**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Министерство образования Республики Мордовия**

**Администрация Торбеевского муниципального района**

**МБОУ "Торбеевская средняя общеобразовательная школа №3"**

|  |  |  |
| --- | --- | --- |
| РАССМОТРЕНО  Методическое объединение учителей математики, физики и информатики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Мелёшкин В.С.  Протокол №1  от «30» августа 2024 г. |  | УТВЕРЖДЕНО  Директор школы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Хакназарова И.Г.  Приказ №80/8  от «02» сентября 2024 г. |

**РАБОЧАЯ ПРОГРАММА**

**внеурочной деятельности 10 класса**

**Основы программирования**

**Составитель: Мелёшкин В.С.**

**п. Торбеево** **2024**

**Пояснительная записка.**

Требования высших учебных заведений к выпускникам школ существенно возросли и расширились. Это обстоятельство вынуждает особенно тщательно согласовывать разработку программы преподавания предмета информатики в школе с дальнейшим изучением соответствующего материала в вузе.

Поэтому необходимо «Основы программирования» выделить как отдельный курс, изучение которого базируется на современных системах и языках программирования, решении практических задач. Значительное внимание уделяется фундаментальным вопросам технического и технологического обеспечения информатики, логическим и арифметическим основам компьютера.

Цель курса: систематизировать и расширить знания обучающихся в области программивания.

Содержание курса сочетает в себе три существенных сейчас основных подхода в обучении информатики в школе и отражает важнейшие аспекты ее образовательной значимости:

«пользовательский» аспект, связанный с формированием компьютерной грамотности, информационной культуры, подготовкой школьников к практической деятельности в условиях широкого использования информационных технологий. Такое обучение целесообразно вводить как можно раньше, чтобы учащиеся могли использовать различные доступные их возрасту программные продукты, применяя компьютер в качестве инструмента для своих целей;

алгоритмический (программистский) аспект, связанный в большей мере с развитием мышления учащихся;

кибернетический аспект, связанный с формированием мировоззренческих представлений о роли информации в управлении, специфики самоуправляющих систем, общих закономерностях информационных процессов в системах разной природы.

Курс в целом охватывает следующие группы вопросов:

Вопросы, связанные с пониманием сущности информационных процессов, информационных основ процессов управления в системах различной природы и представлением о передаче информации, канале передачи информации, количестве информации, способах представления информации для формального исполнителя (информационный аспект);

Методы и средства формализованного описания действий исполнителя (алгоритмический аспект);

Вопросы, связанные с выбором исполнителя для решения задачи, анализом его свойств, возможностей и эффективности его применения для решения данной задачи;

Вопросы, связанные с адекватным описанием реальных объектов и явлений для их использования с помощью ЭВМ, проведение компьютерного эксперимента (моделирование).

В процессе обучения заложены следующие необходимые умения и навыки:

умение планировать структуру действий, необходимых для достижения заданной цели при помощи фиксированного набора средств;

умение организовывать поиск информации, необходимой для решения поставленной задачи;

умение строить информационные структуры (модели) для описания объектов и систем;

технические навыки работы с компьютером и его периферийными устройствами.

Именно при формировании таких навыков и умений можно будет говорить о высокой эффективности использования компьютеров и достижении учащихся не только уровня знаний и умений, заложенных в «Обязательный минимум содержания образования», но и достаточно более высокого уровня, необходимого в дальнейшем обучении.

Курс предлагается для изучения в 10 классе, рассчитан на 17 часа и предусматривает выполнение 7 практических работ по различным темам, а также самостоятельную работу учащихся над проектами.

При изучении языка программирования следует стремиться к использованию формального описания конструкций языка программирования (синтаксис и семантику), а при решении задач использовать только изученные элементы. Это способствует не только глубокому пониманию учебного материала, но и дает возможность использовать изученный материал в дальнейшем.

При обучении программированию следует добиваться определенного стиля оформления текстов программ.

«Программировать трудно. Программировать хорошо – очень трудно. И о том, как нужно правильно программировать пишут много. Но … сколько людей – столько и мнений…», - написал редактор перевода очередной книги «Практика программирования**».** (Цитата из книги «Программирование на Паскале для начинающих». Для студентов физического факультета /ГОУ ВПО «Кемеровский госуниверситет»; сост. Т. Ю. Павлова. Кемерово, 2007 г.). Однако из всех советов по стилю программирования можно все же выбрать, несомненно, полезные для начинающего программиста, которые можно рекомендовать детям на уроках.

***Советы для учащихся.***

При написании программы нужно руководствоваться принципом, что написать программу – это не только разработать ее текст, исправить ошибки и заставить ее выполняться достаточно быстро. Нужно написать ее так, чтобы в случае необходимости другой человек или вы сами через некоторое время, могли легко разобраться в этой программе, чтобы модернизировать ее. А программу, написанную хорошо, куда проще понять, чем написанную плохо. Кроме того, культура написания текста позволит создавать программы с меньшим числом ошибок.

Текст программы должен быть прост и понятен, для этого придерживайтесь правил:

имя переменной или функции должно содержать информацию о ней. Это не просто, так как используется латинский алфавит, но латинскими буквами можно написать русское имя или использовать знакомые английские слова; используйте одни и те же имена для счетчиков цикла (например: i, j).

