

Министерство образования Московской области Государственное бюджетное профессиональное
образовательное
учреждение Московской области
«ОРЕХОВО-ЗУЕВСКИЙ ЖЕЛЕЗНОДОРОЖНЫЙ ТЕХНИКУМ
ИМЕНИ В.И. БОНДАРЕНКО»

Специальность
«Программирование в компьютерных системах»

09.02.03

ОП.05 Основы программирования

Контрольно-оценочные средства

Преподаватель
Гусев Иван Евгеньевич

Орехово-Зуево 2016 г.

Рассмотрена на заседании комиссии профессионального цикла «Программирование в компьютерных системах», математических и профильных учебных дисциплин

Соответствует требованиям ФГОС СПО по специальности Программирование в компьютерных системах

Протокол № от «__» _____ 201_ г.
Председатель комиссии

Зам. директора по учебно-воспитательной работе

_____/ /

_____/ /

«__» _____ 201_ г.

Организация-разработчик ГБПОУ МО ОРЕХОВО-ЗУЕВСКИЙ
ЖЕЛЕЗНОДОРОЖНЫЙ ТЕХНИКУМ ИМЕНИ В.И. БОНДАРЕНКО

Разработчик: преподаватель общепрофессиональных дисциплин Гусев И.Е.

1. Общие положения

Контрольно-оценочные средства (КОС) являются составной частью образовательной программы среднего профессионального образования по подготовке специалистов среднего звена 09.02.03 *Программирование в компьютерных системах* и предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины ОП.05 «Основы программирования».

КОС включают контрольные материалы для проведения промежуточной аттестации в форме экзамена.

КОС разработаны на основании:

Положения о Фонде оценочных средств (ФОС);

Рекомендаций по разработке контрольно-оценочных средств (КОС);

рабочей программы учебной дисциплины.

2. Результаты освоения дисциплины, подлежащие проверке

КОС для промежуточной аттестации направлены на проверку и оценивание результатов обучения, знаний и умений:

Результаты освоения дисциплины «Основы алгоритмизации»			
Результаты обучения (освоенные умения, усвоенные знания)	Коды формируемых профессиональных и общих компетенций	Основные показатели оценки	№ заданий, включенных в КОС
У1 Работать в среде программирования	ОК 1 – ОК 10 ПК 1.1, ПК 1.3, ПК 1.2	Уметь правильно применять полученные знания в использовании средств систем программирования	ЛР. 1-23
У2 Реализовывать построенные алгоритмы в виде программ на корректном языке программирования	ОК 1 – ОК 10 ПК 1.1, ПК 1.3, ПК 1.2	Уметь использовать системы программирования с компиляторами и интерпретаторами	ЛР. 1-9
З1 Этапы решения задачи на компьютере	ОК 1 – ОК 10 ПК 1.1, ПК 1.3, ПК 1.2	Знать понятия: постановки задачи, блок-схемы, алгоритма, программирования и тестирования	ЛР. 24-27
З2 Типы данных	ОК 1 – ОК 10 ПК 1.3, ПК 1.5, ПК 1.4	Хорошее знание типов данных и описание их в программах	ЛР. 28,29
З3 Принципы структурного и модульного программирования	ОК 1 – ОК 10 ПК 1.1, ПК 1.3, ПК 1.2	Знать методы отладки программы на уровне модуля по сценарию	ЛР. 10
З4 Принципы объектно-ориентированного программирования	ОК 1 – ОК 10 ПК 1.1, ПК 1.3, ПК 1.2	Знать основные принципы технологии структурного и объектно-ориентированного программирования	ЛР. 30-37

3. Распределение КОС по темам учебной дисциплины

Контрольно-оценочные средства представляют собой перечень заданий различного уровня трудности (от 1 до 3).

ПАСПОРТ ОЦЕНОЧНОГО СРЕДСТВА

«Основы программирования»			
Наименование тем	Кол-во часов	Порядковый номер задания	Уровень трудности **
Тема 1.1 Основы алгоритмизации.	40	1	1
Тема 1.2 Языки программирования	16	10	1
Тема 1.3. Типы данных	6		
Тема 2.1 Операторы языка программирования	12	9	2
		11	2
		16	2
Тема 2.2 Условный оператор	10	2	2
		3	2
		4	2
		5	2
		6	3
		7	2
		8	2
Тема 2.3 Операторы цикла	17		
Тема 3.1 Процедуры и функции	10	20	2
		21	2
		22	2
		23	2
Тема 3.2. Структуризация в программировании	6	15	2
		17	2
		18	2
		26	2
		27	2
Тема 3.3. Модульное программирование	15	28	2
		12	2
		14	2
		29	2
		30	2
		31	2
		32	3
		33	3
Раздел 4. Структуры данных			
Тема 4.1. Массивы	22		
Тема 4.2. Строки	16		
Тема 4.3. Множества	6		
Тема 4.4. Записи	4		
Тема 4.5. Файлы	20		
Тема 4.6. Указатели	14		
Раздел 5. Объектно-ориентированное программирование			
Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)	9		
Тема 5.2 Интегрированная среда разработчика	8		
Тема 5.3 Этапы разработки приложения	8		
Тема 5.4 Иерархия классов	16		

Тема 5.5. Визуальное событийно-управляемое программирование	18		
Тема 5.6. Разработка оконного приложения	12		

**

Низкий -1

Средний -2

Высокий-3

4. Содержание КОС

Содержание банка КОС в полной мере отражает требования ФГОС по специальности и содержания рабочей программы учебной дисциплины. В состав банка включены теоретические вопросы и практические задания.

Форма текущего контроля - ЭКЗАМЕН

Критерии оценки знаний студента

Оценка «отлично» ставится за краткий, четкий, обоснованный ответ на теоретический вопрос и правильно решенную задачу.

Оценка «хорошо» ставится за теоретический вопрос, освещенный не полностью и не совсем точно и правильно решенную задачу.

Оценка «удовлетворительно» ставится за правильно решенную задачу и наводящие вопросы преподавателя по теоретическому вопросу.

