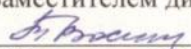


Муниципальное общеобразовательное учреждение
Нижнеякушкинская основная общеобразовательная школа

РАССМОТРЕНО на заседании Педагогического совета Протокол № 1 от «28» августа 2020 г.	СОГЛАСОВАНО Заместителем директора по УВР  /Васина Т.Н./ «28» августа 2020 г.	УТВЕРЖДЕН Приказом МОУ Нижнеякушкинская ООШ № 100 от 28 августа 2020 г. Директор  /А.В.Долгова/
---	--	---

РАБОЧАЯ ПРОГРАММА

Учебный предмет	Информатика
Класса	8 класс
Количество часов - годовых и недельных	Годовых – 68 ч., недельных – 2 ч.
Срок реализации программы	2020-2021 учебный год
Учебная программа (примерная или авторская), на основе которой разработана рабочая программа (издательство, год издания)	Информатика. Рабочие программы Программы для общеобразовательных организаций: 5–9 классы. Учебное издание / Автор-составитель: Л.Л. Босова, А.Ю. Босова.-М.: БИНОМ. Лаборатория знаний, 2017
Учебник, с указанием авторов, издательства, года издания	Информатика: учебник для 8 класса / Л.Л. Босова, А.Ю. Босова – М.: БИНОМ. Лаборатория знаний, 2018
	Васина Татьяна Николаевна
Год разработки программы	2020

Планируемые результаты освоения учебного предмета

Таблица 1. Личностные и метапредметные результаты освоения учебного предмета

Планируемые результаты	
Личностные	Метапредметные
8-й класс	
Личностные результаты – это сформировавшаяся в образовательном процессе система ценностных отношений учащихся к себе, другим участникам образовательного процесса, самому образовательному процессу, объектам познания, результатам образовательной деятельности. Основными личностными результатами, формируемыми при изучении информатики в основной школе, являются: • наличие представлений об информации как важнейшем стратегическом ресурсе развития личности, государства, общества; • понимание роли информационных процессов в современном мире; • способность увязать учебное содержание с собственным жизненным опытом, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества; • готовность к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и ИКТ; • способность и готовность к общению и сотрудничеству со сверстниками и взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой деятельности; • способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ.	Метапредметные результаты – освоенные обучающимися на базе одного, нескольких или всех учебных предметов способы деятельности, применимые как в рамках образовательного процесса, так и в других жизненных ситуациях. Основными метапредметными результатами, формируемыми при изучении информатики в основной школе, являются: • владение общепредметными понятиями «объект», «система», «модель», «алгоритм», «исполнитель» и др.; • владение умениями самостоятельно планировать пути достижения целей; соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; оценивать правильность выполнения учебной задачи; • владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности; • ИКТ-компетентность - широкий спектр умений и навыков использования средств информационных и коммуникационных технологий для сбора, хранения, преобразования и передачи различных видов информации, навыки создания личного информационного пространства (обращение с устройствами ИКТ; создание письменных сообщений; создание графических объектов; коммуникация и социальное взаимодействие; поиск и организация хранения информации; анализ информации).

Таблица 2. Предметные результаты освоения учебного предмета

Планируемые результаты
Предметные результаты включают в себя: освоенные обучающимися в ходе изучения учебного предмета умения специфические для данной предметной области, виды деятельности по получению нового знания в рамках учебного предмета, его преобразованию и применению в учебных, учебно-проектных и социально-проектных ситуациях, формирование научного типа мышления, научных представлений о ключевых теориях, типах и видах отношений, владение научной терминологией, ключевыми понятиями, методами и приемами. В соответствии с федеральным государственным образовательным стандартом общего образования основные предметные результаты изучения информатики в основной школе отражают: • формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств; • формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойствах; • развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами — линейной, условной и циклической; • формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей — таблицы, схемы, графики, диаграммы, с использованием соответствующих программных средств обработки данных; • формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

