Муниципальное казенное общеобразовательное учреждение
«Лебяжьевская средняя общеобразовательная школа»

РАССМОТРЕНА	ПРИНЯТА	УТВЕРЖДЕНА
на заседании методического совета	на заседании	Приказом № 57/1
Протокол № 1	Педагогического Совета.	от «20» 05 2017 г.
от «12» 05 2017 г.	Протокол № 9 от «19» 05 2017 г.	Директор школы: Н.В. Гончарова

РАБОЧАЯ ПРОГРАММА
ЭЛЕКТИВНОГО КУРСА
«АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»
10 КЛАСС

ЛЕБЯЖЬЕ – 2017

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ 10 КЛАСС

Е.Г. Квашнин,
г. Курган, доцент кафедры ЕМО, ИПКиПРО

Пояснительная записка

Основной **целью** изучения элективного курса «Алгоритмизация и программирование» является знакомство обучающихся с применением современных методов информатики для решения математических задач, а также с математическими методами, используемыми в информатике.

Задачи курса:

1. Изучение алгоритмического языка программирования Turbo Pascal.
2. Создание проектов, используя данные языки программирования.
3. Формирование исследовательских умений и навыков обучающихся.

На изучение курса отводится 34 часа (1 час в неделю).

В 10 классе программирование изучается на языке Turbo Pascal, это связано с его чёткой логической структурой и с возможностями, которые позволяют использовать Turbo Pascal для решения разнообразных задач. Среди них вычисления и обработка данных, компьютерная графика, работа со звуком, системное программирование. Turbo Pascal позволяет применять приёмы объектно-ориентированного программирования – одной из ведущих современных компьютерных технологий.

Содержание обучения
10 класс

Алгоритмическое программирование на языке Turbo Pascal (34 ч)

Среда программирования (реализация основных способов организации действий в языке программирования). Типы данных. Основные операторы, функции, арифметические операции… Линейные алгоритмы. Понятие ветвления. Применение алгоритмов с ветвлением. Понятие цикла в форме «Пока», «До тех пор», «Для каждого». Применение циклических алгоритмов. Понятие массива и его элемента. Операции над массивами. Применение массивов при решении задач. Понятие вспомогательного алгоритма, заголовка, аргументов и результатов вспомогательного алгоритма.

Обучающиеся должны знать:

- интегрированную среду Turbo Pascal (рабочая область редактора, строка меню, строка статуса, главное меню…);
- назначение главных пунктов меню;
- основные операторы, функции, арифметические операции, типы данных;
- определение линейного алгоритма;
- определение двух форм ветвления: полной (имеющей две части) и неполной (имеющей одну ветвь);
- определение цикла и три его формы;
- определение массива, обозначения элементов массива;
- основные операции, выполняемые над массивами;
- определение вспомогательного алгоритма как произвольного алгоритма, снабжённого заголовком, позволяющим вызывать этот алгоритм из других алгоритмов.

Обучающиеся должны понимать, что:

- ветвление в алгоритмах появляется тогда, когда исполнителю необходимо сделать выбор одного или несколько наборов действий в зависимости от некоторого условия;
- проверка этого условия должна являться допустимым действием исполнителя;
- при записи ветвлений необходим указатель конца ветвления, отделяющий ветвление от остальной части программы (при отсутствии такого указателя алгоритм становится двусмысленным);
- появление циклов в алгоритме обусловлено необходимостью повторять определённый набор действий до тех пор, пока выполняется некоторое условие;
- условие продолжения цикла проверяется только перед очередным выполнением тела цикла; исполнение прекращается лишь в том случае, если к моменту очередного выполнения тела цикла условие оказывается нарушенным; по ходу исполнения тела цикла условие может нарушиться, но это не вызовет прекращения исполнения тела цикла;
- при записи цикла необходим указатель конца цикла, отделяющий тело цикла от остальной программы;
- в алгоритмах обработки массивов целесообразно применять цикл «Для каждого…» (поскольку в таких случаях обычно заранее известно число повторений тела цикла);
- в роли вспомогательного может выступать любой алгоритм, если его снабдить соответствующим заголовком;
- в заголовке нужно указать название, аргументы (имена тех переменных, значения которых передаются вспомогательному алгоритму из основного) и результаты (имена тех переменных, значения которых передаются из вспомогательного алгоритма основному);
- создание вспомогательного алгоритма равносильно для исполнителя добавлению ещё одного его допустимого действия: в результате выделения вспомогательного алгоритма подробные объяснения того, что нужно делать, можно заменить одной командой.