Оценка «неудовлетворительно» ставится за не решенную задачу.

Перечень вопросов и практических задач по разделам и темам:

№ раздела и тем	Перечень вопросов	Перечень практических задач
1	2	3
Раздел 1.	Основы алгоритмизации	
Тема 1.1.	Алгоритмы, их свойства, алгоритмический язык Алгоритмический язык как средство выражения и записи алгоритмов. Алгоритмы, их свойства, исполнители алгоритмов. Линейный алгоритм, их свойства, способы описания, примеры. Разветвляющийся алгоритм, полная и неполная ветка. Циклический алгоритм, цикл, тело цикла.	Записать алгоритм одним из трех способов: словесным, табличным, блок-схемой
Тема 1.2.	Языки программирования Языки программирования, их классификация. Системы программирования Способы описаний языков программирования. Нотация Бекуса и IBM Виды программирования. Принципы построения ПО. Трансляторы. Компиляторы и интерпретаторы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.	Написать простейший линейный, разветвляющийся, циклический алгоритм
Тема 1.3.	Типы данных Величины. Объявление объектов данных. Внутреннее представление данных в памяти компьютера. Типы данных. Простые типы данных. Производные типа	Пояснить работу команды вызова вспомогательного

	данных. Структурированное типы данных.	алгоритма Формально исполнить алгоритм по таблице значений
Раздел 2.	Основные конструкции языков программирования	
Тема 2.1.	Операторы языка программирования Синтаксис языка. Арифметические выражения. Стандартные функции Структура программы. Ввод и вывод данных. Оператор присваивания.	Подготовить сообщение по 1 из языков программирования, напечатать схему решение задач по эвм
Тема 2.2.	Условный оператор Составной оператор. Условный оператор . Оператор выбора. Написание программ с использованием условного оператора.	
Тема 2.3.	Операторы цикла Цикл с постусловием. Цикл с предусловием. Цикл с параметром Вложенные циклы. Тестированием по темам 2.1 - 2.3	
Раздел 3	Структурное и модульное программирование	
Тема 3.1.	Процедуры и функции Общие сведения о подпрограммах. Определение и вызов подпрограмм Область видимости и время жизни переменной. Механизм передачи параметров.	Записать арифметические выражения Пояснить работу команд языка
Тема 3.2.	Структуризация в программировании Основы структурного программирования. Методы структурного программирования	
Тема 3.3.	Модульное программирование Модульное программирования. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули	
Раздел 4	Структуры данных(Массивы, Строки, Множества, Записи, Файлы, Указатели)	
	Понятие массива. Особенности программирования Написание программы с массивами. Решение задач Символьный и строковый типы. Объявление типов. Поиск, удаление, замена и добавление символ в строке. Операции со строками. Стандартные функции и процедуры для работы со строками. Решение задач. Понятие множества. Объявление множества. Операции над множествами. Определение типа запись. Правила работы с записями. Типы файлов. Организация доступа к файлам. Файлы последовательного доступа. Открытие и закрытие файла последовательного доступа. Запись в файл и чтение из файла последовательного доступа. Файлы произвольного доступа. Порядок работы с файлами произвольного доступа. Создание структуры и записи. Открытие и закрытие файла произвольного доступа. Использование файла произвольного доступа Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти.	Объяснить команды OPEN и CLOSE Записать операции вырезки и склеивания текста

	Создание и удаление динамических переменных. Структуры данных на основе указателей.	
Раздел 5	Объективно-ориентированное программирование Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Событийно-управляемая модель программирования. Компонентное-ориентированный подход. Классы объектов. Компоненты и их свойства.	реферат по теме «история развития ООП». Чтение и анализ литературы [2] стр. 157-162
Тема 5.2	Интегрированная среда разработчика Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	Сообщение по теме «Интегрированная средаразработчика» Чтение и анализ литературы [2] стр. 164
Тема 5.3	Этапы разработки приложения Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Программирование приложения. Тестирование, отладка приложения. Создание документации.	Создание проекта «Таблица умножения» Сообщение на тему «Юзабилити»
Тема 5.4	Иерархия классов Классы объективно-ориентированного языка программирования; виды, назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса Наследование. Перегрузка методов.	Выполнение домашних заданий по теме 5.4
Темы 5.5 Темы 5.6	Визуальное событийно-управляемое программирование Основные компоненты(элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления. Свойства компонентов(элементов управления) Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства. События компонентов(элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов, Вызов событий Разработка оконного приложения Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Создание процедур обработки событий. Компиляция и запуск приложения.	Выполнение индивидуального проектного задания по теме «Визуальное событийно-управляемое программирование»

Практические задания: по дисциплине «Основы программирования»

Задачи с решениями.

Задача № 1

1) **Постановка задачи:** Составить программу нахождения площади прямоугольника со сторонами X и Y.

2) Интерфейс задачи:

Form1

Площадь прямоугольника со сторонами X и Y:

Введите X:

Введите Y:

Результат:

Далее>>

Задача № 2

1) **Постановка задачи:** Составить программу перевода строки в нижний регистр.

2) Интерфейс задачи:

Form2

Перевод строки в нижний регистр.

Введите строку в верхнем регистре:

Результат

<<Назад Далее>>

Задача № 3

1) **Постановка задачи:** Составить программу перевода температуры из шкалы Фаренгейта в шкалу Цельсия (0 F соответствует -17,8, а 0 C соответствует+32 F).

2) Интерфейс задачи:

Form3

Перевод температуры из шкалы Фаренгейта в шкалу Цельсия.

Ведите температуру по Фаренгейту:

Результат по Цельсию:

<<Назад Далее>>

Задача № 4

1) **Постановка задачи:** Составить программу определения, в норме ли вес обследуемого пациента (нормой считается вес, равный $(\text{рост}(\text{см})-100)\pm 5\text{кг}$).

2) Интерфейс задачи:

Form4

В норме ли вес?