форматируйте текст программы. Для этого сдвигайте начало написания части операторов вправо на несколько позиций. Сдвигайте операторы, находящиеся внутри составного оператора begin – end (для языка программирования Pascal), пишите ветвь else под ветвью then в условном операторе и т.п.

При работе с составными типами данных необходимо их предварительно описывать в разделе type.

В хорошо написанной программе каждая константа может встречаться только один раз: или в разделе констант, или в начале программы (подпрограммы) для присваивания значения соответствующей переменной.

1. **Планируемые результаты изучения предмета, курса**

**Основные понятия о языках программирования**

*В результате изучения темы выпускник научится:*

- отличать компилятор от интерпретатора;

- распознавать основные средства, алфавит языка программирования, служебные слова, структуру программы;

- использовать команду присваивания;

- использовать типы данных, переменные и константы;

- составлять общую характеристику системы программирования.

*-* составлять простейшие программы с использованием команды присваивания;

- компилировать и отлаживать программу.

*В результате изучения темы выпускник получит возможность научится:*

*-*программировать простейшие задачи и задачи повышенного уровня сложности .

**Встроенные функции в языки программирования.**

*В результате изучения темы выпускник научится:*

*-* записывать встроенные функции;

- использовать оператор ввода;

- использовать оператор вывода;

-использовать оператор присваивания;

-использовать оператор комментария;

- использовать блок-схему конструкции «Следование»

*В результате изучения темы выпускник получит возможность научится:*

Составлять простейшие программы с использованием операторов присваивания, ввода-вывода информации.

**Условный оператор.**

*В результате изучения темы выпускник научится:*

- использовать блок-схему полной формы условного оператора, блок-схему сокращенной формы условного оператора, блок-схему вложенного условного оператора, оператора варианта;

- использовать запись условного оператора с помощью языка программирования;

- использовать запись оператора варианта с помощью языка программирования.

- составлять программы с использованием полной и сокращенной формы условного оператора;

- составлять программы с использованием вложенных условных операторов;

- составлять программы, содержащие сложные условия;

- составлять программы с использованием оператора варианта.

*В результате изучения темы выпускник получит возможность научится:*

*-*программировать простейшие задачи и задачи повышенного уровня сложности .

**Циклы.**

*В результате изучения темы выпускник научится:-* использоватьблок-схему цикла с параметром, блок-схему цикла с предусловием, блок-схему цикла с постусловием;

- использовать запись циклов с логическим условием;

- использовать запись циклов с помощью языка программирования.

*-* программировать задачи с использованием циклов с параметром, с предусловием, с постусловием;

- программировать задачи с использованием циклов с логическим условием;

- программировать задачи с использованием вложенных циклов.

*В результате изучения темы выпускник получит возможность научится:*

*-*программировать простейшие задачи и задачи повышенного уровня сложности .

**Массивы.**

*В результате изучения темы выпускник научится:*

- применять понятие массива, переменной с индексом, простой переменной;

- применять понятие одномерного массива;

-понимать, как происходит присвоение значений элементам одномерного массива;

-понимать работу датчика случайных чисел;

- применять понятие двумерного массива;

-понимать, как происходит присвоение значений элементам двумерного массива;

*-* организовывать задание одномерного и двумерного массива;

- подсчитывать сумму, произведение и количество элементов одномерного и двумерного массивов, отвечающих заданным условиям;

- находить максимальный и минимальный элемент одномерного и двумерного массивов;

-осуществлять перестановку элементов одномерного и двумерного массивов;

- осуществлять слияние и отбор данных в одномерных и двумерных массивах;

- осуществлять поиск, подбор и группировку данных;

- осуществлять сортировку элементов массива.

*В результате изучения темы выпускник получит возможность научится:*

*-*программировать простейшие задачи и задачи повышенного уровня сложности .

**Строковый, символьный типы данных.**

*В результате изучения темы выпускник научится:*

*-* использовать основные сведения о символьных величинах;

- использовать реализацию символьных величин с помощью языков программирования;

*-* использовать стандартные функции для работы с символьными величинами: сравнение, конкатенацию, копирование, удаление, замену (вставку), длину строки, подстроку.

- программировать алгоритмы обработки текста;

- программировать задачи на поиск и подсчет (лингвистическая статистика).

*В результате изучения темы выпускник получит возможность научится:*

*-*программировать простейшие задачи и задачи повышенного уровня сложности .

**Процедуры и функции. Понятие подпрограмм. Механизм реализации подпрограмм с помощью процедур и функций.**

*В результате изучения темы выпускник научится:*

*-* использовать понятие подпрограммы;

- использовать понятие процедуры;

- использовать оператор процедуры;

- использовать понятие функции;

-использовать оператор задания функции;

- понимать, как осуществляется обмен между основной программой и подпрограммой;

- использовать понятие глобальной и локальной переменной;

- использовать механизм реализации подпрограммы с помощью процедур и функций.

- уметь программировать задачи с использованием процедур и функций.

*В результате изучения темы выпускник получит возможность научится:*

*-*программировать простейшие задачи и задачи повышенного уровня сложности .

**Работа с файлами**

*В результате изучения темы выпускник научится:*

*-*использовать понятие файла: имя, расширение;

- использовать типы файлов;

- использовать операции над файлами;

- использовать файлы с произвольным доступом.