Обучающиеся должны уметь:

- использовать пункты главного меню для выполнения действий (хранение программы, создание нового файла…);
- записывать линейные алгоритмы;
- записывать разветвляющиеся алгоритмы, не допуская двусмысленности записи (от учащихся не требуется строго соблюдения какой – либо жёстко фиксированной формы записи, но требование отсутствия двусмысленности обязательно: в частности, из записи алгоритма должно быть понятно, где начинается и кончается ветвление);
- записывать циклические алгоритмы, не допуская двусмысленности записи (так, из записи алгоритма должно быть понятно, из каких действий состоит тело цикла, где начинается и заканчивается цикл);
- использовать массивы при составлении программ;
- записывать вспомогательные алгоритмы в виде подпрограмм, понятных исполнителям, имитируемым на компьютере;
- использовать простейшие приёмы отладки разветвляющихся и циклических программ, а также программ, содержащих подпрограммы;
- применять ветвления, циклы, массивы и вспомогательные алгоритмы при решении задач и создании проектов.

Тематическое планирование

№	Тема занятия	Кол-во часов	Образовательные задачи (должны знать и уметь)
10 класс Тема 1. Алгоритмическое программирование на языке Turbo Pascal			
1	Среда программирования	1	*Обучающиеся должны знать:* - назначение программирования;

	(реализация основных способов организации действий в языке программирования)		- интегрированную среду Turbo Pascal (рабочая область редактора, строка меню, строка статуса, главное меню…). *Обучающиеся должны уметь:* - использовать пункты главного меню для выполнения действий (хранение программы, создание нового файла…).
2	Среда программирования (реализация основных способов организации действий в языке программирования)	1	*Обучающиеся должны знать:* - интегрированную среду Turbo Pascal (рабочая область редактора, строка меню, строка статуса, главное меню…); назначение главных пунктов меню. *Обучающиеся должны понимать:* - принципы работы программы, написанной на языке программирования (назначение транслятора). *Обучающиеся должны уметь:* - использовать пункты главного меню для выполнения действий (хранение программы, создание нового файла…).
3	Типы данных	1	*Обучающиеся должны знать:* - основные операторы, функции, арифметические операции, типы данных. *Обучающиеся должны уметь:* - записывать блок «описания данных».
4-6	Основные операторы, функции, арифметические операции	3	*Обучающиеся должны знать:* - основные операторы, функции, арифметические операции, типы данных. *Обучающиеся должны понимать, что:* - изменение хотя бы одного символа в служебных словах (операторах) языка программирования Turbo Pascal приводит к невыполнению программы. *Обучающиеся должны уметь:* - записывать линейные алгоритмы.
7-8	Линейные алгоритмы	2	*Обучающиеся должны знать:* - основные операторы, функции, арифметические операции, типы данных; - определение линейного алгоритма. *Обучающиеся должны уметь:* - записывать линейные алгоритмы.
9	Понятие ветвления	1	*Обучающиеся должны знать:* - определение двух форм ветвления: полной (имеющей две части) и неполной (имеющей одну ветвь). *Обучающиеся должны понимать, что:* - ветвление в алгоритмах появляется тогда, когда исполнителю необходимо сделать выбор одного или несколько наборов действий в зависимости от некоторого условия. *Обучающиеся должны уметь:* - записывать разветвляющиеся алгоритмы, не допуская двусмысленности записи (от учащихся не требуется строго соблюдения какой – либо жёстко фиксированной формы записи, но требование

