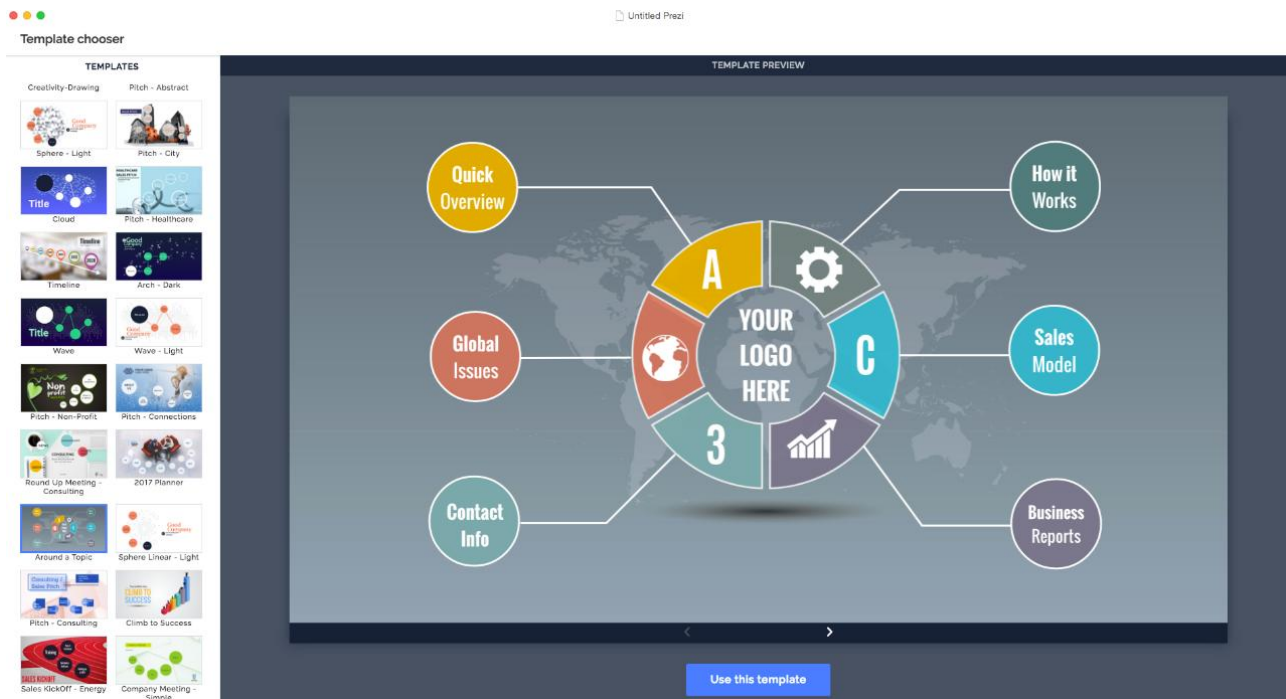


Рисунок – 3 Программа Prezi

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.

Для создания ментальной карты была выбрана программа Xmind, так как она проста в использовании и более других приспособлена для создания подобных карт или схем.

Было решено создать две карты, отдельно для раздела «Основы алгоритмизации», и так же отдельно для раздела «Начала программирования».

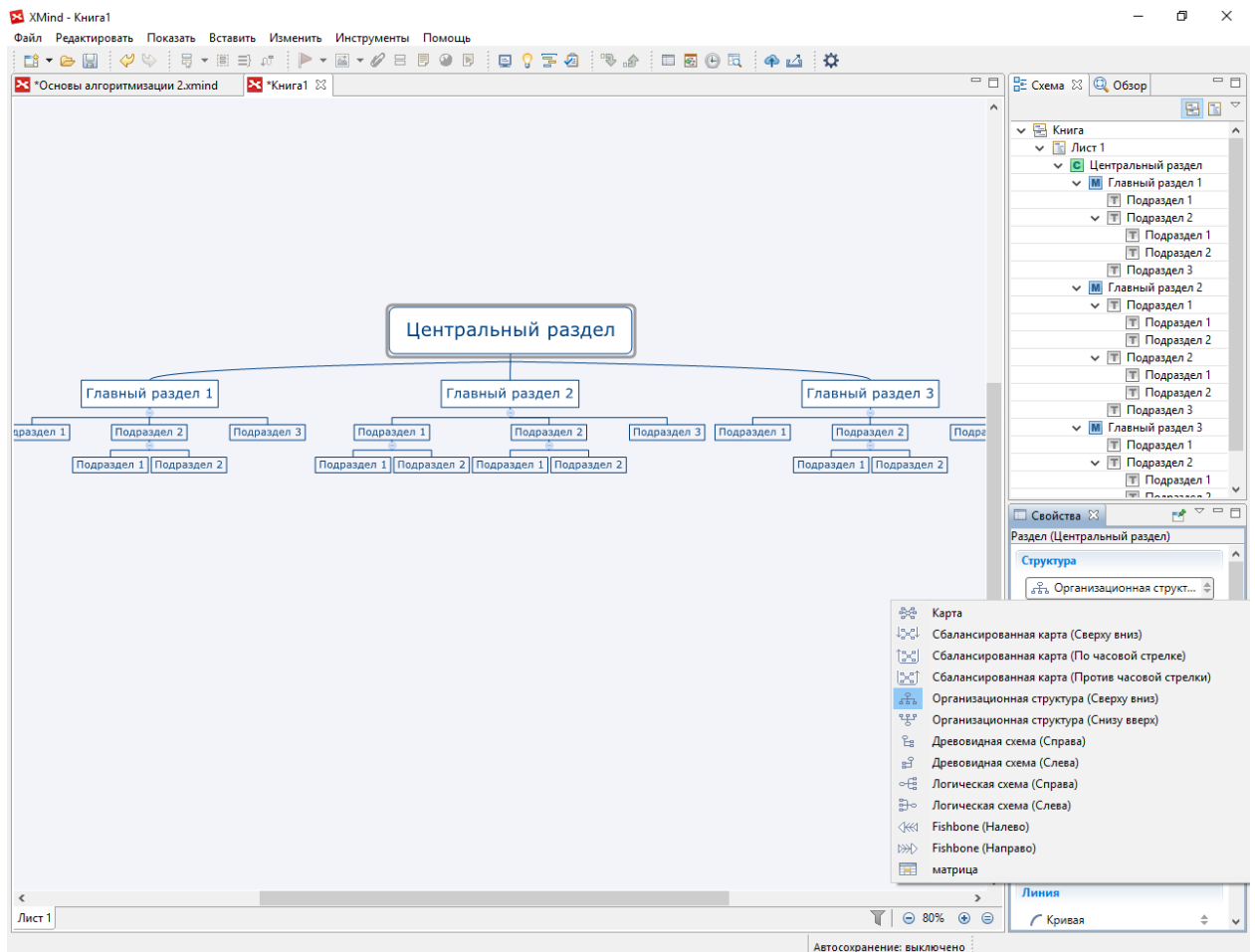


Рисунок – 4 Создание ментальной карты

Для начала выбираем нужную нам структуру ментальной карты, в данном случае мы выбрали организационную структуру с направлением вниз. Далее сверяясь с содержанием создаем нужное нам количество пунктов и подпунктов соответствующее количеству тем и подтем, таким образом у нас получится схема подобная той, что представлена на четвертом рисунке. В процессе готовые разделы и подразделы можно скрывать для упрощения работы с ментальной картой. Далее, в созданной схеме заполняем соответствующие разделы названиями тем и подтем и прочим учебным материалом на данные темы таким образом чтобы у нас получилось схематичное изображение учебного материала уроков.