*-*решать задачи на программирование с использованием файлов.

*В результате изучения темы выпускник получит возможность научится:*

*-*программировать простейшие задачи и задачи повышенного уровня сложности .

**Содержание учебного предмета, курса.**

**Основные понятия о языках программирования (1ч).**

История появления языков QBASIC и TURBO PASCAL. Языки программирования интерпретирующего и компилирующего типа. Основные средства языка. Алфавит языка. Служебные слова. Структура программы. Переменные и константы. Типы данных. Присваивание. Технология разработки программного обеспечения. Система и языки программирования. Общая характеристика системы программирования. Система оперативной подсказки. Редактор исходного текста. Пример простой программы. Компиляция и отладка программы.

**Встроенные функции в языки программирования (2 ч).**

Запись на языке QBASIC функций |X|*,* sin x, Cos x, tg x, целая часть x, x2 и т. д.Операторы INPUT, PRINT. Работа оператора присваивания. Выражение вида N=N+1. Блок-схема конструкции «Следование». Оператор комментария. Набор, отладка и запуск программы в среде программирования QBASIC.

**Условный оператор (2 ч).**

Блок-схема конструкции «Ветвление».Условный оператор. Служебные слова IF, THEN, ELSE. Работа полного условного оператора. Работа сокращенного условного оператора.Вложенные условные операторы. Блок – схема вложенных условных операторов. Конструкция вложенного условного оператора. Служебные слова IF, THEN, ELSE, END IF. Логические связки AND, OR, NOT. Программирование задач с использованием вложенных условных операторов и сложных условий. Блок-схема оператора варианта. Структура оператора варианта. Служебные слова SELECT, CASE, END SELECT.

**Циклы (3 ч).**

Блок-схема цикла с предусловием. Тело цикла. Оператор цикла с предусловием. Служебные слова WHILE, WEND. Работа оператора цикла с предусловием. Блок-схема цикла с постусловием. Тело цикла. Оператор цикла с постусловием. Служебные слова DO, LOOP, WHILE, UNTIL. Работа оператора цикла с постусловием. Блок-схема цикла с параметром. Тело цикла. Параметр цикла. Начальное и конечное значение параметра цикла. Шаг цикла. Оператор цикла с параметром. Служебные слова FOR, TO, NEXT. Работа оператора цикла с параметром. Конструкция вложенных циклов. Конструкция сочетания цикла и условного оператора.

**Массивы (4 ч).**

Понятие массива. Понятие индекса. Переменная с индексом. Простая переменная. Одномерные массивы. Описание массива: DIM имя массива (n1 TO n2) AS тип элементов. Присвоение значений элементам массива. Способы задания одномерных массивов. Понятие матрицы. Двумерные массивы. Нумерация элементов двумерного массива. Способы описания двумерного массива. Способы задания двумерных массивов. Проверка на четность. Подсчет количества элементов, отвечающих заданным условиям. Нахождение суммы, произведения и количества элементов массива, отвечающих заданным условиям. Максимальный и минимальный элементы. Сортировка элементов массива. Метод «пузырька». Оператор SWAP. Перестановка элементов массива. Поиск, подбор и группировка данных. Слияние и отбор данных в одномерных и двумерных массивах.

**Строковый, символьный типы данных (2 ч).**

Основные сведения о символьных величинах. Стандартные функции, процедуры для работы с символьными величинами: сравнение, конкатенация, копирование, удаление, замена (вставка), длина строки, подстрока числа и строки.Понятие шифровки, дешифровки текста. Способы шифровки текста.

**Процедуры и функции. Понятие подпрограмм. Механизм реализации подпрограмм с помощью процедур и функций (1 ч).**

Понятие рекурсии. Рекурсивные алгоритмы. Задачи, сводимые к рекурсивным. Понятие подпрограммы, процедуры, функции, Описание процедуры. Оператор SUB. Описание функции. Оператор FUNCTION. Механизм реализации подпрограмм с помощью процедур и функций.Обмен информацией между основной программой и подпрограммой. Глобальные и локальные переменные.

**Работа с файлами (1 ч).**

Понятие файла. Классификация файлов. Операции над файлами: открытие файла, чтение и запись обрабатываемых данных, закрытие файлов. Файл произвольного доступа. Операторы и функции работы с файлом произвольного доступа.