Введите рост(см):

Введите вес(кг):

Результат

<<Назад Далее>>

Project1

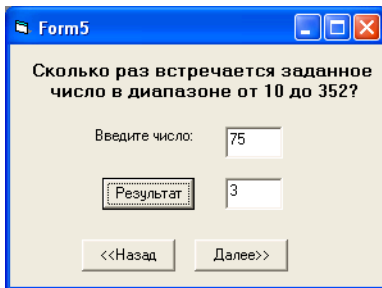
Вес не в норме

ОК

Задача № 5

1) **Постановка задачи:** Составить программу, определяющую сколько раз встречается заданное число (вводится с клавиатуры) в диапазоне от 10 до 352.

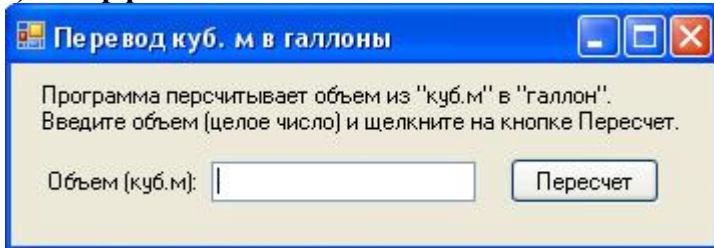
2) Интерфейс задачи:



Задача № 6

1) **Постановка задачи:** Программа пересчитывает из кубического метра в галлон.

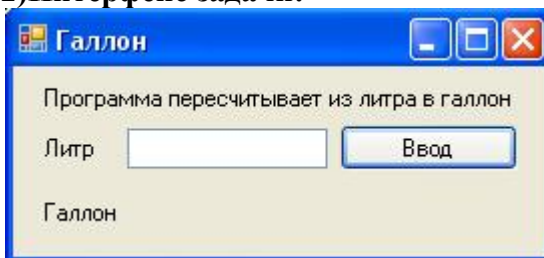
2) **Интерфейс задачи:**



Задача № 7

1) **Постановка задачи:** Программа пересчитывает из литра в галлон.

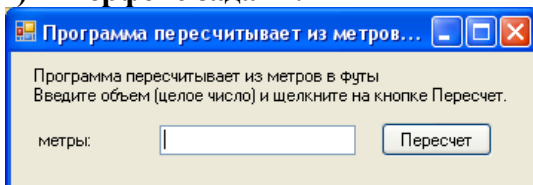
2) **Интерфейс задачи:**



Задача № 8

1) **Постановка задачи:** Программа пересчитывает из метров в футы.

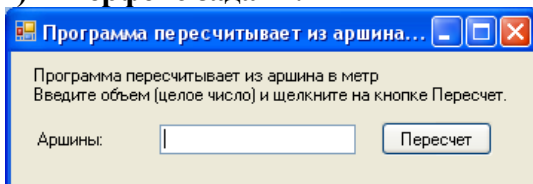
2) **Интерфейс задачи:**



Задача № 9

1) **Постановка задачи:** Программа пересчитывает из аршина в метр.

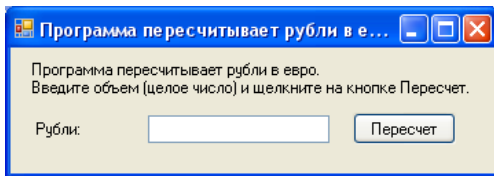
2) **Интерфейс задачи:**



Задача № 10

1) **Постановка задачи:** Программа пересчитывает рубли в евро.

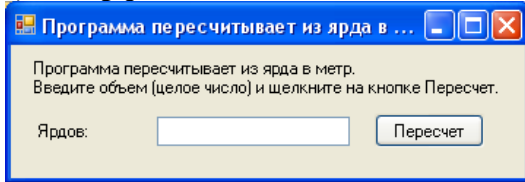
2) **Интерфейс задачи:**



Задача № 11

1) **Постановка задачи:** Программа пересчитывает из ярда в метр.

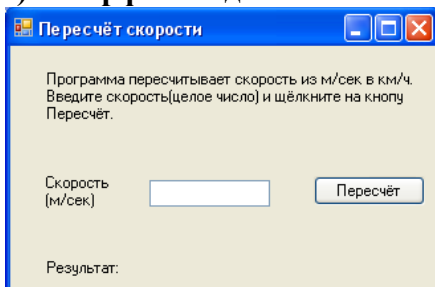
2) **Интерфейс задачи:**



Задача № 12

1) **Постановка задачи:** Программа пересчитывает скорость ветра из м/с в км/ч.

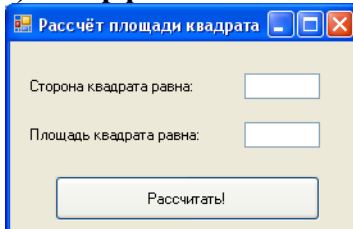
2) **Интерфейс задачи:**



Задача № 13

1) **Постановка задачи:** Программа рассчитывает площадь квадрата.

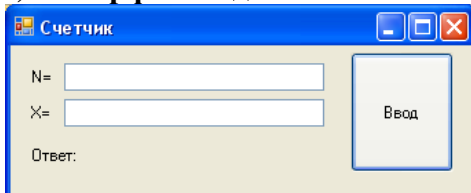
2) **Интерфейс задачи:**



Задача № 14

1) **Постановка задачи:** Программа рассчитывает значение по заданной формуле. $S = \sum_{i=1}^N (\sin x)^i$

2) **Интерфейс задачи:**

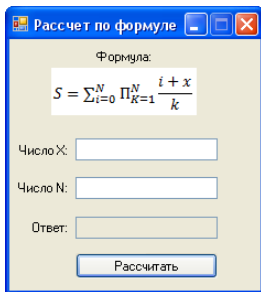


Задача № 15

1) **Постановка задачи:** Программа рассчитывает значение по заданной формуле.