Выпускник научится	Выпускник получит возможность научиться
8-й класс	
Тема 1. Математические основы информатики Выпускник научится: • записывать в двоичной системе целые числа от 0 до 1024; • переводить заданное натуральное число из десятичной записи в двоичную и из двоичной в десятичную; • сравнивать числа в двоичной записи; складывать и вычитать числа, записанные в двоичной системе счисления; • записывать логические выражения, составленные с помощью операций «и», «или», «не» и скобок, определять истинность такого составного высказывания, если известны значения истинности входящих в него элементарных высказываний; Тема 2. Основы алгоритмизации Выпускник научится: • понимать смысл понятия «алгоритм» и широту сферы его применения; анализировать предлагаемые последовательности команд на предмет наличия у них таких свойств алгоритма как дискретность, детерминированность, понятность, результативность, массовость; • оперировать алгоритмическими конструкциями «следование», «ветвление», «цикл» (подбирать алгоритмическую конструкцию, соответствующую той или иной ситуации; переходить от записи алгоритмической конструкции на алгоритмическом языке к блок-схеме и обратно); • понимать термины «исполнитель», «формальный исполнитель», «среда исполнителя», «система команд исполнителя» и др.; понимать ограничения, накладываемые средой исполнителя и системой команд, на круг задач, решаемых исполнителем; • исполнять линейный алгоритм для формального исполнителя с заданной системой команд; • составлять линейные алгоритмы, число команд в которых не	<i>Выпускник получит возможность:</i> • переводить небольшие десятичные числа из восьмеричной и шестнадцатеричной системы счисления в десятичную систему счисления; • познакомиться с тем, как информация представляется в компьютере, в том числе с двоичным кодированием текстов, графических изображений, звука; • научиться решать логические задачи с использованием таблиц истинности; • научиться решать логические задачи путем составления логических выражений и их преобразования с использованием основных свойств логических операций. <i>Выпускник получит возможность научиться:</i> • исполнять алгоритмы, содержащие ветвления и повторения, для формального исполнителя с заданной системой команд; • составлять все возможные алгоритмы фиксированной длины для формального исполнителя с заданной системой команд; • определять количество линейных алгоритмов, обеспечивающих решение поставленной задачи, которые могут быть составлены для формального исполнителя с заданной системой команд; • подсчитывать количество тех или иных символов в цепочке символов, являющейся результатом работы алгоритма; • по данному алгоритму определять, для решения какой задачи он предназначен; • разрабатывать в среде формального исполнителя короткие алгоритмы, содержащие базовые алгоритмические конструкции.

превышает заданное; • исполнять линейные алгоритмы, записанные на алгоритмическом языке. • исполнять алгоритмы с ветвлениями, записанные на алгоритмическом языке; • понимать правила записи и выполнения алгоритмов, содержащих цикл с параметром или цикл с условием продолжения работы; • определять значения переменных после исполнения простейших циклических алгоритмов, записанных на алгоритмическом языке. Тема 3. Начала программирования <i>Выпускник научится:</i> • составлять несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных с использованием основных управляющих конструкций последовательного программирования и записывать их в виде программ на выбранном языке программирования; выполнять эти программы на компьютере	<i>Ученик получит возможность:</i> • познакомиться с использованием в программах строковых величин и с операциями со строковыми величинами; • создавать программы для решения задач, возникающих в процессе учебы и вне ее.
--	---

Предметные результаты.

Соответствие между предметными результатами, определяющими ФГОС ООО, и содержанием учебника.

Предметные результаты	Соответствие содержания учебника
1.Формирование информационной и алгоритмической культуры;	На формирование данного результата ориентировано все содержание учебников и других компонентов УМК.
2.Формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойства.	
2.2. Формирование представления о понятии алгоритма и его свойствах	§ 2.1. Алгоритмы и исполнители. § 2.2. Способы записи алгоритмов. § 2.3. Объекты алгоритмов. § 2.4. Основные алгоритмические конструкции.
Развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и	