Также, во время создания карты стоит обратить внимание на то что, помимо всего прочего в программе Xmind, как вы можете увидеть на пятом

рисунке, существует возможность прикреплять к разделам и подразделам различные объекты будь то рисунок, гиперссылка на веб ресурс, текстовая или аудиозаметка, а так же доступна возможность ссылаться на текстовые документы и презентации.

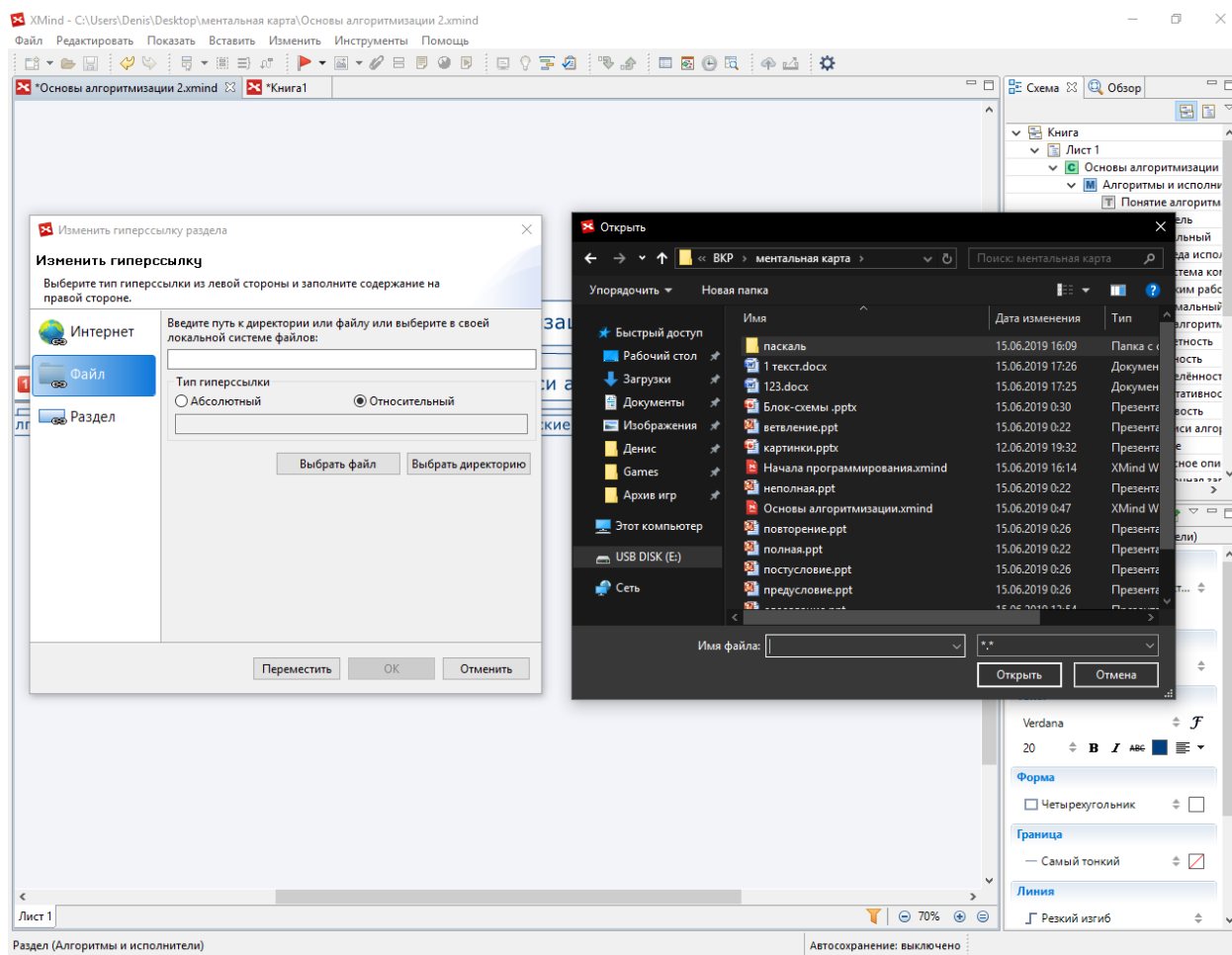


Рисунок – 5 Создание гиперссылок

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 пошагово изучать содержательную линию «Алгоритмизация и программирование» либо проверить степень уже усвоенных знаний.

2.3 Пример применения ментальных карт по основам алгоритмизации и программирования

В данной работе, продемонстрируем несколько способов использования ментальных карт на уроках.

Во-первых можно провести урок пользуясь готовой ментальной картой по теме. Продemonстрируем это на теме «Алгоритмы и исполнители».

Алгоритмы и исполнители

Планируемые образовательные результаты:

- *предметные* – понимание смысла понятия «алгоритм»; умение анализировать предлагаемые последовательности команд на предмет наличия у них таких свойств алгоритма как дискретность, детерминированность, понятность, результативность, массовость; понимание терминов «исполнитель», «формальный исполнитель», «среда исполнителя», «система команд исполнителя» и др.; умение исполнять алгоритм для формального исполнителя с заданной системой команд;

- *метапредметные* – понимание смысла понятия «алгоритм» и широты сферы его применения; понимание ограничений, накладываемых средой исполнителя и системой команд на круг задач, решаемых исполнителем;

- *личностные* – алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе.

Решаемые учебные задачи:

1) обобщение первоначальных представлений учащихся о понятиях «алгоритм» и «исполнитель»;

2) рассмотрение свойств алгоритма;

3) рассмотрение возможности автоматизации деятельности человека за счет возможности формального исполнения алгоритма.

Основные понятия, изучаемые на уроке:

- алгоритм;
- свойства алгоритма:
- дискретность;
- понятность;
- определенность;
- результативность;
- массовость;
- исполнитель;
- характеристики исполнителя:
- круг решаемых задач;
- среда;
- режим работы;
- система команд;
- формальное исполнение алгоритма.

Используемые на уроке средства ИКТ:

- персональный компьютер (ПК) учителя, мультимедийный проектор, экран;
- ПК учащихся.

Электронные образовательные ресурсы

- презентация «Алгоритмы и исполнители»;
- ресурсы федеральных образовательных порталов:
 - 1) демонстрация «Исполнители алгоритмов»;
 - 2) демонстрация «Происхождение и определение понятия алгоритма»;
 - 3) анимация «Решето Эратосфена».

Особенности изложения содержания темы урока

1. Организационный момент (1 минута)

Приветствие учащихся, сообщение темы и целей урока.