$$S = \sum_{i=0}^N \prod_{k=1}^N \frac{i+x}{k}$$

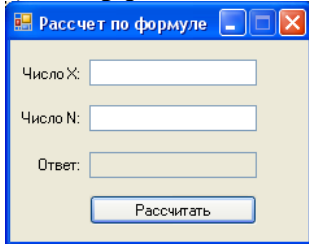
2) **Интерфейс задачи:**



Задача № 16

1) **Постановка задачи:** Программа рассчитывает значение по заданной формуле. $S = \sum_{i=0}^N \prod_{k=1}^N \frac{i+x}{k}$

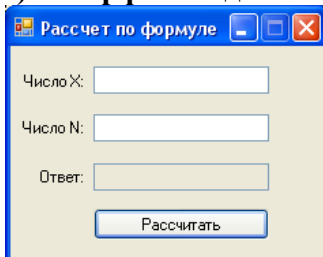
2) **Интерфейс задачи:**



Задача № 17

1) **Постановка задачи:** Программа рассчитывает значение по заданной формуле. $P = \prod_{i=1}^N (x - i)$

2) **Интерфейс задачи:**



Описание процедуры экзамена

Комплексный экзамен по дисциплине ОП.05 «Основы программирования»

Количество экзаменационных билетов превышает количество обучающихся.

В билете – 1 теоретический вопрос и 1 практическое задание. На подготовку к ответу обучающемуся отводится до 30 минут.

Обучающийся предъявляет ответы в устной (письменной, смешанной) форме: устно раскрывает теоретические вопросы; решение задачи представляется в письменном виде с устными комментариями (пояснениями), использование ПК обязательно.

Требования охраны труда: инструктаж по технике безопасности, правилам поведения на занятии, по соблюдению дисциплины, наличие инструктора (преподаватель).

Критерии оценки:

Оценка "отлично" ставится обучающемуся, обнаружившему всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоившему основную литературу, рекомендованную программой, взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившему творческие способности в понимании, изложении и использовании учебно-программного материала.

Оценка "хорошо" ставится обучающему, показавшему полное знание учебно-программного материала, успешно выполняющему предусмотренные в программе задания, усвоившему основную литературу, рекомендованную в программе, показавшему систематический характер знаний по дисциплине и способному к их самостоятельному пополнению и обновлению в ходе

дальнейшей учебной работы и профессиональной деятельности.

Оценка "удовлетворительно" ставится обучающему, показавшему знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справляющемуся с выполнением заданий, предусмотренных программой, знакомому с основной литературой, рекомендованной программой. Оценка "удовлетворительно" выставляется обучающемуся, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

Оценка "неудовлетворительно" выставляется обучающемуся, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий. Оценка "неудовлетворительно" ставится обучающимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании техникума без дополнительных занятий по соответствующей дисциплине.

Перечень материалов, оборудования и информационных источников, используемых при проведении промежуточной аттестации

При проведении промежуточной аттестации обучающимся могут пользоваться конспектом лекций, ПК, справочными материалами.

Эталоны ответов на теоретические вопросы:

Эталоны ответов на теоретические вопросы: по дисциплине «Основы программирования»

1. Основные понятия алгоритмизации. Понятия «алгоритм», «исполнитель алгоритма».

Алгоритм — набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное число действий. В старой трактовке вместо слова «порядок» использовалось слово «последовательность», но по мере развития параллельности в работе компьютеров слово «последовательность» стали заменять более общим словом «порядок». Это связано с тем, что работа каких-то инструкций алгоритма может быть зависима от других инструкций или результатов их работы. Таким образом, некоторые инструкции должны выполняться строго после завершения работы инструкций, от которых они зависят. Независимые инструкции или инструкции, ставшие независимыми из-за завершения работы инструкций, от которых они зависят, могут выполняться в произвольном порядке, параллельно или одновременно, если это позволяют используемые процессор и операционная система. Исполнитель алгоритма — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

2. Свойства и формы записи алгоритмов.

Основные свойства алгоритмов следующие:

1. Понятность для исполнителя — исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.
2. Дискретность (прерывность, раздельность) — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).
3. Определенность — каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.
4. Результативность (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за

невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

Массовость означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

На практике наиболее распространены следующие формы представления алгоритмов:

- словесная (запись на естественном языке);
- графическая (изображения из графических символов);
- псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- программная (тексты на языках программирования).

Основные алгоритмические конструкции: линейные, разветвляющиеся и циклические.

1. Базовая структура "следование". Образуется последовательностью действий, следующих одно за другим.

2. Базовая структура "ветвление". Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Структура ветвление существует в четырех основных вариантах:

если—то;

если—то—иначе;

выбор;

выбор—иначе.

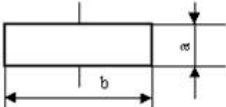
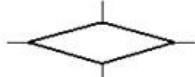
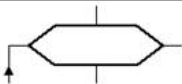
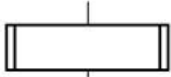
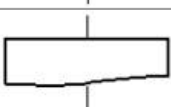
3. Базовая структура "цикл". Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла.


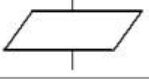

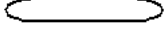

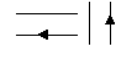


3. Составление блок-схем алгоритмов.

Блок-схемой называют графическое представление алгоритма, в котором он изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.

В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий.

Приведем наиболее часто употребляемые символы.

Название	Элемент	Комментарий
Процесс		Вычислительное действие или последовательность вычислительных действий
Решение		Проверка условия
Модификация		Заголовок цикла
Предопределенный процесс		Обращение к процедуре
Документ		Вывод данных, печать данных

Перфокарта		Ввод данных
Ввод/Вывод		Ввод/Вывод данных
Соединитель		Разрыв линии потока
Начало, Конец		Начало, конец, пуск, останов, вход и выход во вспомогательных алгоритмах
Комментарий		Используется для размещения надписей
Горизонтальные и вертикальные потоки		Линии связей между блоками, направление потоков
Слияние		Слияние линий потоков
Межстраничный соединитель		Нет

Языки и системы программирования. Языки высокого и низкого уровня.