**Повторение. Решение задач. (1 ч).**

**Календарно - тематическое планирование.**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **урока** |  | **Содержание учебного материала** | **Кол-во часов** |
| 1-2 |  | **Основные понятия о языках программирования (2 ч).**   * 1. **Понятие о языке программирования высокого и низкого уровня.**   *Цели:* дать основные сведения о языках программирования, средствах языков, структуре программ.  *Краткий обзор теоретического материала.* История создания языков программирования Бейсик и Паскаль. Система и языки программирования. Технология разработки программного обеспечения. Общая характеристика системы программирования. Основные средства языка программирования. Алфавит языка. Служебные слова. Структура программы. Система оперативной подсказки. Редактор исходного текста. Пример простой программы. Компиляция и отладка программы.  *Примеры возможных заданий.*  Подготовить ответы на вопросы:   * + 1. Для чего служит компьютер?     2. Что называется алгоритмом?     3. Что называют алфавитом языка?     4. Что такое интерпретация и компиляция?     5. Что представляет собой программа?     6. Что включает в себя алфавит языка QBASIC?     7. Выполнение творческого проекта (отчета) по теме «История языков программирования».   1. **Переменные и константы. Описание типов.**   *Цели:* дать основные сведения о константах, переменных; познакомить учащихся с различными типами данных.  *Краткий обзор теоретического материала.* Понятие переменной. Переменные числового типа: целые, длинные целые, обычной точности, двойной точности; переменные символьного типа: строка переменной длины, строка фиксированной длины; переменные пользовательского типа. Понятие константы.  *Примеры возможных заданий.*  Подготовить ответы на вопросы:   * + 1. Чем характеризуется переменная?     2. В каком разделе происходит описание переменных?     3. Как описываются переменные?     4. Перечислите типы данных.     5. Какова структура программы?     6. С чего начинается основная программа?     7. Чем заканчивается программа?     8. Определите тип выражения: 1+0.0; 20/4. | 1  1 |
| 3-6 |  | **Встроенные функции в языки программирования. Простейшие операторы языка**  **(4 ч)**   1. **Встроенные функции.**   *Цели:* дать представления о встроенных функциях; научить учащихся использовать встроенные функции при записи математических выражений на языке программирования QBASIC.  *Краткий обзор теоретического материала.* Запись на языке QBASIC функций |X|*,* sin x, Cos x, tg x, целая часть x, x2 и т. д.  *Примеры возможных заданий.*  Записать на языке QBASIC следующие выражения:   * + - 1. Y=5sinX       2. Z= 14X4 – 5X3 +11X – 17       3. X1=       4. D=b2 – 4ac   Сделать математическую запись следующих выражений:   1. LOG(X)-32/ (ABS(X)-1) 2. SQR(64) 3. **Операторы ввода и вывода информации. Оператор присваивания.**   *Цели*: познакомить учащихся с операторами ввода – вывода информации, оператором присваивания; программировать простейшие задачи с использованием операторов ввода – вывода и присваивания.  *Краткий обзор теоретического материала.* Операторы INPUT, PRINT. Работа оператора присваивания. Выражение вида N=N+1. Блок-схема конструкции «Следование»  *Примеры возможных заданий.*   * + - 1. Вывести на экран приветствие: «Добрый день»       2. Вычислить периметр прямоугольника со сторонами А и В.       3. Занести в переменную Р по очереди значения некоторых вышеприведенных функций, аргументом которых является сумма А+В.       4. Напечатайте случайное число в пределах от А до В.       5. Напечатайте значения переменной Р в формате: 3 позиции для целой части и 2 для дробной.  1. **Создание и отладка элементарной программы. Печать исходного текста. Комментарии**.   *Цели:* отрабатывать навыки написания программ в среде программирования QBASIC, навыки работы с функциональными клавишами среды QBASIC.  *Краткий обзор теоретического материала.* Оператор комментария. Набор, отладка и запуск программы в среде программирования QBASIC.  *Примеры возможных заданий.*   1. Известны количество жителей (A) в государстве и площадь его территории (S). Определите плотность населения в государстве. 2. Даны два целых числа. Найти их среднее арифметическое. Модернизируйте программу так, чтобы на экран было выведено среднее геометрическое этих чисел. 3. Составить программу обмена значениями двух переменных.      1. **Практическая работа №1 «Составление простейших программ».**   Смотрите приложение. | 1  1  1  1 |
| 7-11 |  | **Условный оператор (5 ч).**   1. **Условный оператор. Полная и неполная формы оператора.**   *Цели:* познакомить учащихся с полной и сокращенной формой условного оператора. Рассмотреть типы задач, для которых лучше использовать полную и для которых – неполную форму ветвления.  *Краткий обзор теоретического материала.* Блок-схема конструкции «Ветвление».Условный оператор. Служебные слова IF, THEN, ELSE. Работа полного условного оператора. Работа сокращенного условного оператора.  *Примеры возможных заданий.*   1. Дано вещественное число. Вывести на экран его абсолютную величину (условно принимая, что соответствующей стандартной функции нет). Полный условный оператор не использовать. 2. Определите значение переменной x после выполнения фрагмента программы:   F = 5 : D = 7  IF F>=D THEN X = F ELSE X = D   1. Даны три целых числа. Вывести на экран те из них, которые являются четными. 2. Известны два расстояния: одно в километрах, другое в футах (1фут=0,45 м). Какое из расстояний меньше? 3. Составить программу вычисления значений функции  , еслиx>0,  Y =   x 2 , если x ≤0. 4. **Вложенные условные операторы. Логические условия.**   *Цели:* познакомить учащихся с использованием вложенных условных операторов и сложных (логических) условий при программировании различных задач.  *Краткий обзор теоретического материала.* Вложенные условные операторы. Блок – схема вложенных условных операторов. Конструкция вложенного условного оператора. Служебные слова IF, THEN, ELSE, END IF. Логические связки AND, OR, NOT. Программирование задач с использованием вложенных условных операторов и сложных условий.  *Примеры возможных заданий.*   1. Даны три числа. Составить программу вывода на экран наибольшего из них 2. Используя вложенные условные операторы; 3. Используя последовательные условные операторы. 