2. Изучение нового материала (25 минут)

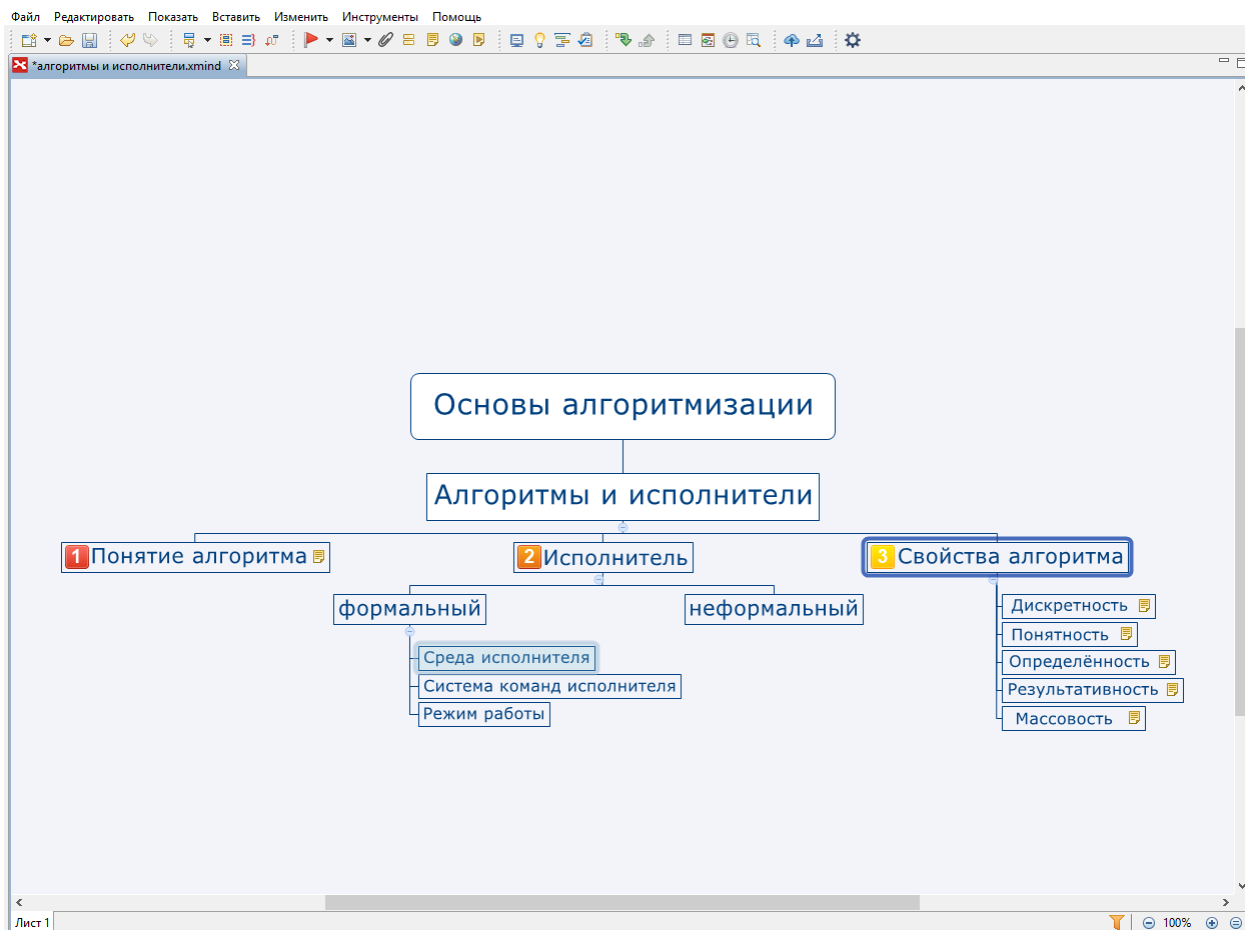
Новый материал излагается в сопровождении презентации «Алгоритмы и исполнители» и ментальной карты.

1 слайд – название презентации;

2 слайд – ключевые слова;

- алгоритм
- свойства алгоритма
- дискретность
- понятность
- определённость
- результативность
- массовость
- исполнитель
- характеристики исполнителя
- круг решаемых задач
- среда
- режим работы
- система команд
- формальное исполнение алгоритма

Учитель: Итак, прежде чем мы перейдем к изучению новой темы давайте взглянем на некую схему. Такие схемы называют ментальными картами. Данная карта отражает материал, который вы должны будете изучить на данном уроке. Внимательно просмотрите все разделы.



Давайте приступим к изучению и начнем с первого раздела, он называется «Понятие алгоритма». Итак, давайте разберемся что же такое алгоритм.

Каждый человек в повседневной жизни, в учёбе или на работе решает огромное количество задач самой разной сложности. Сложные задачи требуют длительных размышлений для нахождения решения; простые и привычные задачи человек решает не задумываясь, автоматически. В большинстве случаев решение каждой задачи можно разбить на простые этапы (шаги). Для многих таких задач (установка программного обеспечения, сборка шкафа, создание сайта, эксплуатация технического устройства, покупка авиабилета через Интернет и т. д.) уже разработаны и предлагаются пошаговые инструкции, при последовательном выполнении которых можно прийти к желаемому результату.

3 слайд – примеры алгоритмов;

Оплата через терминал.

4 слайд – примеры алгоритмов;

Рисование лошади.

5 слайд – примеры алгоритмов;

Вычислительный алгоритм.

Общая схема работы алгоритма.

6 слайд – происхождение и определение понятия алгоритма;

Можно сказать, что алгоритм – это описание последовательности шагов в решении задачи, приводящих от исходных данных к требуемому результату.

Просмотр и обсуждение демонстрации «Происхождение и определение понятия алгоритма».

Учитель: Хорошо, с понятием алгоритма разобрались, давайте снова обратимся к карте. Взгляните на следующий раздел, он называется «Исполнитель», рассмотрите подразделы. Итак, давайте разберемся что такое исполнители и в чем разница между формальными и неформальными.

7 слайд – исполнитель алгоритма (*схема*);

Исполнитель - это некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.

Круг решаемых задач. Каждый исполнитель создаётся для решения некоторого круга задач – построения цепочек символов, выполнения вычислений, построения рисунков на плоскости т. д.

Среда исполнителя. Область, обстановку, условия, в которых действует исполнитель, принято называть средой данного исполнителя. Исходные данные и результаты любого алгоритма всегда принадлежат среде того исполнителя, для которого предназначен алгоритм.

Система команд исполнителя. Предписание исполнителю о выполнении отдельного законченного действия называется командой. Совокупность всех команд, которые могут быть выполнены некоторым исполнителем, образует систему команд данного исполнителя (СКИ). Алгоритм составляется с учётом возможностей конкретного исполнителя, иначе говоря, в системе команд исполнителя, который будет его выполнять.

Режимы работы исполнителя. Для большинства исполнителей предусмотрены режимы непосредственного управления и программного управления. В первом случае исполнитель ожидает команд от человека и каждую поступившую команду немедленно выполняет. Во втором случае исполнителю сначала задаётся полная последовательность команд (программа), а затем он выполняет все эти команды в автоматическом режиме. Ряд исполнителей работает только в одном из названных режимов.

Просмотр и обсуждение демонстрации «Исполнители алгоритмов».