Язык программирования — формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программы и действия, которые выполнит исполнитель (компьютер) под её управлением.

Языки программирования низкого уровня.

Первые компьютеры приходилось программировать двоичными машинными кодами. Однако программировать таким образом - довольно трудоемкая и тяжелая задача. Для упрощения этой задачи начали появляться языки программирования низкого уровня, которые позволяли задавать машинные команды в понятном для человека виде. Для преобразования их в двоичный код были созданы специальные программы - трансляторы.

Языки программирования высокого уровня.

Особенности конкретных компьютерных архитектур в них не учитываются, поэтому созданные приложения легко переносятся с компьютера на компьютер. В большинстве случаев достаточно просто перекомпилировать программу под определенную компьютерную архитектурную и операционную систему. Разрабатывать программы на таких языках значительно проще и ошибок допускается меньше. Значительно сокращается время разработки программы, что особенно важно при работе над большими программными проектами.

Правила записи выражений и операций. Типы данных. Синтаксис.

Типы данных *Visual Basic*.

Тип данных	Объем занимаемой памяти	Краткая запись
<i>Целые типы</i>		
Integer (целое)	2 байта	%
Long (целое двойной длины)	4 байта	&
Byte (Байт)	1 байт	
Boolean (булево)	2 байта	
<i>Плавающие типы</i>		
Single Десятичные числа	4 байта	!
Double Десятичные числа	8 байт	#
<i>Строковые типы</i>		
String Текстовая информация	1 байт на каждый символ	\$

Объектные типы		
Object Рисунки или ссылки на любой другой объект	4 байта	
Variant-типы		
Variant (числовые типы)	16 байт	
Variant (строковые типы)	22 байта +длина строки	
Прочие типы		
Currency Число в денежном формате	8 байт	@
Date Дата	8 байт	

Синтаксис.

Dim a as Integer

Понятие модуля и формы.

В Visual Basic весь программный код находится внутри процедур (подпрограмм). Общая структура описания подпрограммы Visual Basic:

Sub <имя_подпрограммы> [(<параметры>)]

<операторы>

End Sub

Такие процедуры могут вызываться или самим Visual Basic, или другими процедурами. Вызов подпрограммы выполняется с помощью следующих операторов:

Call <имя_подпрограммы> [(<параметры>)];

<имя_подпрограммы> [<параметры>]. Функция построена точно так же, как и процедура, однако результатом работы функции является возвращаемое значение (только одно).

Общая структура описания функции следующая:

Function <имя_функции> [(<аргументы>)] [As <Тип>] <операторы>

<имя_функции> = <выражение> End Function

Вызов функции выполняется с помощью оператора; присваивания:

<переменная> = <имя_функции> {(<аргументы>)}.</p>
</div>

Пользовательские типы данных.

Содержит данные в определенном пользователем формате. Оператор Structure определяет формат.

Формат объявления

Объявление структуры начинается с оператора Оператор Structure и завершается оператором EndStructure. Оператор Structure содержит имя структуры, которая является также идентификатором типа данных, определяемых структурой. Другие части кода могут использовать этот идентификатор для объявления переменных, параметров и возвращаемых значений функций в качестве типа данных структуры.

Объявления между операторами Structure и EndStructure определяют члены структуры.

Уровни доступа к членам

Каждый член необходимо объявить с помощью оператора Оператор Dim (Visual Basic) или оператора, определяющего уровень доступа, например, Public (Visual Basic) или Friend (Visual Basic) или Private (Visual Basic). Если используется оператор Dim, то уровень доступа по умолчанию является открытым (Public).

Использование форм, событий и методов.

Как объекты, так и формы могут выполнять методы и реагировать (откликаться) на события.

При каждом изменении размера формы в результате действий пользователя или программным способом инициируется событие Resize (Изменить размер) формы. Это позволяет изменять размеры элементов управления на форме или перемещать их, когда изменены размеры самой формы.

Событие Activate (Активизировать) происходит всегда, когда форма становится активной, а событие Deactivate (Деактивировать) - когда активной становится другая форма приложения. Эти события удобны для организации поведения формы при ее инициировании и завершении работы с ней. Например, можно написать код, который в случае события Activate выделит текст в каком-нибудь текстовом окне, а в случае события Deactivate внесенные изменения будут сохранены в файле или базе данных.

Чтобы сделать форму видимой, следует вызвать метод show (Показать):

Form2.Show

Вызов метода show имеет тот же эффект, что и установка значения свойства visible (Видимый) формы в True (Истина).

Многие из методов формы работают с текстом или графикой. Методы Print (Печатать), Line (Линия), Circle (Окружность) и Refresh (Обновить) полезны для печати или рисования непосредственно на поверхности формы.

Свойство Container

Свойство container (Набор) используется для изменения набора объектов внутри формы. Следующие элементы управления могут содержать другие элементы управления:

- Рамка
- Графическое окно

Использование управляющих элементов.

Элемент управления ADO Data

Элемент управления ADO Data (рис. 1) обеспечивает связь между поставщиком данных OLE DB (OLE DB Provider) и элементами управления, обладающими свойством DataSource (CheckBox, ComboBox, Image, Label, ListBox, PictureBox и TextBox).

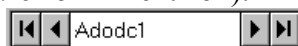


Рис. 1

ADO Data может применяться в следующих случаях:

- для связи с локальной или удаленной базой данных;
- для открытия (просмотра) заданной таблицы в базе данных или определения группы записей, основанных либо на SQL-запросе, либо на хранящейся процедуре;
- для передачи значений полей данных в связанные с данными элементы управления;
- для обновления базы данных.

ADO Data добавляется к формам аналогично любому другому элементу управления, причем их количество в форме не ограничено. Однако следует помнить о том, что данный элемент управления является ресурсоемким: ему требуется по крайней мере два соединения для первого компонента и по одному для каждого последующего.

Составление программ циклической структуры.

