Рабочая программа по информатике и ИКТ для 8 класса составлена на основе «Примерной программы основного общего образования по информатике и ИКТ (утверждена приказом Минобразования России от 09.03.04. № 1312), примерной программы среднего (полного) общего образования по курсу «Информатика и ИКТ» на базовом уровне. Учебник: Босова Л.Л., Босова А.Ю. Информатика: Учебник для 8 класса. – М.: БИНОМ. Лаборатория знаний, 2016.

**Тематическое планирование с определением основных видов учебной деятельности**

| Тема | Основное содержание по темам | Характеристика деятельности ученика |
|---|---|---|
| **Тема 1. Математические основы информатики (12 часов)** | 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, восьмеричной и шестнадцатеричной системы счисления в десятичную. Двоичная арифметика.<br>Логика высказываний (элементы алгебры логики). Логические значения, операции (логическое отрицание, логическое умножение, логическое сложение), выражения, таблицы истинности. | *Аналитическая деятельность:*<br>• выявлять различие в унарных, позиционных и непозиционных системах счисления;<br>• выявлять общее и отличия в разных позиционных системах счисления;<br>• анализировать логическую структуру высказываний.<br><br>*Практическая деятельность:*<br>• переводить небольшие (от 0 до 1024) целые числа из десятичной системы счисления в двоичную (восьмеричную, шестнадцатеричную) и обратно;<br>• выполнять операции сложения и умножения над небольшими двоичными числами;<br>• записывать вещественные числа в естественной и нормальной форме;<br>• строить таблицы истинности для логических выражений;<br>• вычислять истинностное значение логического выражения. |
| **Тема 2. Основы алгоритмизации (10 часов)** | Учебные исполнители Робот, Удвоитель и др. как примеры формальных исполнителей. Понятие алгоритма как формального описания последовательности действий исполнителя при заданных начальных данных. Свойства алгоритмов. Способы записи алгоритмов.<br>Алгоритмический язык – формальный язык для записи алгоритмов. Программа – запись алгоритма на алгоритмическом языке. Непосредственное и программное управление исполнителем.<br>Линейные программы. Алгоритмические конструкции, связанные с проверкой условий: ветвление и повторение.<br>Понятие простой величины. Типы величин: целые, вещественные, символьные, строковые, логические. Переменные и константы. | *Аналитическая деятельность:*<br>• определять по блок-схеме, для решения какой задачи предназначен данный алгоритм;<br>• анализировать изменение значений величин при пошаговом выполнении алгоритма;<br>• определять по выбранному методу решения задачи, какие алгоритмические конструкции могут войти в алгоритм;<br>• сравнивать различные алгоритмы решения одной задачи.<br><br>*Практическая деятельность:*<br>• исполнять готовые алгоритмы для конкретных исходных данных;<br>• преобразовывать запись алгоритма с одной формы в другую;<br>• строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя арифметических действий;<br>• строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя, преобразующего строки символов;<br>• строить арифметические, строковые, логические выражения и вычислять их значения |

| | | |
|---|---|---|
| | Алгоритм работы с величинами – план целенаправленных действий по проведению вычислений при заданных начальных данных с использованием промежуточных результатов. | |
| **Тема 3. Начала программирования (10 часов)** | Язык программирования. Основные правила языка программирования Паскаль: структура программы; правила представления данных; правила записи основных операторов (ввод, вывод, присваивание, ветвление, цикл).<br>Решение задач по разработке и выполнению программ в среде программирования Паскаль. | *Аналитическая деятельность:*<br>• анализировать готовые программы;<br>• определять по программе, для решения какой задачи она предназначена;<br>• выделять этапы решения задачи на компьютере.<br><br>*Практическая деятельность:*<br>• программировать линейные алгоритмы, предполагающие вычисление арифметических, строковых и логических выражений;<br>• разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного уравнения и пр.), в том числе с использованием логических операций;<br>• разрабатывать программы, содержащие оператор (операторы) цикла |