8 слайд – исполнитель Робот;

9 слайд – исполнитель Кузнечик;

10 слайд – разработка алгоритма (*схема*);

При разработке алгоритма:

- 1) выделяются фигурирующие в задаче объекты, устанавливаются свойства объектов, отношения между объектами и возможные действия с объектами;
- 2) определяются исходные данные и требуемый результат;
- 3) определяется последовательность действий исполнителя, обеспечивающая переход от исходных данных к результату;
- 4) последовательность действий записывается с помощью команд, входящих в систему команд исполнителя.

Алгоритм – модель деятельности исполнителя алгоритмов

Учитель: Отлично, мы разобрались что же такое исполнитель и какие они бывают, давайте вновь вернемся к карте. Два раздела мы уже прошли остался последний на сегодня, и он называется «Свойства алгоритма». Давайте посмотрим какими свойствами обладает алгоритм.

11 слайд – свойства алгоритма (*схема*);

Не любая инструкция, последовательность предписаний или план действий может считаться алгоритмом. Каждый алгоритм обязательно обладает следующими свойствами: дискретность, понятность, определённость, результативность и массовость.

12 слайд – дискретность;

Дискретность (от лат. discretus – разделенный, прерывистый) указывает, что любой алгоритм должен состоять из конкретных действий, следующих в определенном порядке. Образованная структура алгоритма оказывается дискретной: только выполнив одну команду, исполнитель сможет приступить к выполнению следующей.

13 слайд – понятность;

Понятность означает, что алгоритм состоит только из команд, входящих в систему команд исполнителя, т. е. из таких команд, которые исполнитель может воспринять и по которым может выполнить требуемые действия.

14 слайд – определенность;

Определённость означает, что в алгоритме нет команд, смысл которых может быть истолкован исполнителем неоднозначно; недопустимы ситуации, когда после выполнения очередной команды исполнителю неясно, какую команду выполнять на следующем шаге.

15 слайд – результативность;

Результативность означает, что алгоритм должен обеспечивать возможность получения результата после конечного, возможно, очень большого, числа шагов. При этом результатом считается не только обусловленный постановкой задачи ответ, но и вывод о невозможности продолжения по какой-либо причине решения данной задачи.

16 слайд – массовость;

Массовость означает, что алгоритм должен обеспечивать возможность его применения для решения любой задачи из некоторого класса задач с различными исходными данными.

17 слайд – решето Эратосфена;

Просмотр и обсуждение анимации «Решето Эратосфена».

Рассмотренная последовательность действий является алгоритмом, так как она удовлетворяет свойствам:

- дискретности - процесс нахождения простых чисел разбит на шаги;

- понятности - каждая команда понятна ученику 9 класса, выполняющему этот алгоритм;
- определённости - каждая команда трактуется и выполняется исполнителем однозначно; имеются указания об очерёдности выполнения команд;
- результативности - через некоторое число шагов достигается результат;
- массовости - последовательность действий применима для любого натурального n .

18 слайд – алгоритм (*точное определение*);

Алгоритм - это предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами:

- дискретности
- понятности
- определённости
- результативности
- массовости

19 слайд – возможности автоматизации деятельности человека (*схема*);

Разработка алгоритма – как правило, трудоёмкая задача, требующая от человека глубоких знаний, изобретательности и больших временных затрат.

Решение задачи по готовому алгоритму требует от исполнителя только строгого следования заданным предписаниям.

20 слайд – автоматизация деятельности человека;

21 слайд – самое главное.

- Исполнитель - некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд. Формальный исполнитель одну и ту же команду всегда выполняет одинаково. Для каждого формального исполнителя можно указать: круг решаемых задач, среду, систему команд и режим работы.
- Алгоритм - предназначенное для конкретного исполнителя описание

последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами дискретности, понятности, определённости, результативности и массовости.

- Способность исполнителя действовать формально обеспечивает возможность автоматизации деятельности человека.

Вопросы и задания

22 слайд – вопросы и задания;

3. Подведение итогов урока. Сообщение домашнего задания. Выставление оценок (4 минуты)

Учитель: Вот и подходит к концу наш урок. Давайте снова взглянем на карту и посмотрим все ли разделы и подразделы мы сегодня рассмотрели и рассмотрим те из них которые вызывают у вас трудности.

23 слайд – опорный конспект;

24 слайд – Д/з.

Также, можно дать ученикам самим составить ментальную карту после изучения урока и таким образом проверить степень усвоенного материала. Продемонстрируем это на теме «Способы записи алгоритмов».

Способы записи алгоритмов

Планируемые образовательные результаты:

- *предметные* – знание различных способов записи алгоритмов;
- *метапредметные* – умение анализировать предлагаемые последовательности команд на предмет наличия у них таких свойств алгоритма как дискретность, детерминированность, понятность, результативность, массовость; понимание преимущества и недостатков той или иной формы записи алгоритмов; умение переходить от одной формы записи алгоритмов к другой; умение выбирать форму записи алгоритма, соответствующую решаемой задаче;
- *личностные* – алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе.

Решаемые учебные задачи:

- 1) закрепление понятий «алгоритм», «исполнитель», «свойства алгоритма»;
- 2) рассмотрение словесных способов записи алгоритмов;
- 3) рассмотрение графического способа записи алгоритмов с помощью блок-схем;
- 4) формирование общих представлений об алгоритмических языках.

Основные понятия, изучаемые на уроке:

- алгоритм;
- словесное описание;
- построчная запись;
- блок-схема;
- школьный алгоритмический язык.

Используемые на уроке средства ИКТ:

- персональный компьютер (ПК) учителя, мультимедийный проектор, экран;
- ПК учащихся.

Электронные образовательные ресурсы

- презентация «Способы записи алгоритмов»;

Особенности изложения содержания темы урока

1. Организационный момент (1 минута)

Приветствие учащихся, сообщение темы и целей урока.

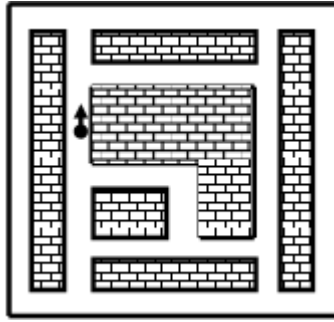
2. Повторение (5 минут)

3. Изучение нового материала (20 минут)

Обсудить следующие ситуации:

Задание 1 (базовый уровень)

Человек находится в лабиринте и начинает двигаться в направлении, указанном стрелкой, согласно следующему предписанию: идти шаг за шагом, не отрывая руки от правой стены; шагать, пока не выйдешь из лабиринта.



Почему данное предписание нельзя считать алгоритмом выхода из лабиринта? Какое свойство алгоритма здесь нарушено?

- 1) Детерминированность
- 2) Понятность
- 3) Результативность
- 4) Массовость

Ответ: 3.

Задание 2 (базовый уровень)

Исполнитель Вычислитель выполняет следующую последовательность действий:

- 1) переменной s присвоить значение 0;
- 2) переменной i присвоить значение 1;
- 3) если $i > 10$, то перейти к выполнению п. 6; иначе выполнить п. 4;
- 4) увеличить значение переменной s на i ;
- 5) увеличить значение переменной i на 1;
- 6) сообщить значение s .

Каким свойством не обладает данная последовательность действий?

- 1) Дискретность
- 2) Детерминированность
- 3) Результативность
- 4) Массовость

Ответ: 4.