Операторы циклов

В языке VB существует 3 вида циклов: 1) цикл с параметром или цикл типа for; 2) цикл с предусловием или цикл типа while, 3) цикл с постусловием или цикл типа do ... while. Во всех этих циклах условие продолжения цикла заключается в круглые скобки. В циклах типов for и while повторяющаяся часть состоит из одного оператора, если требуется выполнить в цикле несколько операторов, они заключаются в фигурные скобки, образуя составной оператор. В цикле с постусловием тело цикла помещается между словами do и while. В отличие от цикла с предусловием, цикл с постусловием выполнится хотя бы один раз. Циклы с пред- и постусловием продолжают, если условие продолжения истинно.

Логические операторы и операторы сравнения If...Then, SelectCase.

Конструкция If...Then...Else аналогична конструкции If...Then, но позволяет задать действия, исполняемые как при выполнении условий, так и в случае их невыполнения.

Конструкция имеет следующий синтаксис:

If условиеThen

конструкции для обработки истинного условия

Else

конструкции для обработки ложного условия

End If

Ключевые слова If и End If имеют тот же смысл, что и в конструкции If...Then. Если заданное в конструкции условие не выполняется (результат проверки равен False) и конструкция содержит ключевое слово Else, Visual Basic выполнит последовательность конструкций, расположенных следом за Else. После чего управление перейдет к конструкции, следующей за End If.

Например:

```
If x >= 0 Then
```

```
Label1.Text = "Значение больше или равно 0"
```

```
Else
```

```
Label1.Text = "Значение меньше 0"
```

```
End If
```

Конструкция Select Case позволяет обрабатывать в программе несколько условий. Она аналогична блоку конструкций If...Then...Else. Эта конструкция состоит из анализируемого выражения и набора операторов Case на каждое возможное значение выражения. Работает данная конструкция следующим образом.

Основы программирования в Visual Basic 2010 65

Сначала Visual Basic вычисляет значение заданного в конструкции выражения. Затем полученное значение сравнивается со значениями, задаваемыми в операторах Case конструкции. Если найдено искомое значение, выполняются команды, приписанные данному оператору Case. После завершения выполнения конструкций управление будет передано конструкции, следующей за ключевым словом End Select.

Синтаксис конструкции Select Case следующий:

```
Select Case сравниваемоеЗначение
```

```
Case значение1
```

```
операторы1
```

```
Case значение2
```

```
операторы2
```

```
...
```

```
Case Else
```

```
операторыN
```

```
End Select
```

В начале конструкции расположены ключевые слова Select Case, указывающие, что находящийся рядом с ними параметр сравниваемоеЗначение будет проверяться на несколько значений. Затем в конструкции размещены группы команд, начинающиеся с ключевого слова Case. Если параметр сравниваемоеЗначение равно значению, указанному в текущем операторе Case, то будут выполняться команды, расположенные между этим и следующим ключевым словом Case. Конструкция может содержать любое количество ключевых слов Case с соответствующими им блоками операторов. Если ни одно значение не подошло, будут выполнены операторы, следующие за ключевыми словами Case Else. Ключевые слова Case Else могут быть опущены.

Обзор структуры цикла. Использование Do...Loop, Использование For...Next.

Цикл, задаваемый **конструкцией Do...Loop**, выполняется до тех пор, пока истинно задаваемое в цикле условие.

Синтаксис конструкции Do...Loop имеет следующий вид:

```
Do While условие
```

```
операторы
```

```
Loop
```

Аргумент конструкции условие является логическим выражением, значение

которого проверяется перед каждым проходом цикла. Если это значение равно True, выполняется последовательность команд, которые расположены между Do While и ключевым словом Loop. Эти конструкции образуют тело цикла. Если при очередном проходе цикла условие равно False, происходит выход из цикла и управление передается конструкции, следующей за Loop. Возможна ситуация, при которой операторы цикла не выполняются ни разу. Она возникает в том случае, если при первой проверке условие оказывается ложным.

Конструкция For...Next выполняет последовательность команд определенное число раз. Такую конструкцию называют циклом, а выполняемые ею программные коды — телом цикла.

Синтаксис конструкции For...Next следующий:

For счетчик[As типДанных] = начЗначение To конЗначение[Step шаг]

операторы

Next [счетчик]

Первый аргумент конструкции счетчик определяет имя переменной, которая будет "читать" количество выполненных циклов. Эту переменную можно объявить прямо в конструкции. Параметр начЗначение указывает числовое значение, которое присваивается переменной-счетчику перед первым проходом цикла. Цикл выполняется до тех пор, пока значение счетчика не превысит конечное значение, указанное после ключевого слова To. После каждого прохода значение счетчика изменяется на величину шаг, указанную после ключевого слова Step. Ключевое слово Next обозначает конец тела цикла и является обязательным.

Перед каждым проходом цикла Visual Basic сравнивает значения счетчика и аргумента конЗначение. Если значение счетчика не превышает установленного значения конЗначение, выполняются конструкции тела цикла. В противном случае управление переходит к следующей за Next конструкции.

Работа с логическими операторами и операторами сравнения.

Над условными выражениями можно выполнять действия логической математики (логические операции), а именно:

AND(И) — возвращает значение True, если все участвующие в операции выражения имеют значение True. В остальных случаях возвращается значение False;

OR(ИЛИ) — возвращает значение True, если хотя бы одно из участвующих в операции выражений имеет значение True. В случае, когда все выражения имеют значение False, возвращается значение False;

XOR(исключающее ИЛИ) — возвращает значение True, если только одно из участвующих в операции выражений имеет значение True. В остальных случаях возвращается значение False;

NOT(НЕ) — операция отрицания. Возвращает обратное значение для значения выражения, т. е. если выражение равно True, то возвращается False, и наоборот, если значение выражения равно False, то возвращается значение True.

Написание кода с использованием операторов и циклов.

1. Разложить по этапам написание программ с использованием циклов.
2. Какие циклические конструкции бывают.
3. Операторы, какие они бывают.

Отслеживание и анализ ошибок.