			отсутствия двусмысленности обязательно: в частности, из записи алгоритма должно быть понятно, где начинается и кончается ветвление).
10-12	Применение алгоритмов с ветвлением	3	*Обучающиеся должны знать:* - определение двух форм ветвления: полной (имеющей две части) и неполной (имеющей одну ветвь). *Обучающиеся должны понимать, что:* - ветвление в алгоритмах появляется тогда, когда исполнителю необходимо сделать выбор одного или несколько наборов действий в зависимости от некоторого условия; - проверка этого условия должна являться допустимым действием исполнителя; - при записи ветвлений необходим указатель конца ветвления, отделяющий ветвление от остальной части программы (при отсутствии такого указателя алгоритм становится двусмысленным). *Обучающиеся должны уметь:* - записывать разветвляющиеся алгоритмы, не допуская двусмысленности записи (от учащихся не требуется строго соблюдения какой – либо жёстко фиксированной формы записи, но требование отсутствия двусмысленности обязательно: в частности, из записи алгоритма должно быть понятно, где начинается и кончается ветвление).
13-16	Понятие цикла в форме «Пока», «До тех пор», «Для каждого»	4	*Обучающиеся должны знать:* - определение цикла и три его формы. *Обучающиеся должны понимать, что:* - появление циклов в алгоритме обусловлено необходимостью повторять определённый набор действий до тех пор, пока выполняется некоторое условие; - условие продолжения цикла проверяется только перед очередным выполнением тела цикла; исполнение прекращается лишь в том случае, если к моменту очередного выполнения тела цикла условие оказывается нарушенным; по ходу исполнения тела цикла условие может нарушиться, но это не вызовет прекращения исполнения тела цикла; - при записи цикла необходим указатель конца цикла, отделяющий тело цикла от остальной программы. *Обучающиеся должны уметь:* - записывать циклические алгоритмы, не допуская двусмысленности записи (так, из записи алгоритма должно быть понятно, из каких действий состоит тело цикла, где начинается и заканчивается цикл).
17-19	Применение циклических алгоритмов	3	*Обучающиеся должны знать:* - определение цикла и три его формы. *Обучающиеся должны понимать, что:* - появление циклов в алгоритме обусловлено необходимостью повторять определённый набор

			действий до тех пор, пока выполняется некоторое условие; - условие продолжения цикла проверяется только перед очередным выполнением тела цикла; исполнение прекращается лишь в том случае, если к моменту очередного выполнения тела цикла условие оказывается нарушенным; по ходу исполнения тела цикла условие может нарушиться, но это не вызовет прекращения исполнения тела цикла; - при записи цикла необходим указатель конца цикла, отделяющий тело цикла от остальной программы. *Обучающиеся должны уметь:* - записывать циклические алгоритмы, не допуская двусмысленности записи (так, из записи алгоритма должно быть понятно, из каких действий состоит тело цикла, где начинается и заканчивается цикл).
20	Понятие массива и его элемента	1	*Обучающиеся должны знать:* - определение массива, обозначения элементов массива. *Обучающиеся должны понимать, что:* - в алгоритмах обработки массивов целесообразно применять цикл «Для каждого…» (поскольку в таких случаях обычно заранее известно число повторений тела цикла). *Обучающиеся должны уметь:* - использовать массивы при составлении программ.
21	Операции над массивами	1	*Обучающиеся должны знать:* - определение массива, обозначения элементов массива; - основные операции, выполняемые над массивами. *Обучающиеся должны понимать, что:* - в алгоритмах обработки массивов целесообразно применять цикл «Для каждого…» (поскольку в таких случаях обычно заранее известно число повторений тела цикла). *Обучающиеся должны уметь:* - использовать массивы при составлении программ.
22-23	Применение массивов при решении задач	2	*Обучающиеся должны знать:* - определение массива, обозначения элементов массива; - основные операции, выполняемые над массивами. *Обучающиеся должны понимать, что:* - в алгоритмах обработки массивов целесообразно применять цикл «Для каждого…» (поскольку в таких случаях обычно заранее известно число повторений тела цикла). *Обучающиеся должны уметь:* - использовать массивы при составлении программ.
24-26	Понятие вспомогательного алгоритма, заголовка,	3	*Обучающиеся должны знать:* - в роли вспомогательного может выступать любой алгоритм, если его снабдить соответствующим заголовком;