4. Выяснить, в какой координатной четверти расположен треугольник, образованный прямой, заданной уравнением y=ax + b и осями координат. 5. Определите значение переменной x после выполнения фрагмента программы:   A = 53 : B = 14 : C = 2  IF A>=B THEN  IF A>=C THEN  X = A-B-C  ELSE  X=C-A-B  END IF  ELSE  X=B-A  END IF   1. Составьте программу для нахождения корней квадратного уравнения. 2. Определить, попадет ли точка с заданными координатами в область I (для простоты принять, что точка не попадает на границу этой области).   y  -2 **I**  0 x  -3    **I**   1. **Оператор выбора.**   *Цели:* познакомить учащихся с использованием оператора выбора (варианта) при программировании различных задач.  *Краткий обзор теоретического материала.* Блок-схема оператора варианта. Структура оператора варианта. Служебные слова SELECT, CASE, END SELECT.  *Примеры возможных заданий.*   1. С клавиатуры вводится какой-либо год ХХ века. Определить, мирным ли был этот год для России или она в то время участвовала в войне. Если год был военный, то выяснить, какая именно тогда была война. 2. Найти остаток от деления целого выражения С=К(А+В) на 4 и вывести сообщение о величине остатка. Если остаток равен 0, то значение переменной С оставить без изменения, если 1 или 3 – уменьшить на величину остатка, если 2 – увеличить на величину остатка. Новое значение С вывести на экран (оно делится на 4 без остатка). 3. **Программирование простых и условных вычислительных алгоритмов. Вычисление простых и условных математических выражений**.   *Цели:* программировать различные задачи с использованием условного оператора.  *Примеры возможных заданий.*   1. Составить программу, которая определяет принадлежность точки с координатами (x, y) сектору, являющемуся пересечением круга радиуса R с центром в начале координат и 1 координатной четверти. 2. Составить программу, которая в зависимости от порядкового номера дня недели (1, 2, …, 7) выводит на экран его название (понедельник, вторник, …, воскресенье). 3. Составить программу для вычисления значения функции  -1, если x < -1,  Y= x, если x < 1,  1, если x = 1. 4. **Практическая работа №2 «Ветвление».**   Смотрите приложение. | 1  1  1  1  1 |
| 12-16 |  | **Циклы (5 ч).**   1. **Оператор цикла с предусловием.**   *Цели:* познакомить учащихся с циклом с предусловием, алгоритмом его работы; программировать задачи с использованием цикла с предусловием.  *Краткий обзор теоретического материала.* Блок-схема цикла с предусловием. Тело цикла. Оператор цикла с предусловием. Служебные слова WHILE, WEND. Работа оператора цикла с предусловием.  *Примеры возможных заданий.*   1. Дано натуральное число n. Посчитать количество цифр в числе. 2. Вычислить приближенное значение суммы 1+ + + + … . Суммирование продолжать до тех пор, пока очередное слагаемое не станет меньше заданного числа H. 3. Сумма в P рублей положена в сберегательный банк. При этом прирост составляет M% ежегодно. Считая, что прирост осуществляется непрерывно, написать программу, подсчитывающую, через какой промежуток времени сумма достигнет K рублей. 4. **Оператор цикла с постусловием**.   *Цели:*  познакомить учащихся с циклом с постусловием, алгоритмом его работы; программировать задачи с использованием цикла с постусловием.  *Краткий обзор теоретического материала.* Блок-схема цикла с постусловием. Тело цикла. Оператор цикла с постусловием. Служебные слова DO, LOOP, WHILE, UNTIL. Работа оператора цикла с постусловием.  *Примеры возможных заданий.*   * + - 1. Вычислить приближенное значение суммы + + + … . Суммирование продолжать, пока очередное слагаемое (кроме первого) будет больше заданной точности вычислений.       2. Составить программу планирования закупки товара в магазине на сумму, не превышающую заданную величину.       3. Написать программу нахождения НОД двух натуральных чисел.  1. **Оператор цикла с неизвестным числом повторений (параметром).**   *Цели:* познакомить учащихся с циклом с параметром, алгоритмом его работы; программировать задачи с использованием цикла с параметром.  *Краткий обзор теоретического материала.* Блок-схема цикла с параметром. Тело цикла. Параметр цикла. Начальное и конечное значение параметра цикла. Шаг цикла. Оператор цикла с параметром. Служебные слова FOR, TO, NEXT. Работа оператора цикла с параметром.  *Примеры возможных заданий.*   * + - 1. Вычислить:       2. Составьте программу вычисления первых десяти членов последовательности, заданной формулой n-го члена: Xn = 3n -5.       3. Вычислить произведение всех двузначных нечетных чисел.       4. Определить, сколько из N заданных точек принадлежит графику Y=|X|.  1. **Вложенность циклов. Программирование циклических алгоритмов.**   *Цели*: дать понятие о вложенных циклах и особенностях их выполнения*,* научить программировать задачи с использованием вложенных циклов, задачи на сочетание цикла и условного оператора.  *Краткий обзор теоретического материала.* Конструкция вложенных циклов. Конструкция сочетания цикла и условного оператора.  *Примеры возможных заданий.*   * + - 1. Даны натуральные числа n и k. Составить программу вычисления выражения   1k + 2k + … + nk.   * + - 1. *Старинная задача.* Сколько можно купить быков, коров и телят, если плата за быка 10 рублей, за корову – 5 рублей, за теленка – полтинник (0,5 рубля), если на 100 рублей надо купить 100 голов скота.       2. Написать программу, которая находит и выводит на печать все четырехзначные числа abcd, a, b, c, d – различные цифры, для которых выполняется : ab –cd = a + b + c + d.       3. Вывести на экран все целые числа от 100 до 200, кратные 3.  1. **Практическая работа №3 «Циклы».**   Смотрите приложение. | 1  1  1  1  1 |