Ошибки в программе могут быть синтаксическими или смысловыми. Синтаксические ошибки наиболее очевидны. Они возникают, если код написан без соблюдения правил языка программирования. Эти ошибки обнаруживаются компилятором, который выдает соответствующее сообщение. В Visual

Basic 2010 такие сообщения отображаются в окне Error List(Список ошибок). В них указаны номер строки, файл, в котором обнаружена ошибка, и краткое ее описание. Обнаружение и исправление данных ошибок является достаточно легким и быстрым.

В набор инструментария отладки Visual Basic 2010 входят такие основные инструменты, как:

панель инструментов Standard(Стандартная), а также Debug(Отладка)

с кнопками выполнения команд для отладки приложения;

окно Immediate Window(Окно непосредственного выполнения), предназначенное для непосредственного ввода команд, требующих немедленного выполнения;

окно Watch (Наблюдение), служащее для просмотра значений выражений, включенных в список просмотра;

окно Locals(Локальные переменные), предназначенное для просмотра значений переменных;

редактор кода со встроенными возможностями просмотра переменных, констант, свойств, выражений при отладке приложения в точках останова и пошаговом выполнении приложения;

окно Call Stack(Стек вызовов) для просмотра вызванных, но незавершенных процедур.

Обзор стандартных элементов.

Помимо индивидуальных свойств, каждый элемент управления содержит общие для большинства свойства. В табл. 4.1 перечислены наиболее часто используемые свойства элементов управления.

Свойство	Описание
BackColor	Задаёт цвет фона элемента управления
ContextMenu	Контекстное меню, открываемое при нажатии правой кнопки мыши на элементе управления. Задаёт элемент управления ContextMenu
Dock	Задаёт расположение элемента управления в форме. Может принимать одно из значений перечисления DockStyle: Bottom (Снизу), Fill (На всю форму), Left (Слева), None (В любом месте формы), Right (Справа) и Top (Сверху)
Enabled	Определяет, доступен ли элемент управления. Значение False блокирует использование элемента
Font	Задаёт шрифт для отображения текста элемента управления. Данное свойство в окне Properties (Свойства) задаётся с помощью диалогового окна настройки шрифта
ForeColor	Определяет цвет располагаемого на элементе управления текста
Image	Задаёт рисунок, который будет отображаться на элементе управления
Location	Определяет расположение левого верхнего угла элемента управления. Для задания координат расположения можно также использовать свойства X и Y
Locked	Значение True этого свойства не позволяет перемещать элемент управления и изменять его размеры во время разработки формы приложения
Name	Задаёт имя, используемое при обращении к элементу управления
Size	Определяет размер элемента управления, включающий ширину и высоту элемента управления. Для задания ширины и высоты также могут отдельно применяться свойства Width и Height соответственно
TabIndex	Задаёт порядок элемента управления в форме. По умолчанию значение свойству присваивается в порядке добавления элемента в форму. Так первому элементу, добавленному в форму, присваивается значение 0
TabStop	Определяет, может ли установиться фокус на элементе управления с помощью клавиши <Tab>. Если установлено значение False, то элемент управления пропускается при переходе от одного элемента к другому
Visible	Задаёт видимость элемента управления при выполнении приложения. Значение True определяет, что элемент виден пользователю

Создание программы с использованием полос прокрутки, таймера и заданием даты.

На форме можно размещать горизонтальные и вертикальные полосы прокрутки с помощью элементов управления **HScrollBar** и **VScrollBar**. Полосы прокрутки встречаются при работе с документами программы Microsoft Word и другими программными продуктами, работающими в среде Windows. Они также используются в многострочных текстовых полях и списках, в которых информация целиком не помещается. Элементы управления **VScrollBar** и **HScrollBar** отличаются от полос прокрутки, встроенных в перечисленные элементы, т. к. они существуют самостоятельно и применяются для элементов, не имеющих собственных полос прокрутки, или группы элементов.

В Visual Basic существует элемент управления, который обрабатывает данные системных часов. Этот объект называется **таймером**. Его можно применять для выполнения определенных действий через заданный интервал времени.

Для размещения в форме таймера используется элемент управления **Timer**.

Событие **Tick** таймера наступает через каждый установленный в свойстве **Interval** промежуток времени. В процедуре обработки данного события необходимо определить действия, выполняемые с заданной частотой.

Для запуска и останова таймера помимо свойства **Enabled** можно использовать методы **Start** и **Stop**.

В Visual Basic существуют элементы управления **MonthCalendar** и **DateTimePicker**, позволяющие работать с датами. Объект **MonthCalendar** представляет собой календарь, с помощью которого можно выбрать некоторый диапазон дат. Элемент управления **DateTimePicker** имеет вид текстового поля с расположенной справа кнопкой, при нажатии которой открывается календарь. Этот элемент управления, как правило, используют для экономии места на форме и при выборе одной даты.

Создание вкладок, индикатора прогресса, ползунка.

Visual Basic позволяет создавать формы, содержащие несколько **вкладок**.

Объекты данного типа удобно использовать в том случае, когда необходимо разместить большой объем информации или когда для удобства работы требуется основную, наиболее часто используемую информацию, сгруппировать в одном месте, отделив ее от менее важной информации.

Для создания вкладок в форме предназначен элемент управления **TabControl**.

Рассмотрим размещение данного элемента в форме и настройку его свойств:

1. Откройте форму, в которой хотите создать вкладки.
2. Нажмите кнопку **TabControl** на панели элементов управления.
3. Установите указатель в форму и, удерживая кнопку мыши в нажатом состоянии, переместите курсор по диагонали так, чтобы получилась рамка размером с форму.
4. Откройте окно свойств созданного объекта. Для добавления вкладок выберите свойство **TabPage** и нажмите кнопку с тремя точками, расположенную справа.
5. В открывшемся диалоговом окне **TabPage Collection Editor** (Редактор вкладок) с помощью кнопки **Add** (Добавить) добавьте необходимое количество вкладок.
6. В этом же окне можно настроить свойства вкладок. Для этого выберите в поле **Members** (Элементы) нужную вкладку, а с помощью расположенной справа области задайте требуемые значения для свойств. Например, используя свойство **Text**, задайте заголовки вкладок.
7. Нажмите кнопку **ОК** для закрытия диалогового окна.