	аргументов и результатов вспомогательного алгоритма		- в заголовке нужно указать название, аргументы (имена тех переменных, значения которых передаются вспомогательному алгоритму из основного) и результаты (имена тех переменных, значения которых передаются из вспомогательного алгоритма основному); - создание вспомогательного алгоритма равносильно для исполнителя добавлению ещё одного его допустимого действия: в результате выделения вспомогательного алгоритма подробные объяснения того, что нужно делать, можно заменить одной командой. *Обучающиеся должны понимать, что:* - в роли вспомогательного может выступать любой алгоритм, если его снабдить соответствующим заголовком; - в заголовке нужно указать название, аргументы (имена тех переменных, значения которых передаются вспомогательному алгоритму из основного) и результаты (имена тех переменных, значения которых передаются из вспомогательного алгоритма основному); - создание вспомогательного алгоритма равносильно для исполнителя добавлению ещё одного его допустимого действия: в результате выделения вспомогательного алгоритма подробные объяснения того, что нужно делать, можно заменить одной командой. *Обучающиеся должны уметь:* - записывать вспомогательные алгоритмы в виде подпрограмм, понятных исполнителям, имитируемым на компьютере; - использовать простейшие приёмы отладки разветвляющихся и циклических программ, а также программ, содержащих подпрограммы.
27-33	Подготовка проекта	7	*Обучающиеся должны знать:* - основные операторы языка программирования Turbo Pascal; - основные алгоритмические конструкции. *Обучающиеся должны понимать:* - цели, задачи и последовательность создания проекта. *Обучающиеся должны уметь:* - использовать простейшие приёмы отладки разветвляющихся и циклических программ, а также программ, содержащих подпрограммы; - применять ветвления, циклы, массивы и вспомогательные алгоритмы при решении задач и создании проектов.
34	Защита проекта	1	

МЕТОДЫ ОБУЧЕНИЯ

Основными методами обучения в данном элективном курсе являются практические методы выполнении заданий практикума. Практическая деятельность позволяет развить исследовательские и творческие способности учащихся, а также отработать основные умения. Роль учителя состоит в кратком по времени объяснении нового материала и постановке задачи, а затем консультировании учащихся в процессе выполнения практического задания.

ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНЫХ ЗАНЯТИЙ

Учебно-методический комплект предусматривает организацию учебного процесса в двух взаимосвязанных и взаимодополняющих формах:

- урочная форма, в которой учитель объясняет новый материал и консультирует учащихся в процессе выполнения ими практических заданий на компьютере;
- внеурочная форма, в которой учащиеся после уроков (дома или в школьном компьютерном классе) самостоятельно выполняют задания.

МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ:

1. Компьютерный класс с операционной системой Windows-XP и программным обеспечением Microsoft Office, Turbo Pascal;
2. Локальная компьютерная сеть;
3. Глобальная сеть Интернет;
4. Видео-проектор, экран.

СПИСОК ЛИТЕРАТУРЫ ДЛЯ ОБУЧАЮЩИХСЯ

1. Л. З. Шауцукова. Информатика: Учебное пособие для 10 – 11 классов общеобразовательных учреждений. – М.: Просвещение, 2002.
2. А. Горячев, Ю. Шафрин. Практикум по информационным технологиям. – М.: Лаборатория Базовых Знаний, 2002.
3. С. Немнюгин, Л. Перколаб. Изучаем Turbo Pascal. – СПб.: Питер, 2002.

СПИСОК ЛИТЕРАТУРЫ ДЛЯ УЧИТЕЛЯ

1. Л. З. Шауцукова. Информатика: Учебное пособие для 10 – 11 классов общеобразовательных учреждений. – М.: Просвещение, 2002.
2. А. Горячев, Ю. Шафрин. Практикум по информационным технологиям. – М.: Лаборатория Базовых Знаний, 2002.
3. С. Немнюгин, Л. Перколаб. Изучаем Turbo Pascal. – СПб.: Питер, 2002.