Задание 3 (базовый уровень)

Исполнитель Вычислитель работает с числами, записанными на доске. Он может:

- подсчитывать количество чисел на доске;
- сравнивать количество имеющихся на доске чисел с 1;
- стирать с доски два любых числа, находить их сумму, увеличивать эту сумму на 1 и записать на доске полученный результат.

Пока на доске более одного числа, Вычислитель стирает любые два из них и вместо них записывает их сумму, увеличенную на единицу. Каким свойством не обладает последовательность действий, совершаемая исполнителем?

- 1) Детерминированность
- 2) Понятность
- 3) Результативность
- 4) Массовость

Ответ: 1.

Новый материал излагается в сопровождении презентации «Способы записи алгоритмов».

1 слайд – название презентации;

2 слайд – ключевые слова;

- алгоритм
- словесное описание
- построчная запись
- блок-схема
- школьный алгоритмический язык

Существуют различные способы записи алгоритмов. Основными среди них являются:

- словесные;
- графические;
- на алгоритмических языках.

3 слайд – Марков А.А.;

Теоретические исследования нашего соотечественника Андрея Андреевича Маркова (младшего) (1903–1979), выполненные в середине прошлого века, показали, что в общем случае алгоритмы должны содержать предписания двух видов:

- 1) предписания, направленные на непосредственное преобразование информации (функциональные операторы);
- 2) предписания, определяющие дальнейшее направление действий (логические операторы).

Именно эти операторы положены в основу большинства способов записи алгоритмов.

4 слайд – основные способы записи алгоритма (*схема*);

5 слайд – словесное описание;

Словесное описание - самая простая запись алгоритма в виде набора высказываний на обычном разговорном языке.

Пример. Словесное описание алгоритма нахождения наибольшего общего делителя (НОД) пары целых чисел (алгоритм Евклида).

Чтобы найти НОД двух чисел, составьте таблицу из двух столбцов и назовите столбцы X и Y. Запишите первое из заданных чисел в столбец X, а второе - в столбец Y. Если данные числа не равны, замените большее из них на результат вычитания из большего числа меньшего.

Повторяйте такие замены до тех пор, пока числа не окажутся равными, после чего число из столбца X считайте искомым результатом.

6 слайд – построчная запись (*схема*);

Это запись на естественном языке, но с соблюдением некоторых дополнительных правил:

- каждое предписание записывается с новой строки;
- предписания (шаги) алгоритма нумеруются;
- исполнение алгоритма происходит в порядке возрастания номеров шагов, начиная с первого (если не встречается никаких специальных указаний).

Кроме слов естественного языка предписания могут содержать математические выражения и формулы.

7 слайд – построчная запись алгоритма Евклида;

1. Начало.
2. Обозначить первое из заданных чисел X , второе - Y .
3. Если $X = Y$, то перейти к п. 9.
4. Если $X > Y$, то перейти к п. 5, иначе перейти к п. 7.
5. Заменить X на $X - Y$.
6. Перейти к п. 3.
7. Заменить Y на $Y - X$.
8. Перейти к п. 3.
9. Считать X искомым результатом.
10. Конец.

8 слайд – графические способы;

Последовательные картинки

Структурограмма

9 слайд – блок-схемы;

Блок-схема представляет собой графический документ, дающий представление о порядке работы алгоритма.

В блок-схеме предписания изображаются с помощью различных геометрических фигур, а последовательность выполнения шагов указывается с помощью линий.

Рассмотреть условные обозначения, применяемые в блок-схемах.

10 слайд – запись алгоритма Евклида с помощью блок-схемы;

Создание детальной блок-схемы сложного алгоритма – трудоёмкая задача. Кроме того, блок-схема, не уместяющаяся на одном стандартном листе, теряет своё основное преимущество – наглядность. При разработке сложных алгоритмов блок-схемы удобно использовать в качестве средства для наглядного представлений решения задачи в общем виде.

11 слайд – алгоритмические языки (*схема*);

Алгоритмические языки – формальные языки, предназначенные для записи алгоритмов. Каждый из них характеризуется:

- алфавитом – набором используемых символов;
- синтаксисом – системой правил, по которым из символов алфавита образуются правильные конструкции языка;
- семантикой – системой правил, строго определяющей смысл и способ употребления конструкций языка.

Класс алгоритмических языков очень широк. При изучении курса информатики в школах используются различные версии школьного (учебного) алгоритмического языка.

12 слайд – общий вид программы на школьном алгоритмическом языке;

Для записи алгоритмов на школьном алгоритмическом языке используется некоторое ограниченное число слов, смысл и способ употребления которых заданы раз и навсегда. Это так называемые служебные слова: **алг** (алгоритм), **дано**, **надо**, **нач** (начало), **кон** (конец), **арг** (аргумент), **рез** (результат) и др. При записи алгоритмов в книгах служебные слова выделяются жирным шрифтом, в тетради и на доске – подчёркиванием.

В общем виде программу на школьном алгоритмическом языке можно представить так:

алг <название алгоритма>

нач

<последовательность команд>

кон

13 слайд – алгоритм для исполнителя Водолей;

Алгоритм, позволяющий из полного сосуда ёмкостью 12 л отлить половину, пользуясь двумя пустыми сосудами ёмкостью 8 и 5 л.

алг переливания

нач

наполнить сосуд ёмкостью 8 л из сосуда ёмкостью 12 л

наполнить сосуд ёмкостью 5 л из сосуда ёмкостью 8 л

вылить всё из сосуда ёмкостью 5 л в сосуд ёмкостью 12 л
вылить всё из сосуда ёмкостью 8 л в сосуд ёмкостью 5 л
наполнить сосуд ёмкостью 8 л из сосуда ёмкостью 12 л
долить из сосуда ёмкостью 8 л сосуд ёмкостью 5 л
вылить всё из сосуда ёмкостью 5 л в сосуд ёмкостью 12 л
конец

14 слайд – самое главное.

Существуют различные способы записи алгоритмов:

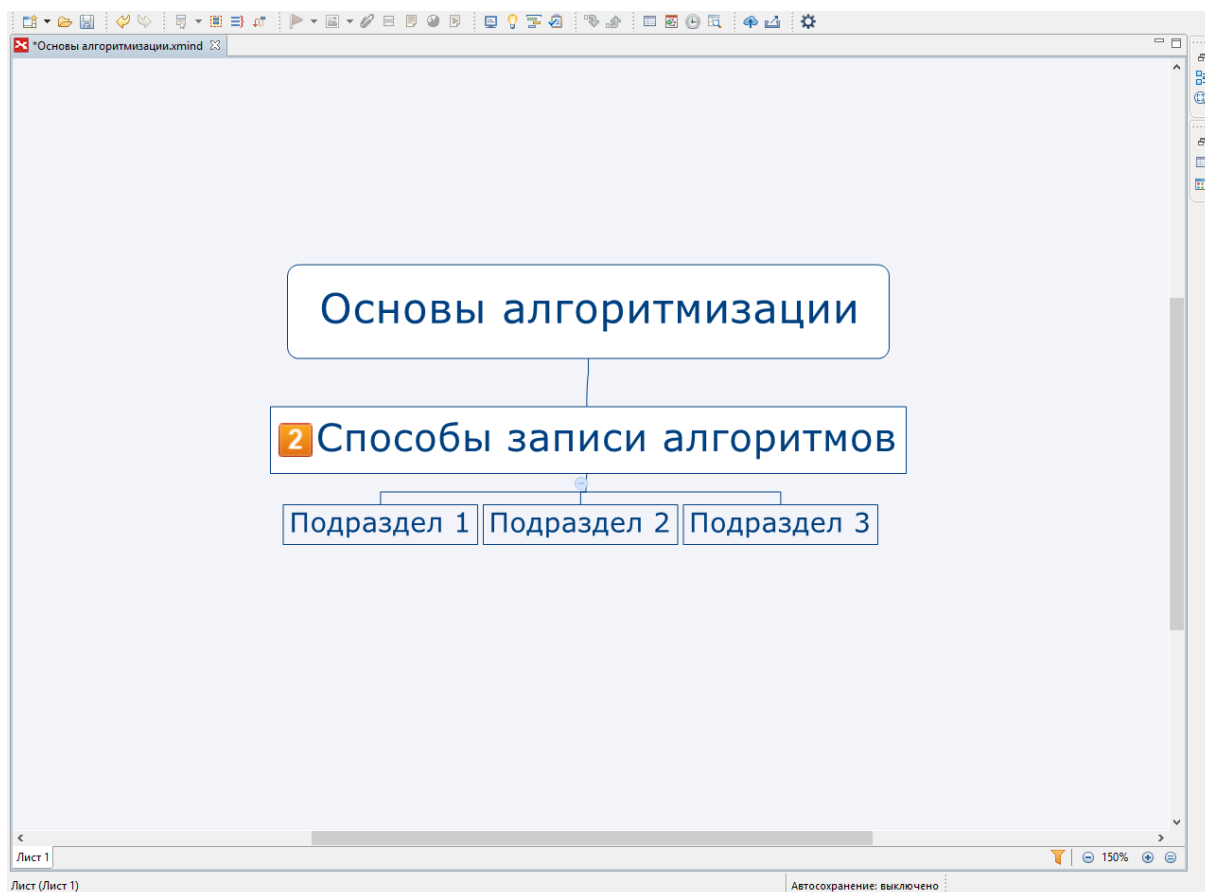
- словесное описание
- построчная запись
- блок-схема
- школьный алгоритмический язык и другие.

Каждый из этих способов обладает своими достоинствами и недостатками.

Вопросы и задания

15 слайд – вопросы и задания;

4. Практическая часть. Решение задач (15 минут)



Итак, ребята, помните ментальную карту которой мы пользовались на прошлом уроке? Сейчас вам предстоит самим сделать подобную карту. У вас на компьютерах есть заготовки. Используя ваши тетради и вам нужно дополнить заготовку и сделать из нее ментальную карту на тему сегодняшнего урока, вписать туда все что вы сегодня изучили и структурировать.

5. Подведение итогов урока. Сообщение домашнего задания. Выставление оценок (4 минуты)

17 слайд – опорный конспект;

18 слайд – Д/з.

Таким образом, ментальные карты можно использовать различными способами в учебном процессе, они представляют собой идеальное решение для проверки знаний учащихся и помощника при планировании, выполнении и осуществлении контроля усвоенных знаний учащимися.

ЗАКЛЮЧЕНИЕ

Цель нашего исследования заключалась в разработке методического обеспечения для изучения содержательной линии «Алгоритмизация и программирование».

В ходе исследования мы решили следующие задачи:

1) Рассмотрели обучение основам алгоритмизации и программирования в школе и существующие проблемы. Федеральный базисный учебный план для образовательных учреждений РФ отводит 105 часов для обязательного изучения информатики и информационных технологий на ступени основного общего образования и 70 часов на ступени полного общего образования. На алгоритмизацию и программирование отводится около 19 часов. В случае профильного изучения информатики ситуация значительно улучшается. Федеральный базисный учебный план отводит 280 часов для обязательного изучения информатики и информационных технологий на ступени среднего общего образования, то есть по 140 учебных часов на каждый год обучения. Как мы видим, на изучение раздела программирования отводится недостаточное количество времени, это приводит к тому, что изучение некоторых тем проходит поверхностно, а некоторые исключаются вовсе[12].

Перед началом обучения учитель сталкивается с проблемой выбора языка программирования для изучения. Мнения о том, какой язык программирования лучше преподавать в школе, разнятся: от того, что программирование изучать не нужно, а следует просто поднимать компьютерную грамотность и осваивать офисные программы (как Западе), до того, что нужно изучать операционные системы и несколько языков программирования с различными парадигмами. Это крайние случаи, но золотую середину найти непросто. Некоторые считают, что не так важно, какой язык программирования взять: на уроках информатики нужно учить не языку программирования, а методам программирования и системному подходу решения задач. Нужно развивать алгоритмическое

мышление и на примерах знакомиться с принципами построения современных компьютерных систем.

У каждого преподавателя есть свой список требований к учебному языку программирования. Набор требований может быть, например, таким: простой, интуитивный синтаксис, наличие высокоуровневых инструментов для обнаружения и недопущения ошибок и для отладки программ, наличие качественной документации с примерами, наличие дружелюбной среды разработки, кроссплатформенность (наличие версий под различные платформы).

В последнее время наметилась явная тенденция перехода обучения от алгоритмических к объектно-ориентированным языкам с использованием визуальных систем программирования (Delphi, Visual Basic, Visual C++ и т. д.).

2) Проанализировали содержание линии алгоритмизации и программирования в школьном курсе информатики. В примерной программе теме «Алгоритмы и исполнители» выделяется 19 часов. Содержание темы расписано более подробно, чем в стандарте: Алгоритм. Свойства алгоритма. Способы записи алгоритмов; блок-схемы. Возможность автоматизации деятельности человека. Исполнители алгоритмов (назначение, среда, режим работы, система команд). Компьютер как формальный исполнитель алгоритмов (программ). Алгоритмические конструкции: следование, ветвление, повторение. Разбиение задачи на подзадачи, вспомогательный алгоритм. Алгоритмы работы с величинами: типы данных, ввод и вывод данных».

В этом же разделе говорится о программировании: «Языки программирования, их классификация. Правила представления данных. Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла. Правила записи программы. Этапы разработки программы: алгоритмизация – кодирование – отладка – тестирование».

В учебнике Л.Л. Босовой раздел относящийся к Алгоритмизации по содержанию почти совпадает с первой частью вышеописанной примерной программы, раздел программирования полностью относится только к языку

программирования Паскаль. Н.Д. Угринович в своем учебнике совмещает алгоритмизацию и программирование в одну главу, но ее содержание сильно отличается от ранее упомянутой примерной программы. Большая часть учебника И.Г. Семакина посвящена алгоритмизации и программированию.

В подходах к определению алгоритма в разных учебниках имеются заметные различия. В учебнике Л.Л. Босовой при раскрытии понятия алгоритма рассматриваются этапы решения задач на компьютере, алгоритм представляется как один из шести этапов. Алгоритм это четкая инструкция, задающая необходимую последовательность действий для решения задачи.

В учебнике И.Г. Семакина при раскрытии понятия алгоритма рассматривается кибернетическая модель системы управления, в которой управляющий объект и объект управления связаны между собой каналами прямой и обратной связи. Алгоритм есть последовательность команд, которая передается от управляющего через канал прямой связи к объекту управления. Исполнителем алгоритма является объект управления, поскольку именно он осуществляет действия, предписанные командами алгоритма.

