

Тема 9. Представление об алгоритмизации и программировании

Цель изучения: развитие и углубление представления об алгоритме и алгоритмизации, о программе и программировании.

Ключевые слова: алгоритм, алгоритмизация, блок-схема алгоритма, ветвление, метаязык, программа, программирование, следование, среда программирования, типовые конструкции алгоритма, формы представления алгоритма, цикл, язык программирования, язык программирования BASIC, язык программирования Pascal.

Методическое обеспечение: [1, 7]: глава 8, [3, 4].

Количество часов:

Класс	Базовый вариант		Расширенный вариант	
	Урочная деятельность, ч	Самостоятельная деятельность, ч (рекомендация)	Урочная деятельность, ч	Самостоятельная деятельность, ч (рекомендация)
10	2	2	3	2
11	0	0	1	2
Итого:	2	2	4	4

В результате изучения темы учащиеся

должны знать:

- понятие алгоритма;
- типы алгоритмических конструкций: следование, ветвление, цикл;
- способы описания алгоритмов: блок-схему, словесный и др.;
- понятие программы;
- состав среды программирования;
- понятие языка программирования;
- понятие алфавита, синтаксиса и семантики языка программирования;
- понятие метаязыка;
- основные этапы технологии работы в среде программирования;

должны уметь:

- формально исполнять готовые алгоритмы для конкретного исполнителя с фиксированным набором команд;
- определять результат выполнения готового алгоритма;
- составлять алгоритм для формального исполнителя, используя базовые алгоритмические конструкции: следование, ветвление, цикл;
- выполнять набор, редактирование и сохранение программы в среде программирования;

получат возможность:

- познакомиться с возможностями метаязыка;
- оценить универсальность структурной алгоритмизации для различных исполнителей, для различных данных;
- познакомиться с широким диапазоном версий изучаемых языков программирования и задуматься над этапами их развития.

Место темы в непрерывном курсе информатики в 7–11 классах

Изучение темы «Представление об алгоритмизации и программировании» базируется на предметных знаниях, полученных в курсе информатики основной школы при изучении содержательной линии «Алгоритмы и элементы программирования» (ПООП ООО [10]):

- понятий алгоритма и форм его представления;
- понятий базовых алгоритмических конструкций;
- интерфейса выбранного языка программирования,

а также на умениях описания алгоритмов с помощью типовых алгоритмических конструкций в различных формах.

Особенности жизни современного человека, деятельность которого подчинена алгоритмам, а также задачи развития мышления учащихся определяют особую роль алгоритмических знаний в фундаментальном образовании. Поэтому изучение алгоритмизации должно быть продолжено в 10–11 классах и подчинено, во-первых, задаче развития алгоритмического мышления, во-вторых, задаче развития навыков программирования.

В результате обучения на базовом уровне должны быть достигнуты следующие предметные результаты освоения образовательной программы в соответствии с требованиями ФГОС [9]: владение «навыками алгоритмического мышления и понимание необходимости формального описания алгорит-

мов», «умением анализировать алгоритмы с использованием таблиц». Так как перечень предметных результатов освоения образовательной программы является инвариантным по отношению к ступени обучения, обучение в средней школе должно обеспечивать углубление знаний, полученных на ступени основной школы.

Ожидаемые результаты обучения:

- личностные: Л-4, Л-5, Л-6, Л-7, Л-9;
- метапредметные: М-1, М-3, М-4;
- предметные О-2, О-3, И-2.

Педагогические технологии, используемые при обучении теме:

- классно-урочная технология обучения;
- диалогическая технология;
- технология организации самостоятельной деятельности;
- технология организации проектной деятельности.

Рекомендации по раскрытию содержания темы

Изучение темы опирается на базовые понятия: алгоритм, свойства алгоритма, формы представления алгоритма, базовые алгоритмические конструкции, которые изучены на ступени основной школы и представлены в материалах учебника [2, 7].

Параграф 8.1 «Понятие алгоритма. Свойства, формы представления и типовые конструкции алгоритма» включает описание базовых понятий алгоритмизации, вопросы и задания для контроля и определения результата алгоритма, составления алгоритма.

Изучение темы следует начать с самостоятельной деятельности учащихся:

- 1) с учебником [2, 7];
- 2) с интернет-ресурсами по теме «Алгоритм. Формы описания и свойства алгоритма. Базовые алгоритмические конструкции» из Единой коллекции цифровых образовательных ресурсов;
- 3) по подготовке устных ответов на вопросы и задания 8.1–8.8.

Учащиеся готовы к выделенному объёму самостоятельной работы, так как эта деятельность предполагает повторение изученного материала. Таким образом, данный этап работы

может внести вклад в решение задачи мотивирования на самообразование в течение всей своей жизни, которая сформулирована во ФГОС [9] при описании личностных характеристик выпускника школы.