## Тематическое планирование уроков информатики и ИКТ для 8 класса

| № | Тема урока | Кол-во часов | Содержание учебного предмета | Планируемые предметные результаты |
|---|---|---|---|---|
| 1 | Цели изучения курса информатики и ИКТ. Техника безопасности и организация рабочего места. | 1 | Вводный инструктаж правил по технике безопасности, поведения в кабинете информатики. | Формируются умения и навыки безопасного и целесообразного поведения при работе в компьютерном классе; способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ. |
| **Тема: Математические основы информатики** | | | | |
| 2 | Общие сведения о системах счисления | 1 | Общие сведения о системах счисления. Понятие о непозиционных и позиционных системах счисления. | Учащиеся должны знать различие в унарных, позиционных и непозиционных системах счисления; Учащиеся должны уметь выявлять общее и отличия в разных позиционных системах счисления. |
| 3 | Двоичная система счисления. Двоичная арифметика | 1 | Знакомство с двоичной системой счисления, запись в нее целых десятичных чисел от 0 до 1024. Перевод небольших целых чисел из двоичной системы счисления в десятичную. Двоичная арифметика. | Учащиеся должны уметь выполнять операции сложения и умножения над небольшими двоичными числами; переводить небольшие (от 0 до 1024) целые числа из десятичной системы счисления в двоичную и обратно. |

| | | | | |
|---|---|---|---|---|
| 4 | Восьмеричная и шестнадцатеричные системы счисления. Компьютерные системы счисления | 1 | Знакомство с восьмеричной и шестнадцатеричной системами счисления, запись в них целых десятичных чисел от 0 до 1024. | Учащиеся должны уметь выполнять операции сложения и умножения над небольшими двоичными числами; переводить небольшие (от 0 до 1024) целые числа из десятичной системы счисления в восьмеричную, шестнадцатеричную и обратно. |
| 5 | Правило перевода целых десятичных чисел в систему счисления с основанием q | 1 | Двоичная арифметика. Правило перевода целых десятичных чисел в систему счисления с основанием q. Перевод десятичиных чисел в систему счисления с основанием q. | Учащиеся должны уметь выполнять операции сложения и умножения над небольшими двоичными числами; переводить небольшие (от 0 до 1024) целые числа из десятичной системы счисления в двоичную (восьмеричную, шестнадцатеричную) и обратно. |
| 6 | Представление целых чисел | 1 | Компьютерное представление целых чисел. | Учащиеся должны понимать ограничения на диапазон значений величин при вычислениях; понимать возможности представления вещественных чисел в широком диапазоне, важном для решения научных и инженерных задач. Учащиеся должны уметь записывать вещественные числа в естественной и нормальной форме. |
| 7 | Представление вещественных чисел | 1 | Представление вещественных чисел. | Учащиеся должны понимать ограничения на диапазон значений величин при вычислениях; понимать возможности представления вещественных чисел в широком диапазоне, важном для решения научных и инженерных задач. Учащиеся должны уметь записывать вещественные числа в естественной и нормальной форме. |
| 8 | Высказывание. Логические операции. | 1 | Высказывания. Логические операции. Логические выражения. | Учащиеся должны уметь выполнять анализ логической структуры высказываний; понимать связи между логическими операциями и логическими связками, между логическими операциями и операциями над множествами анализировать логическую структуру высказываний. |
| 9 | Построение таблиц истинности для логических выражений | 1 | Построение таблиц истинности для логических выражений. | Учащиеся должны уметь проводить формализацию и анализ логической структуры высказываний; видеть инвариантную сущность во внешне различных объектах. строить таблицы истинности для логических выражений. |