В учебнике Н.Д. Угриновича сказано: «Алгоритмы широко используются в технике в системах управления объектами. В любой системе управления существует управляющий объект, который является исполнителем алгоритма управления». Утверждение противоположно тому, что дается в [5]. Приводится пример управления обогревателем воздуха со стороны человека, наблюдающего за температурой в помещении. Делается утверждение, что исполнителем алгоритма является человек.

3) Ментальный подход к обучению предполагает смещение целеполагания учебного процесса в сторону развития когнитивных способностей обучаемых, в частности, если рассматривать обучение программированию – то основной целью будет являться развитие алгоритмического стиля мышления.

Алгоритмический способ мышления позволяет принимать оптимальные решения в любой сфере человеческой деятельности, и сам по себе никак не

взаимодействует с программированием и вычислительной техникой. В таком некоем виде он существовал постоянно, то есть изначально принадлежал человеческому мышлению. Благодаря появлению вычислительной техники и профессии программиста привело к тому, что определенный круг специалистов нуждается в необходимости алгоритмического способа мышления. Алгоритмический способ для выполнения программистских задач является единственно возможным.

Общий алгоритмический подход в программистской практике углубляется и детализируется: структура предметной области становится формализованной информационной структурой, в ней вычленяются количественные взаимосвязи, образующие математическую модель предметной области, превращение алгоритма в компьютерную программу. Т. е. у людей программистов, алгоритмическое мышление развито на более высоком, профессиональном уровне. Для того, чтобы сформировать алгоритмическое мышление у школьников, на повседневном уровне, у самого учителя оно должно быть развито на профессиональном уровне. Кроме того, современные подходы к совершенствованию системы среднего образования предъявляют высокие требования к уровню предметной подготовки учителя. Несомненно, его педагогический кругозор должен намного превышать рамки школьной программы. Следовательно, учитель информатики должен владеть всеми современными технологиями программирования.

4) Разработали ментальные карты для изучения содержательной линии «Алгоритмизация и программирование».

Таким образом, все задачи решены, цель работы достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лапчик, М.П. Теория и методика обучения информатике : учебник / М.П. Лапчик. – Москва : Издательский центр «Академия», 2008. – 592с.
2. Босова, Л.Л. Информатика. Учебник для 8 класса. / Л.Л. Босова, А.Ю. Босова. – 2-е изд., испр. – Москва : БИНОМ. Лаборатория знаний, 2014. – 160 с.
3. Босова, Л.Л. Информатика. 9 класс. / Л.Л. Босова. – Москва : БИНОМ. Лаборатория знаний, 2017. – 208 с.
4. Угринович, Н.Д. Информатика. Учебник для 9 класса. / Н.Д. Угринович. – 4-е изд. – Москва : БИНОМ. Лаборатория знаний, 2016. – 152 с.
5. Семакин, И.Г. Информатика. 9 класс. / И.Г. Семакин. – 3-е изд. – Москва : БИНОМ. Лаборатория знаний, 2015. – 200 с.
6. Акулов, О. А. Информатика: базовый курс : Учебник для студентов вузов, бакалавров, магистров, обучающихся по направлениям «Информатика и вычислительная техника» / О. А. Акулов, Н. В. Медведев. – Москва : Омега-Л, 2004. – 552с.
7. Греков, А.В. Структуры и алгоритмы обработки данных / А. В. Греков, С. П. Кириллов, А. Б. Суворов. – Учебное пособие Острогжск, 2016 – 172с.
8. Ершова, А.П. Основы информатики вычислительной техники. В 2 ч. : пробное учеб. пособие для сред. учеб. заведений / А.П. Ершова, В.М. Монахова. – Москва : Просвещение, 1985 – 192 с.
9. Босова, Л.Л. Информатика. 7–9 классы: методическое пособие / Л.Л. Босова, А.Ю. Босова. – Москва : БИНОМ. Лаборатория знаний, 2016. – 464 с.
10. Деменченко, О.Г. Информатика. Учебное пособие / О. Г. Деменченко, В. И. Демаков, А. С. Косянчук. – Иркутск, 2015 – 111с.

11. Могилев, А. В. Информатика: Учеб. Пособие для студ. пед. вузов / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. 2-е изд. – Москва : Изд. Центр «Академия», 2004. – 848 с.
12. Симонович, С. В. Информатика: Базовый курс. / С. В. Симонович, 3-е изд. Санкт-Петербург : Питер, 2014. – 640с.
13. Угринович, Н. Д. Информатика и ИКТ. Профильный уровень: учебник для 11 класса / Н. Д. Угринович. – 2-е изд., испр. И доп. – Москва : БИНОМ. Лаборатория знаний, 2016 – 308 с.
14. Бородин, М.Н. Информатика. Программы для образовательных учреждений. 2 – 11 классы. / М.Н. Бородин. – Москва : БИНОМ. Лаборатория знаний, 2015 – 576 с.
15. Хлебников, А.А. Информационные технологии: учебник / А.А. Хлебников. — Москва : КНОРУС, 2016. — 466 с.
16. Гейн, А.Г. Информатика. Классы 7 – 9 . / А.Г. Гейн, А.И. Сенокосов, В.Ф. Шелухович. – Москва : Дрофа, 2014 – 576 с.
17. Бобров, А. Н. Проблемы выбора языка программирования в школьном курсе информатики / А.Н. Бобров. – Молодой ученый, 2015 – 64с.
18. Ожегов, С.И. Толковый словарь русского языка: 80.000 слов и фразеологических выражений / С.И. Ожегов, Н.Ю. Шведова. – Российская АН; Российский фонд культуры. 3-е изд. Москва: АЗЪ, 1996.
19. Зимняя, М.И. Педагогическая психология: Учебник для вузов. / М.И. Зимняя. Изд.второе, доп., испр. и перераб. – Москва: Издательская корпорация «Логос», 1999. 384 с.
20. Найссер, У. Познание и реальность. / У. Найссер – Москва: Прогресс, 1981. – 252 с.
21. Joseph D. Novak, Cornell University. The Theory Underlying Concept Maps and How To Construct Them. [Электронный ресурс] – Режим доступа: https://web.stanford.edu/dept/SUSE/projects/ireport/articles/concept_maps/The%20Theory%20Underlying%20Concept%20Maps.pdf