<i>операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами – линейной, условной и циклической.</i>	
3.1. Развитие умений составить и записать алгоритм для конкретного исполнителя	§ 2.1. Алгоритмы и исполнители. § 2.2. Способы записи алгоритмов. § 2.3. Объекты алгоритмов. § 2.4. Основные алгоритмические конструкции.
3.2. Формирование знаний об алгоритмических конструкциях; знакомство с основными алгоритмическими структурами – линейной, условной и циклической.	§ 2.4. Основные алгоритмические конструкции
3.3. Формирование знаний о логических значениях и операциях	§ 3.1. Элементы математической логики. § 2.3. Объекты алгоритмов. § 2.4. Основные алгоритмические конструкции. § 3.5. Программирование разветвляющихся алгоритмов.
3.4. Знакомство с одним из языков программирования	§ 3.1. Общие сведения о языке программирования Паскаль. § 3.2. Организация ввода и вывода данных. § 3.3. Программирование линейных алгоритмов. § 3.4. Программирование разветвляющихся алгоритмов. § 3.5. Программирование циклических алгоритмов.

**Формирование и развитие универсальных учебных действий
определяющими ФГОС ООО, и содержанием учебника.**

Личностные УУД	Соответствие содержания учебника
Действие смыслообразования Действие нравственно-этического оценивания Самопознание и самоопределение:	Глава 3. Начала программирования
Регулятивные УУД	
Целеполагание как постановка учебной задачи; Планирование; Прогнозирование; Контроль; Коррекция; Оценка;	§ 2.1. Алгоритмы и исполнители. § 2.2. Способы записи алгоритмов. § 2.3. Объекты алгоритмов. § 2.4. Основные алгоритмические конструкции. § 3.1. Общие сведения о языке программирования Паскаль. § 3.2. Организация ввода и вывода данных.

Способность к волевому усилию.	§ 3.3. Программирование линейных алгоритмов. § 3.4. Программирование разветвляющихся алгоритмов. § 3.5. Программирование циклических алгоритмов.
Познавательные УУД	
Универсальные логические действия: анализ объектов с целью выделения признаков (существенных) несущественных); синтез как составление целого из частей, в том числе самостоятельно достраивая, восполняя недостающие компоненты; выбор оснований и критериев для сравнения, сериации, классификации объектов; подведение под понятия, выведение следствий; установление причинно-следственных связей, построение логической цепи рассуждений; выдвижение гипотез и их обоснование	§ 1.1. Системы счисления. § 1.3. Элементы алгебры логики.
Действия постановка и решения проблемы: Формулирование проблемы: самостоятельное создание способов решения проблем творческого и поискового характера.	§ 3.5 (3). Многообразие способов записи ветвлений. § 3.6 (4). Различные варианты программирования циклических алгоритмов.

Содержание учебного предмета.

Введение.

Математические основы информатики

Общие сведения о системах счисления. Понятие о непозиционных и позиционных системах счисления. Знакомство с двоичной, восьмеричной и шестнадцатеричной системами счисления, запись в них целых десятичных чисел от 0 до 1024. Перевод небольших целых чисел из двоичной системы счисления в десятичную. Двоичная арифметика.

Компьютерное представление целых чисел. Представление вещественных чисел.

Высказывания. Логические операции. Логические выражения. Построение таблиц истинности для логических выражений. Свойства логических операций. Решение логических задач. Логические элементы.

Аналитическая деятельность:

- анализировать любую позиционную систему как знаковую систему;
- определять диапазон целых чисел в n-разрядном представлении;
- анализировать логическую структуру высказываний;
- анализировать простейшие электронные схемы.

Практическая деятельность:

- переводить небольшие (от 0 до 1024) целые числа из десятичной системы счисления в двоичную, восьмеричную, шестнадцатеричную и обратно;

- выполнять операции сложения и умножения над небольшими двоичными числами;
- строить таблицы истинности для логических выражений;
- вычислять истинностное значение логического выражения.

Основы алгоритмизации

Понятие исполнителя. Неформальные и формальные исполнители. Учебные исполнители (Робот, Чертёжник, Черепаха, Кузнечик, Водолей, Удвоитель и др.) как примеры формальных исполнителей. Их назначение, среда, режим работы, система команд.