| 17-23 |  | **Массивы (7 ч).**   1. **Одномерные массивы.**   *Цели:* познакомить учащихся с понятием одномерного массива, рассмотреть способы задания одномерных массивов; учить программировать задачи с использованием одномерных массивов.  *Краткий обзор теоретического материала.* Понятие массива. Понятие индекса. Переменная с индексом. Простая переменная. Одномерные массивы. Описание массива:  DIM имя массива (n1 TO n2) AS тип элементов. Присвоение значений элементам массива. Способы задания одномерных массивов.  *Примеры возможных заданий.*   * + - 1. Опишите, как организовать вывод линейного массива в строку? В столбец?       2. Заполнить массив чисел так, чтобы          1. значения элементов совпадали с их индексами;          2. значениями элементов оказались случайные числа на интервале [111, 555];          3. каждый элемент вычислялся по формуле b( i ) =       3. Сформировать одномерный массив из N элементов, элементы которого целые случайные числа в пределах от 1 до 45. Напечатать элементы массива в прямом и обратном порядке.       4. Дана последовательность Х0=6765, Х1 – неизвестно, Хk = Хk-1 + Хk-2 (k = 2, 3, …, 20). Найти последовательность Х0, Х1, Х2, …, Х20, если известно, что Х20= 0.   **Двумерные массивы.**  *Цели:* познакомить учащихся с понятием двумерного массива, рассмотреть способы задания двумерных массивов; учить программировать задачи с использованием двумерных массивов.  *Краткий обзор теоретического материала.* Понятие матрицы. Двумерные массивы. Нумерация элементов двумерного массива. Способы описания двумерного массива. Способы задания двумерных массивов.  *Примеры возможных заданий.*   1. Определить, является ли данный квадратный массив симметричным относительно своей главной диагонали. 2. Заполнить массив A размером m x n в виде «змейки» (Например, если n=5, m=5, то массив должен быть следующим:   1 2 3 4 5  10 9 8 7 6  11 12 13 14 15  20 19 18 17 16  21 22 23 24 25 ).   1. Составить программу, запрашивающую координаты ферзя на шахматной доске и показывающую поля доски, находящиеся под боем. 2. **Поиск экстремальных значений величин в одномерных и двумерных массивах чисел.**   *Цели:* программировать задачи на поиск экстремальных значений величин в одномерных и двумерных массивах чисел.  *Краткий обзор теоретического материала.* Проверка на четность. Подсчет количества элементов, отвечающих заданным условиям. Нахождение суммы, произведения и количества элементов массива, отвечающих заданным условиям. Максимальный и минимальный элементы.  *Примеры возможных заданий.*   * + - 1. Найти максимальный (минимальный) элемент одномерного (двумерного) массива, а также его порядковый номер.       2. Посчитать количество, (сумму, произведение) отрицательных и неотрицательных (четных, нечетных) элементов одномерного (двумерного) массива.       3. Найти количество элементов, больших некоторого заданного числа Х в каждой строке двумерного массива (во всем массиве).  1. **Перестановка элементов массива. Сортировка массива.**   *Цели:* программировать задачи, на перестановку и сортировку элементов массива.  *Краткий обзор теоретического материала.* Сортировка элементов массива. Метод «пузырька». Оператор SWAP. Перестановка элементов массива.  *Примеры возможных заданий.*   * + - 1. Отсортировать одномерный массив в порядке возрастания (убывания).       2. Сформировать одномерный массив из N элементов, элементы которого целые случайные числа в пределах от 1 до 45. Переставить элементы, стоящие на нечетных местах, с соответствующими элементами на четных местах.       3. Дан двумерный массив.          1. Поменять местами первую и третью строки.          2. Поменять местами второй и последний столбцы.  1. **Слияние и отбор данных в одномерных и двумерных массивах.**   *Цели:* программировать задачи на слияние и отбор данных (по определенному условию) в одномерных и двумерных массивах чисел.  *Краткий обзор теоретического материала.* Слияние и отбор данных в одномерных и двумерных массивах.  *Примеры возможных заданий.*   1. Произвести удаление (вставку) элемента одномерного массива, заданного датчиком случайных чисел. 2. Вставить строку из нулей на место строки с номером k двумерного массива. 3. Удалить строку с номером k двумерного массива. 4. В массиве A(N, M) удалить все строки, сумма значений элементов которых отрицательна, и вставить пред последней строкой строку, значения элементов равны 1. 5. **Поиск, подбор и группировка данных.**   *Цели:* программировать задачи на поиск, подбор и группировку данных (по определенному условию) в одномерных и двумерных массивах чисел.  *Краткий обзор теоретического материала.* Поиск, подбор и группировка данных.  *Примеры возможных заданий.*   1. Дан массив A(N), сформированный датчиком случайных чисел на интервале (-50; 150). Из элементов массива А, кратных 5, сформировать массив В. 2. Дан двумерный массив размером 4х5, заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен количеству отрицательных элементов, кратных 3 или 5, соответствующей строки. 3. **Практическая работа №4 «Массивы».**   Смотрите приложение. | 1  1  1  1  1  1  1 |
| 24-27 |  | **Строковый, символьный тип данных (4 ч).**   1. **Строковый, символьный тип данных. Основные операции.**   *Цели:* познакомить учащихся с основными сведениями о строковом, символьном типе данных, с операциями над символьными величинами.  *Краткий обзор теоретического материала.* Основные сведения о символьных величинах. Стандартные функции, процедуры для работы с символьными величинами: сравнение, конкатенация, копирование, удаление, замена (вставка), длина строки, подстрока числа и строки.  *Примеры возможных заданий.*   * + - 1. Распечатайте символы, соответствующие различным кодам, и одновременно определите, какой максимальный код можно использовать.       2. Вывести слово и:          1. Напечатать его по вертикали;          2. Переставить буквы слова в обратном порядке.       3. Составить программу «простейший справочник».       4. Даны две символьные строки А и В. Определите, в какой из них больше символов и на сколько.  1. **Операции поиска и замены в символьных строках и массивах.**   *Цели:* программировать задачи на поиск и замену элементов в символьных строках и массивах.  *Краткий обзор теоретического материала.* Удаление, замена (вставка), длина строки, подстрока числа и строки.  *Примеры возможных заданий.*   1. Определите, сколько раз в данном тексте встречается каждый символ (Один и тот же символ дважды не использовать). 2. Составить программу, заменяющую в данном тексте одну подстроку на другую. 3. Реализовать на компьютере возможность автоматического исправления слова <граматика> (правильно <грамматика> ). 4. **Программирование алгоритмов обработки текста.**   *Цели:* программировать задачи на обработку, шифровку и дешифровку текста.  *Краткий обзор теоретического материала.* Понятие шифровки, дешифровки текста. Способы шифровки текста.  *Примеры возможных заданий.*   * + - 1. Дана строка, состоящая из нескольких слов, между словами стоит один пробел, в конце предложения - точка. Посчитать количество слов и вывести на экран только те из них, которые начинаются с буквы «А».       2. Составить программу устранения лишних пробелов в тексте (Лишним будем считать каждый пробел, следующий непосредственно за пробелом).       3. Определить, является ли данная фраза палиндромом       4. Заменить в тексте все символы их кодами.  1. **Практическая работа №5 «Обработка текста»**   Смотрите приложение. | 1  1  1  1 |