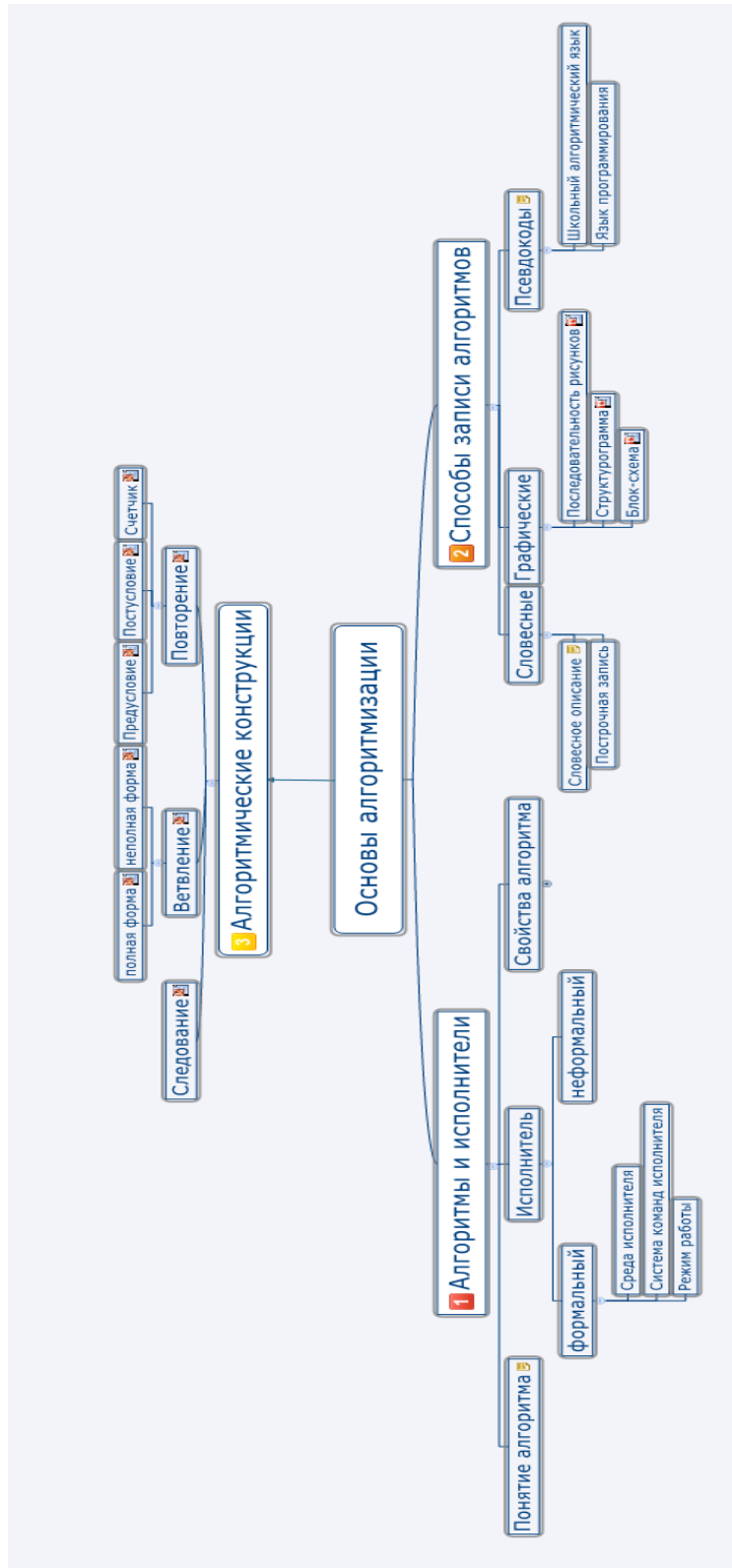
22. Гаврилова, Т.А. «Базы знаний интеллектуальных систем». / Т.А. Гаврилова, В.Ф. Хорошевский. – Питер, 2000
23. Богданова, Г.Н. Интеллект-карты как средство развития интеллектуальных способностей учащихся. [Электронный ресурс] / Г.Н. Богданова. – Режим доступа: <https://refdb.ru/look/2197746.html>
24. Голубцова, А.В. Разработка ментальных алгоритмических карт по теме «Основные алгоритмические структуры». / А.В. Голубцова, Е.Д. Грук, Т.А. Степанова. – Материалы II Всероссийской научно-практической конференции с международным участием «Перспективы и вызовы информационного общества», 2013
25. Жаркая, М.А. Изучаем тему «Понятие алгоритма» посредством создания карт понятий. [Электронный ресурс] / М.А. Жаркая. – Режим доступа: https://inf429.ucoz.ru/publ/statja_izuchaem_temu_ponjatie_algoritma_posredstvom_sozdanija_kart_ponjatij/1-1-0-1
26. Озерецковский, Н.В. Использование средств структурирования содержания в разработке учебных материалов для повышения квалификации в области ИКТ. [Электронный ресурс] / Н.В. Озерецковский, Н.А. Королев. – Режим доступа: <http://www.ict.edu.ru/vconf/files/9386.pdf>
27. Сидоров, С.В. Ментальные карты на лекции по педагогике. – Сайт педагога-исследователя. [Электронный ресурс] / С.В. Сидоров. Режим доступа: <http://sv-sidorov.ucoz.com>
28. Степанова, Т.А. Сущность алгоритмического мышления с позиций информационного подхода / Т.А. Степанова. – Красноярск 2012, №3. С. 95-100
29. Степанова, Т.А. Теория алгоритмического мышления: учебное пособие для магистрантов, учителей общеобразовательных учреждений, преподавателей вузов; / Т.А. Степанова. Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2014. – 72 с.

30. Афанасова, Л. В. Педагогический опыт по теме "Развитие алгоритмического мышления учащихся на уроках информатики" (районный банк данных, 2003 год) [Электронный ресурс] / Л.В. Афанасова. – Режим доступа: <http://afanasova-mila.narod.ru/images/apo2003.doc>
31. Xmind – Конструирование интеллектуальных карт. [Электронный ресурс] Режим доступа: <http://pro-spo.ru/-linux-e/1940-2010-02-17-10-45-32>
32. Угринович, Н.Д. Информатика и ИКТ: учебник для 9 класса / Н.Д. Угринович. – 6-е изд. – Москва: БИНОМ. Лаборатория знаний, 2012. – 295 с.
33. Угринович, Н.Д. Информатика. Программа для основной школы: 7 – 9 классы / Н.Д. Угринович, Н.Н. Самылкина. - Москва: БИНОМ. Лаборатория знаний, 2012. - 53с. - (Программы и планирование).
34. Семакин, И.Г. Информатика. Программа для старшей школы: 10-11 классы. Базовый уровень / И.Г. Семакин. - Москва: БИНОМ. Лаборатория знаний, 2015. - 100 с. - (Программы и планирование).
35. Босова, Л.Л. Информатика. Программа для основной школы: 5-6 классы. 7-9 классы / Л.Л. Босова, А.Ю. Босова. - 3 изд. - Москва: БИНОМ. Лаборатория знаний, 2015. - 88 с.: ил. - (Программы и планирование).
36. Пак, Н.И. Информационное моделирование. Учебное пособие. / Н.И. Пак. – Красноярск: РИО КГПУ, 2010. – 152 с
37. 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.
38. Беляев, М.В. Алгоритмическое мышление как цель современного образования. [Электронный ресурс] / М.В. Беляев. – Режим доступа: www.mbelyaev.chat.ru

39. Кнут, Д. Алгоритмическое мышление и математическое мышление [Электронный ресурс] / Д.Кнут. – Режим доступа: <http://www.kph.npu.edu.ua/e-book/klasik/data/math/knut.html>
40. Копаев, А.В. Алгоритм как модель алгоритмического процесса. [Электронный ресурс] / А.В.Копаев. Режим доступа: <http://www.rusedu.info/Article100.html>

ПРИЛОЖЕНИЕ А

Ментальная карта «Основы алгоритмизации»



ПРИЛОЖЕНИЕ Б

Ментальная карта «Начала программирования»