Самостоятельная работа учащихся должна предшествовать проведению первого урока. Этап самостоятельной работы при ограниченном количестве учебных часов, выделяемых на тему, даёт учителю возможность выполнить на первом уроке темы входной контроль, позволяющий оценить остаточные знания учащихся по содержательной линии «Алгоритмы и элементы программирования» ПООП ООО [10].

Следует напомнить, что развитие мыслительных навыков опирается на таксономию педагогических целей. Таксономия, разработанная Б. Блумом, описывающая когнитивную область, включает шесть категорий целей, которые отражают уровни мыслительного поведения: от простого воспроизведения фактов (знания, понимание и применение) до процессов анализа, синтеза и оценки. Иерархия уровней мыслительного поведения означает, что учащийся способен к действиям на определённом уровне, опираясь на мыслительную деятельность на всех предыдущих уровнях.

Знание обозначает способность воспроизведения изученного материала (учащийся воспроизводит определение алгоритма, приводит примеры алгоритмов, перечисляет формы описания алгоритмов, свойства алгоритмов). Понимание обозначает способность интерпретации материала (учащийся объясняет определение алгоритма, распознаёт определение, данное в различных формулировках, устанавливает соответствие между названиями и содержанием свойств алгоритма, описывает свойства алгоритма).

Если уровни мыслительных навыков «знание» и «понимание» не достигнуты, то следует организовать диалог, в ходе которого учитель сможет понять причины непонимания и компенсировать пробелы самостоятельной работы учащихся. Поэтому на этом этапе особая роль отводится педагогическому мастерству учителя.

Для того чтобы быть способным применять знания, учащийся должен понимать необходимую информацию. Поэтому, оценив достижение знания и понимания учащимися базовых понятий алгоритмизации, следует перейти к этапу, обеспечивающему достижение уровня применения знаний, обозначающего умение использовать знания в конкретных

условиях. Этот этап обучения обеспечивается материалами учебника [2, 7];

- параграфом 8.1;
- вопросами и заданиями 8.9–8.11.

В результате, базируясь на сформированных способностях учащихся на уровнях «знание», «понимание» и «применение», можно планировать на уроке выполнение заданий по следующим типам:

- определение результата готового алгоритма, описанного блок-схемой;
- определение результата готового алгоритма для различных исполнителей;
- описание алгоритма с помощью блок-схемы.

Перечисленные типы заданий представлены:

- в вопросах и заданиях к параграфу 8.1;
- интернет-ресурсами по теме «Алгоритм» из Единой коллекции цифровых образовательных ресурсов.

Организовать повторение базовых понятий «алгоритм», «свойства алгоритма» и «формы представления алгоритма», а также составление и анализ алгоритма можно на основе алгоритма Евклида для вычисления наибольшего общего делителя двух чисел, выбрав его как основу для обсуждения на уроке. Алгоритм известен учащимся из курса математики, он будет использован в дальнейшем при решении задач моделирования в среде программирования, при этом алгоритм достаточно прост, краток, может быть представлен в различных формах, позволяет проиллюстрировать базовые алгоритмические конструкции, а также включить учащихся в процесс разработки алгоритма и определения результата при различных исходных данных.

Алгоритм Евклида. Методические рекомендации для учителя

Алгоритм Евклида — алгоритм вычисления наибольшего общего делителя (НОД) двух чисел. Древнегреческие математики называли этот алгоритм «взаимным вычитанием», алгоритм не был открыт Евклидом, он описал его в своём главном труде «Начала». Алгоритм был предложен для натуральных чисел и геометрических длин, в XIX веке он был обобщён на другие типы чисел, назван в честь Евклида.

Предложите учащимся общее описание алгоритма: «В самом простом случае алгоритм Евклида применяется к паре

положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и разницы между бóльшим и меньшим числами. Процесс повторяется, пока числа не станут равными».

Обратите внимание учащихся на то, что предложенное описание не является алгоритмом. Обсудите, отсутствие каких свойств не позволяет это описание отнести к алгоритмам (дискретности, так как нет выраженных команд, и детерминированности, так как описание не содержит точного указания, что делать, если числа станут равными).

Совместно с учениками в процессе диалога представьте словесное описание алгоритма. Обратите внимание на такое преимущество данного описания, как применение привычных для человека слов и обозначений. Выделите каждое свойство алгоритма на примере его словесного описания.

Алгоритм Евклида в форме словесного описания:

1. Начало.
2. Ввести числа A, B .
3. Если $A = B$, то вывод A . Перейти к пункту 6.
4. Если $A \neq B$, то если $A > B$, то $A = A - B$, иначе $B = B - A$.
5. Перейти к пункту 3.
6. Конец.