| 28-30 |  | **Подпрограммы (3 ч).**   1. **Подпрограммы (функции и процедуры). Назначение. Способы описания.**   *Цели:* дать учащимся представление о подпрограммах и возможностях их использования. Показать на примерах механизм реализации подпрограмм с помощью процедур и функций.  *Краткий обзор теоретического материала.* Понятие подпрограммы, процедуры, функции, Описание процедуры. Оператор SUB. Описание функции. Оператор FUNCTION. Механизм реализации подпрограмм с помощью процедур и функций.Обмен информацией между основной программой и подпрограммой. Глобальные и локальные переменные.  *Примеры возможных заданий.*   * + - 1. Составить процедуру сложения двух чисел, вводимых с клавиатуры.       2. Упорядочить значения трех переменных a, b, c используя подпрограмму упорядочивания двух переменных.       3. Написать функцию, подсчитывающую количество цифр целого числа. Используя ее, определить, в каком из двух чисел больше цифр.       4. Напишите функцию для нахождения НОД двух чисел с помощью алгоритма Евклида и используйте ее в программе для нахождения НОД уже N чисел.  1. **Рекурсия.**   *Цели:* дать учащимся представление о рекурсии и возможностях ее использования.  *Краткий обзор теоретического материала.* Понятие рекурсии. Рекурсивные алгоритмы. Задачи, сводимые к рекурсивным.  *Примеры возможных заданий.*   * + - 1. Перевести натуральное число из десятичной системы счисления в двоичную.       2. Определить, является ли заданное натуральное число простым.       3. Вычислить факториал натурального числа.       4. Найти НОД двух натуральных чисел.  1. **Практическая работа №6 «Подпрограммы»**   Смотрите приложение. | 1  1  1 |
| 31-33 |  | **Работа с файлами (3 ч).**   1. **Файлы. Текстовые файлы. Файлы с фиксированной структурой записи.**   *Цели:* дать учащимся основные сведения о файлах и их классификации; познакомить с операциями над файлами; программировать задачи с использованием файлов.  *Краткий обзор теоретического материала.* Понятие файла. Классификация файлов. Операции над файлами: открытие файла, чтение и запись обрабатываемых данных, закрытие файлов.  *Примеры возможных заданий.*   * + - 1. Открыть файл f86.txt и сохранить в нем N целых чисел в пределах от 65 до 90:          1. Считайте информацию из данного файла;          2. Найдите сумму элементов, находящихся в файле f86.txt.       2. Создать файл, элементы которого вычисляются по формуле M=I\*I+4\*I; I=1,2, … N.       3. Создать текстовый файл, содержащий только целые числа, в каждой строке может быть несколько чисел, которые разделяются пробелами. Вывести на экран все числа с учетом разбиения на строки и подсчитать количество элементов в каждой строке.       4. Дан список данных о группе баскетболистов с указанием имени, роста, числа забитых мячей. Создайте файл, содержащий информацию о баскетболистах. Определите число записей в файле.  1. **Процедуры и функции для работы с файлами.**   *Цели:* дать учащимся основные сведения о процедурах и функциях работы с файлами; программировать задачи с использованием файлов.  *Краткий обзор теоретического материала.* Файл произвольного доступа. Операторы и функции работы с файлом произвольного доступа.  *Примеры возможных заданий.*   * + - 1. Создайте файл прямого доступа, элементами которого являются целые случайные числа в диапазоне от -35до 30. Число элементов в файле нечетно. Установив указатель позиции на первый, средний и последний элемент, считайте данные из файла и найдите их сумму.       2. Создать файл, элементами которого являются целые случайные числа. Упорядочить по убыванию (возрастанию) элементы файла.  1. **Практическая работа №7 «Работа с файлами».**   Смотрите приложение. | 1  1  1 |
| 34 |  | **Повторение, решение задач (1 ч).**  *Цели:* повторить основные разделы программирования; решать задачи.  *Примечание.* На данном уроке можно разобрать олимпиадные задачи по программированию.  *Примеры возможных заданий.*   1. Вася и Петя учатся в школе со следующим расписанием занятий: уроки начинаются в 8:30, каждый урок длится 45 минут, а между уроками есть пятнадцатиминутная перемена. Сейчас идет K-й урок по счету и Вася, уставший после напряженного учебного дня, поскорее хочет домой и спрашивает у Пети, сколько времени. Петя отвечает «Т минут», если с начала урока прошло Т минут и «минус Т минут», если до конца урока осталось Т минут. Составьте программу, которая по известному номеру урока, а также ответу Пети точно определит, сколько сейчас времени. (С клавиатуры вводится два числа через пробел– К и Т (1 ≤ К6, -45≤ Т ≤ 45), где К – номер текущего урока, Т – время, которое назвал Петя.) 2. Один из самых простых шифров – это шифр Цезаря. Его суть – скачек через две буквы алфавита в направлении от его начала к его концу: любая k-я буква, кроме последней и предпоследней, заменяется на (k+2)-ю букву. Предпоследняя буква заменяется на первую, а последняя – на вторую букву.   Составить программу зашифровки и расшифровки русского текста шифром Цезаря.   1. Бесконечная в обе стороны последовательность чисел fi, в которой любой целый элемент fi+2 вычисляется, используя следующее условие fi+2=fi+1+fi.   Пусть заданы два различных члена этой последовательности f i и fj с соответствующими номерами i и j, а также некое целое число n. Необходимо восстановить элемент этой последовательности fn, соответствующий номеру n.   1. Маленький мальчик взял лист бумаги NxN клеток и нарисовал на нем замкнутую m-звенную ломанную с вершинами в узлах клеток. После этого он выписал квадраты длин звеньев в порядке их обхода по ломанной и выкинул рисунок.   Определить, существует ли хотя бы одна ломанная, соответствующая данным, записанным мальчиком  1<=n<=15  3<=m<=20 | 1 |
|  |  | **ИТОГО** | **34** |