| | | | | |
|---|---|---|---|---|
| 10 | Свойства логических операций. | 1 | Свойства логических операций. | Учащиеся должны уметь проводить анализ и преобразования логических выражений;<br>- видеть инвариантную сущность во внешне различных объектах (законы алгебры логики и законы алгебры чисел). |
| 11 | Решение логических задач | 1 | Решение логических задач. | Учащиеся должны уметь проводить формализацию высказываний, анализ и преобразования логических выражений;<br>выбирать метод для решения конкретной задачи.<br>вычислять истинностное значение логического выражения. |
| 12 | Логические элементы | 1 | Логические элементы. | Учащиеся должны уметь представлять одну и ту же информацию в разных формах (таблица истинности, логическое выражение, электронная схема). |
| 13 | Обобщение и систематизация основных понятий темы «Математические основы информатики». Проверочная работа | 1 | Основные понятия темы «Математические основы информатики» | Учащиеся должны уметь выполнять анализ различных объектов;<br>видеть инвариантную сущность во внешне различных объектах;<br>Учащиеся должны уметь переводить небольшие целые числа из двоичной, восьмеричной и шестнадцатеричной системы счисления в десятичную строить таблицы истинности для логических выражений, вычислять истинностное значение логического выражения. |
| **Тема: Основы алгоритмизации** | | | | |
| 14 | Алгоритмы и исполнители | 1 | Учебные исполнители Робот, Удвоитель и др. как примеры формальных исполнителей. Понятие алгоритма как формального описания последовательности действий исполнителя при заданных начальных данных. Свойства алгоритмов. Способы записи алгоритмов. | Учащиеся должны понимать смысл понятия «алгоритм» и широты сферы его применения;<br>Учащиеся должны понимать ограничения, накладываемые средой исполнителя и системой команд на круг задач, решаемых исполнителем. |
| 15 | Способы записи алгоритмов | 1 | Алгоритмический язык – формальный язык для записи алгоритмов. Программа – запись алгоритма на алгоритмическом языке. Непосредственное и программное управление исполнителем. | Учащиеся должны уметь анализировать предлагаемые последовательности команд на предмет наличия у них таких свойств алгоритма как дискретность, детерминированность, понятность, результативность, массовость;<br>Учащиеся должны понимать преимущества и недостатки той или иной формы записи алгоритмов;<br>Учащиеся должны уметь переходить от одной формы записи алгоритмов к |

| | | | | |
|---|---|---|---|---|
| | | | | другой;<br>выбирать форму записи алгоритма, соответствующую решаемой задаче |
| 16 | Объекты алгоритмов | 1 | Понятие простой величины. Типы величин: целые, вещественные, символьные, строковые, логические. Переменные и константы. | Учащиеся должны понимать сущность понятия «величина»; понимать границы применимости величин того или иного типа. |
| 17 | Алгоритмическая конструкция следование | 1 | Линейные программы. Алгоритмические конструкции, связанные с проверкой условий: ветвление и повторение | Учащиеся должны научиться выделять линейные алгоритмы в различных процессах;<br>понимать ограниченности возможностей линейных алгоритмов.<br>определять по блок-схеме, для решения какой задачи предназначен данный алгоритм;<br>анализировать изменение значений величин при пошаговом выполнении алгоритма;<br>определять по выбранному методу решения задачи, какие алгоритмические конструкции могут войти в алгоритм;<br>сравнивать различные алгоритмы решения одной задачи. |
| 18 | Алгоритмическая конструкция ветвление. Полная форма ветвления | 1 | Алгоритмические конструкции, связанные с проверкой условий: ветвление и повторение | Учащиеся должны уметь выделять алгоритмы с ветвлением в различных процессах;<br>определять по блок-схеме, для решения какой задачи предназначен данный алгоритм;<br>анализировать изменение значений величин при пошаговом выполнении алгоритма;<br>сравнивать различные алгоритмы решения одной задачи.<br>исполнять готовые алгоритмы для конкретных исходных данных;<br>строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя арифметических действий; |
| 19 | Неполная форма ветвления | 1 | Линейные программы. Алгоритмические конструкции, связанные с проверкой условий: ветвление и повторение | Учащиеся должны уметь понимать ограниченность возможностей линейных алгоритмов.<br>анализировать изменение значений величин при пошаговом выполнении алгоритма;<br>определять по выбранному методу решения задачи, какие алгоритмические конструкции могут войти в алгоритм;<br>сравнивать различные алгоритмы решения одной задачи.<br>исполнять готовые алгоритмы для |