Совместно с учениками нарисуйте блок-схему алгоритма (рис. 9.1), обсуждая принятые обозначения блоков. Подчеркните такое преимущество блок-схемы, как наглядность. Выделите каждое свойство алгоритма.

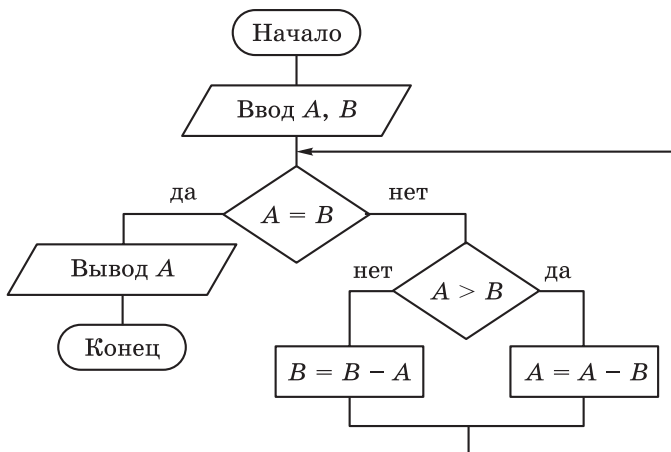


Рис. 9.1. Алгоритм Евклида в форме блок-схемы

Пример «Алгоритм Евклида» позволяет выделить и повторить алгоритмические конструкции: следование, ветвление, цикл.

Используйте этот пример для выполнения заданий по определению результата алгоритма для различных исходных данных.

Обсудите с учащимися возможности описания алгоритма на языке программирования.

Целесообразно во время урока рассмотреть с учениками некоторые предложенные в учебнике задания для анализа и составления алгоритмов из набора вопросов и заданий. Остальные задания следует предложить учащимся в качестве самостоятельной работы, которая позволяет обеспечить базовый уровень знаний при выполнении заданий 8.12–8.22, а также углублённое изучение при выполнении дополнительно заданий 8.23–8.41.

Умение анализировать и составлять алгоритмы для различных исполнителей обеспечивает развитие способностей к анализу, синтезу и оценке, и, как следствие, это обеспечивает «формирование научного типа мышления», определённого ФГОС [9] как предметный результат освоения образовательной программы.

Параграфы 8.2–8.6 дают общее представление о программировании: понятии программы, языка программирования, особенностях языков программирования BASIC и Pascal, метаязыка как способа описания языков программирования, основных этапах технологии работы в среде программирования; содержат вопросы и задания для контроля и набор практических заданий для освоения технологии работы в среде программирования.

Понятия программы, программирования и языка программирования, алфавита, синтаксиса и семантики были изучены в основной школе, поэтому возможно организовать их повторение в диалоге с учителем.

История развития языков программирования BASIC и Pascal, выбранных для изучения, может быть рассмотрена учащимися самостоятельно по материалам параграфа 8.3 и в ходе участия в семинарском занятии «Языки программирования. Этапы большого пути» во внеурочной деятельности.

Освоение понятия и правил метаязыка как способа описания других языков обеспечивает развитие такого универсального учебного действия, как знаково-символическое действие, т. е. представление информации в различной знаковой форме, и обеспечивает развитие высокого уровня абстрагирования. При изучении символов и правил метаязыка следует обратить внимание учащихся на то, что средствами искусственного языка описываются правила другого языка, в том числе и языка программирования. Выбранные для знакомства металингвистические формы Бэкуса – Наура похожи на математические, имеют широкое практическое применение, позволяют выделить ограниченный набор конструкций, достаточных для описания языка программирования. Знакомство с видами и возможностями метаязыков можно продолжить в рамках проектной работы «Знаковая природа метаязыков».

Знакомство со средой программирования будет опираться на знания из курса основной школы, которые необходимо повторить, систематизировать и дополнить новыми сведениями, для чего следует:

- 1) в диалоге с учащимися повторить:
 - состав среды программирования (параграф 8.2);
 - интерфейс среды программирования (параграф 8.5);
 - структуру программы (параграф 8.6);
- 2) обратить внимание учащихся на возможности справочной системы среды программирования для самостоятельного поиска информации, так как в дальнейшем предполагается её активное использование;
- 3) организовать практическую работу учащихся на компьютерах (практические задания 8.31–8.42);
- 4) в диалоге с учащимися описать этапы работы в среде программирования по созданию программы в виде блок-схемы алгоритма и сравнить результат с рисунком 8.10 из учебника. Если учащиеся затрудняются описать этапы работы в виде блок-схемы, то возможно в диалоге с ними проанализировать рис. 8.10.

В качестве закрепления теоретических знаний в области общих представлений о программировании учащимся необходимо подготовить устные ответы на вопросы и задания 8.42–8.58 к параграфам 8.2–8.6.