**Список рекомендуемой литературы**

1. Чернов А.А. Информатика. Конспекты уроков. 9-11 классы. Практикум по программированию. Волгоград: Учитель, 2006г.
2. В.С.Новичков, А.Н.Пылькин. Начала программирования на языке QBASIC.М.: Горячая линия-Телеком, 2007 год.
3. Златопольский Д.М. Я иду на урок информатики. Задачи по программированию. 7-11 классы. Москва «1сентября», 2001 год.
4. Босова Л.Л., Розова В.М., Семенова Е.Е. Разноуровневые дидактические материалы по информатике. Москва «Образование и информатика», 2001 год.
5. Житкова О.А., Кудрявцева Е.К. Тематический контроль по информатике. Бейсик и Паскаль в вопросах и задачах. Рабочая тетрадь 1, 2. «Интеллект – центр» Москва, 2002 г.
6. Турчак Л.И. Основы численных методов: Учеб. пособие.- М.: Наука; Гл. ред. физ.-мат. лит., 1987.- 320 с.
7. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль.- Томск: МП “Раско”, 1991.- 272 с.
8. Тынкевич М.А. Численные методы анализа: Учеб. пособие.- Кемерово, 1997.- 123 с.
9. Абрамов С.А. Гнездилов Г.Г. и др. «Задачи по программированию». М.: НАУКА,1988г.
10. Филиппов С.В. «Занимательный Бейсик», М.: Издательство ЭКОНОМ, 1998 г.
11. Мельникова О.И. Бенюшкина А.Ю. «Начала программирования на языке «Бейсик» для персональных ЭВМ»
12. Зельдер Г.А. «Программируем на языке QUICK BASIC 4.5», М.: ABF, 1997г.