Понятие алгоритма как формального описания последовательности действий исполнителя при заданных начальных данных. Свойства алгоритмов. Способы записи алгоритмов.

Алгоритмический язык – формальный язык для записи алгоритмов. Программа – запись алгоритма на алгоритмическом языке. Непосредственное и программное управление исполнителем.

Линейные программы. Алгоритмические конструкции, связанные с проверкой условий: ветвление и повторение. Разработка алгоритмов: разбиение задачи на подзадачи, понятие вспомогательного алгоритма.

Понятие простой величины. Типы величин: целые, вещественные, символьные, строковые, логические. Переменные и константы. Знакомство с табличными величинами (массивами). Алгоритм работы с величинами – план целенаправленных действий по проведению вычислений при заданных начальных данных с использованием промежуточных результатов.

Управление, управляющая и управляемая системы, прямая и обратная связь. Управление в живой природе, обществе и технике.

Аналитическая деятельность:

- приводить примеры формальных и неформальных исполнителей;
- придумывать задачи по управлению учебными исполнителями;
- выделять примеры ситуаций, которые могут быть описаны с помощью линейных алгоритмов, алгоритмов с ветвлениями и циклами;
- определять по блок-схеме, для решения какой задачи предназначен данный алгоритм;
- анализировать изменение значений величин при пошаговом выполнении алгоритма;
- определять по выбранному методу решения задачи, какие алгоритмические конструкции могут войти в алгоритм;
- осуществлять разбиение исходной задачи на подзадачи;
- сравнивать различные алгоритмы решения одной задачи.

Практическая деятельность:

- исполнять готовые алгоритмы для конкретных исходных данных;
- преобразовывать запись алгоритма с одной формы в другую;
- строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя арифметических действий;
- строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя, преобразующего строки символов;
- составлять линейные алгоритмы по управлению учебным исполнителем;
- составлять алгоритмы с ветвлениями по управлению учебным исполнителем;
- составлять циклические алгоритмы по управлению учебным исполнителем;

- строить арифметические, строковые, логические выражения и вычислять их значения;
- строить алгоритм (различные алгоритмы) решения задачи с использованием основных алгоритмических конструкций и подпрограмм.

Начала программирования

Язык программирования. Основные правила одного из процедурных языков программирования (Паскаль, школьный алгоритмический язык и др.): правила представления данных; правила записи основных операторов (ввод, вывод, присваивание, ветвление, цикл) и вызова вспомогательных алгоритмов; правила записи программы.

Этапы решения задачи на компьютере: моделирование – разработка алгоритма – кодирование – отладка – тестирование.

Решение задач по разработке и выполнению программ в выбранной среде программирования.

Аналитическая деятельность:

- анализировать готовые программы;
- определять по программе, для решения какой задачи она предназначена;
- выделять этапы решения задачи на компьютере.

Практическая деятельность:

- программировать линейные алгоритмы, предполагающие вычисление арифметических, строковых и логических выражений;
- разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного уравнения и пр.), в том числе с использованием логических операций;
- разрабатывать программы, содержащие оператор (операторы) цикла;
- разрабатывать программы, содержащие подпрограмму;
- разрабатывать программы для обработки одномерного массива:
 - нахождение минимального (максимального) значения в данном массиве;
 - подсчёт количества элементов массива, удовлетворяющих некоторому условию;
 - нахождение суммы всех элементов массива;
 - нахождение количества и суммы всех четных элементов в массиве;
 - сортировка элементов массива и пр.

Итоговое повторение

Тематическое планирование с указанием количества часов, отводимых на освоение каждой темы

№ п/п	Тема раздела	Кол-во часов на изучение раздела (блока)	Практические работы	Контрольные работы	Проекты

1	Введение	3			
1	Математические основы информатики	24	3	2	«Система счисления»
2	Основы алгоритмизации	25	5	1	-
3	Начала программирования	14	5	1	-
4	Повторение.	2		1	-
5	Итого	68	13	5	-