| | | | | |
|---|---|---|---|---|
| | | | | конкретных исходных данных; |
| 20 | Алгоритмическая конструкция повторение. Цикл с заданным условием продолжения работы | 1 | Алгоритм работы с величинами – план целенаправленных действий по проведению вычислений при заданных начальных данных с использованием промежуточных результатов. | Учащиеся должны уметь выделять циклические алгоритмы в различных процессах.<br>определять по блок-схеме, для решения какой задачи предназначен данный алгоритм;<br>анализировать изменение значений величин при пошаговом выполнении алгоритма;<br>сравнивать различные алгоритмы решения одной задачи.<br>исполнять готовые алгоритмы для конкретных исходных данных;<br>строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя арифметических действий;<br>строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя, преобразующего строки символов; |
| 21 | Цикл с заданным условием окончания работы | 1 | Алгоритм работы с величинами – план целенаправленных действий по проведению вычислений при заданных начальных данных с использованием промежуточных результатов. | Учащиеся должны уметь выделять циклические алгоритмы в различных процессах.<br>исполнять готовые алгоритмы для конкретных исходных данных;<br>строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя арифметических действий;<br>строить арифметические, строковые, логические выражения и вычислять их значения |
| 22 | Цикл с заданным числом повторений | 1 | Алгоритм работы с величинами – план целенаправленных действий по проведению вычислений при заданных начальных данных с использованием промежуточных результатов. | Учащиеся должны уметь выделять циклические алгоритмы в различных процессах.<br>исполнять готовые алгоритмы для конкретных исходных данных;<br>преобразовывать запись алгоритма с одной формы в другую;<br>строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя арифметических действий;<br>строить арифметические, строковые, логические выражения и вычислять их значения |
| 23 | Обобщение и систематизация основных понятий темы «Основы алгоритмизации». Проверочная работа | 1 | Основные понятия темы «Основы алгоритмизации» | Учащиеся должны уметь самостоятельно планировать пути достижения целей;<br>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; владеть основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности. |
| | **Тема Начала программирования** | | | |
| 24 | Общие сведения о языке программирования Паскаль | 1 | Язык программирования. Основные правила языка программирования Паскаль: структура программы; правила представления данных; правила записи основных операторов (ввод, вывод, присваивание, ветвление, цикл). | Учащиеся должны уметь проводить анализ языка Паскаль как формального языка; анализировать готовые программы; определять по программе, для решения какой задачи она предназначена; |
| 25 | Организация ввода и вывода данных | 1 | Основные правила языка программирования Паскаль: структура программы; правила представления данных; правила записи основных операторов (ввод, вывод, присваивание, ветвление, цикл). | Учащиеся должны уметь выполнять запись простых последовательностей действий на формальном языке. анализировать готовые программы; выделять этапы решения задачи на компьютере. |
| 26 | Программирование линейных алгоритмов | 1 | Основные правила языка программирования Паскаль: структура программы; правила представления данных; правила записи основных операторов (ввод, вывод, присваивание, ветвление, цикл). Решение задач по разработке и выполнению программ в среде программирования Паскаль. | Учащиеся должны уметь выделять этапы решения задачи на компьютере. программировать линейные алгоритмы, предполагающие вычисление арифметических, строковых и логических выражений; |
| 27 | Программирование разветвляющихся алгоритмов. Условный оператор. | 1 | Основные правила языка программирования Паскаль: структура программы; правила представления данных; правила записи основных операторов (ввод, вывод, присваивание, ветвление, цикл). Решение задач по разработке и выполнению программ в среде программирования Паскаль. | Учащиеся должны уметь выделять этапы решения задачи на компьютере. разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного уравнения и пр.), в том числе с использованием логических операций; |
| 28 | Составной оператор. Многообразие способов записи ветвлений. | 1 | Основные правила языка программирования Паскаль: структура программы; правила представления данных; правила записи основных | Учащиеся должны уметь разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного |

| | | | | |
|---|---|---|---|---|
| | | | операторов (ввод, вывод, присваивание, ветвление, цикл). Решение задач по разработке и выполнению программ в среде программирования Паскаль. | уравнения и пр.), в том числе с использованием логических операций; |
| 29 | Программирование циклов с заданным условием продолжения работы. | 1 | Решение задач по разработке и выполнению программ в среде программирования Паскаль. | Учащиеся должны уметь разрабатывать программы, содержащие оператор (операторы) цикла |
| 30 | Программирование циклов с заданным условием окончания работы. | 1 | Решение задач по разработке и выполнению программ в среде программирования Паскаль. | Учащиеся должны уметь разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного уравнения и пр.), в том числе с использованием логических операций;<br>разрабатывать программы, содержащие оператор (операторы) цикла |
| 31 | Программирование циклов с заданным числом повторений. | 1 | Решение задач по разработке и выполнению программ в среде программирования Паскаль. | Учащиеся должны уметь разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного уравнения и пр.), в том числе с использованием логических операций;<br>разрабатывать программы, содержащие оператор (операторы) цикла |
| 32 | Различные варианты программирования циклического алгоритма. | 1 | Решение задач по разработке и выполнению программ в среде программирования Паскаль. | Учащиеся должны уметь разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного уравнения и пр.), в том числе с использованием логических операций;<br>разрабатывать программы, содержащие оператор (операторы) цикла |
| 33 | Обобщение и систематизация основных понятий темы «Начала программирования ». Проверочная работа. | 1 | Основные понятия темы «Начала программирования» | Учащиеся должны уметь определять по программе, для решения какой задачи она предназначена;<br>разрабатывать программы, содержащие оператор/операторы ветвления (решение линейного неравенства, решение квадратного уравнения и пр.), в том числе с использованием логических операций;<br>разрабатывать программы, содержащие оператор (операторы) |

| | | | | |
|---|---|---|---|---|
| | | | | цикла |
| 34 | Основные понятия курса.<br><br>Итоговое тестирование. | 1 | Систематизированные представления об основных понятиях курса информатики | |

Итого 34 часа. Проверочных работ – 3.

**Планируемые результаты изучения учебного предмета**

Обучающийся научится:

– декодировать и кодировать информацию при заданных правилах кодирования;

– оперировать единицами измерения количества информации;

– оценивать количественные параметры информационных объектов и процессов (объём памяти, необходимый для хранения информации; время передачи информации и др.);

– записывать в двоичной системе целые числа от 0 до 256;

– составлять логические выражения с операциями И, ИЛИ, НЕ; определять значение логического выражения; строить таблицы истинности;

– анализировать информационные модели (таблицы, графики, диаграммы, схемы и др.);

– перекодировать информацию из одной пространственно-графической или знаково-символической формы в другую, в том числе использовать графическое представление (визуализацию) числовой информации;

– выбирать форму представления данных (таблица, схема, график, диаграмма) в соответствии с поставленной задачей;

– строить простые информационные модели объектов и процессов из различных предметных областей с использованием типовых средств (таблиц, графиков, диаграмм, формул и пр.), оценивать адекватность построенной модели объекту-оригиналу и целям моделирования;

– 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 превышает заданное;

– ученик научится исполнять записанный на естественном языке алгоритм, обрабатывающий цепочки символов.

– 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;

– разрабатывать и записывать на языке программирования короткие алгоритмы, содержащие базовые алгоритмические конструкции.

Обучающийся получит возможность:

– углубить и развить представления о современной научной картине мира, об

информации как одном из основных понятий современной науки, об информационных процессах и их роли в современном мире;

– научиться определять мощность алфавита, используемого для записи сообщения;

– научиться оценивать информационный объём сообщения, записанного символами произвольного алфавита;

– переводить небольшие десятичные числа из восьмеричной и шестнадцатеричной системы счисления в десятичную систему счисления;

– 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;

– 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;

– исполнять записанные на алгоритмическом языке циклические алгоритмы обработки одномерного массива чисел (суммирование всех элементов массива; суммирование элементов массива с определёнными индексами; суммирование элементов массива, с заданными свойствами; определение количества элементов массива с заданными свойствами; поиск наибольшего/ наименьшего элементов массива и др.);

– разрабатывать в среде формального исполнителя короткие алгоритмы, содержащие базовые алгоритмические конструкции;

– разрабатывать и записывать на языке программирования эффективные алгоритмы, содержащие базовые алгоритмические конструкции.