Некоторые операции вашего приложения могут выполняться довольно продолжительное время. Это может быть, например, обработка большого массива данных или сложная выборка из базы данных, содержащей огромное количество записей. В подобной ситуации пользователь начнет нервничать, не

зависла ли программа. Работу продолжительных задач можно сопровождать отображением на экране индикатора процесса выполнения, используя для этого стандартный элемент управления **ProgressBar**.

Элемент управления **TrackBar** представляет собой ползунок, позволяющий вводить в программу числовые значения.

Свойство `TickStyle` задает расположение делений на линейке ползунка и может принимать следующие значения:

`Both`— деления расположены по обеим сторонам ползунка;

`BottomRight`— у горизонтального ползунка деления расположены снизу, у вертикального — справа;

`None`— деления у ползунка отсутствуют;

`TopLeft`— у горизонтального ползунка деления расположены сверху, у вертикального — слева.

Гиперссылки. Список.

Для создания **гиперссылок** используется элемент управления `LinkLabel`, представляющий собой усовершенствованный элемент `Label`, т. е. обладающий всеми свойствами элемента `Label` и имеющий специфичные, предназначенные для создания гиперссылок, свойства. Каждая гиперссылка может выполнять различные функции в приложении. Например, она может использоваться в качестве ссылки на сайт в Интернете или для открытия новой формы.

Элемент управления `LinkLabel` может содержать одну или более ссылок и в зависимости от этого различают способы настройки данного элемента. Рассмотрим каждый случай отдельно.

Элемент управления **ListView** представляет собой список элементов с использованием пиктограмм, аналогичный используемому в правой части окна Проводника.

В зависимости от свойства `View` список может принимать следующий вид:

`Details`(Таблица), `LargeIcon`(Крупные значки), `List`(Список), `SmallIcon`(Мелкие значки), `Tile`(Плитка).

Чтение и запись файла. Класс `FileStream`.

Для выполнения основных операций с файлами, такими как получение информации о файле, создание нового файла, удаление, копирование и перемещение, предназначены классы `File` и `FileInfo`.

Класс `File` содержит статические методы, при вызове которых требуется указание в качестве параметра имени файла. При работе с классом `FileInfo` с помощью конструктора создается представляющий конкретный файл экземпляр класса.

При работе с текстовыми файлами, например, при записи в них информации и считывании данных, используются классы `FileStream`, `StreamReader`, `StreamWriter`. Для выполнения бинарных операций с файлами применяются классы `BinaryReader` и `BinaryWriter`.

Класс **`FileStream`**, который является производным от абстрактного класса `Stream`, поддерживает операции синхронного и асинхронного открытия, чтения и записи последовательности байтов в файл. Класс имеет следующий конструктор:

```
Sub New(ByVal pathAs String, ByVal modeAs FileMode,
        ByVal accessAs FileAccess, ByVal shareAs FileShare,
        ByVal bufferSizeAs Integer, ByVal useAsyncAs Boolean) _
    As FileStream
```

где:

`path`— полное имя файла, включающее само имя файла и путь к нему;

`mode`— режим доступа к файлу. Может принимать значения, указанные

в табл. 7.4;

access— тип доступа к файлу. Определяет характер действий с файлом (чтение или запись данных). Может принимать значения: Read(Чтение), Write(Запись) или ReadWrite(Чтение и запись). Этот параметр можно опустить, тогда по умолчанию тип доступа будет принимать значение ReadWrite;

Считывание данных из текстового файла.

Для считывания данных из текстового файла используется класс StreamReader, наследуемый от абстрактного класса TextReader. Он имеет следующие основные конструкторы:

```
Sub New(ByVal pathAs String, ByVal encoding As Encoding)
```

```
Sub New(ByVal streamAs Stream, ByVal encoding As Encoding)
```

где:

path— полное имя файла, включающее само имя файла и путь к нему;

stream— поток для чтения;

encoding— кодировка знаков. Может принимать одно из значений перечисления Encoding:

ASCIIEncoding (Кодировка 7-разрядными ASCII-знаками), UnicodeEncoding (Кодировка в виде двух последовательных символов), UTF7Encoding (Кодировка UTF-7) и UTF8Encoding(Кодировка UTF-8).

Параметр можно опустить.

Продемонстрируем примеры создания объекта класса:

```
Dim streamReader1 As New StreamReader ("C:\MyFile.txt")
```

```
Dim fileStream As New FileStream("C:\MyFile.txt", FileMode.Open)
```

```
Dim streamReader2 As New StreamReader(fileStream)
```

Запись данных в текстовый файл.

Класс StreamWriter, производный от абстрактного класса TextWriter, предназначен для записи текстовых данных и имеет следующие основные конструкторы:

```
Sub New(ByVal pathAs String, ByVal appendAs Boolean,  
ByVal encodingAs Encoding)
```

```
Sub New(ByVal streamAs Stream, ByVal encoding As Encoding)
```

где:

path— полное имя файла, включающее само имя файла и путь к нему;

stream— поток для записи;

append— определяет, будет ли файл перезаписываться в случае указания существующего файла. Если указано значение True, файл создается или дописывается. Параметр можно опустить;

encoding— кодировка знаков. Может принимать одно из значений перечисления Encoding:

ASCIIEncoding(Кодировка 7-разрядными ASCII-знаками), UnicodeEncoding(Кодировка в виде двух последовательных символов), UTF7Encoding(Кодировка UTF-7) и UTF8Encoding(Кодировка UTF-8).

Параметр можно опустить.

Продемонстрируем примеры создания объекта класса StreamWriter:

```
Dim streamWriter1 As New StreamWriter("C:\MyFile.txt", true)
```

```
Dim fileStream As New FileStream("C:\MyFile.txt", FileMode.Append)
```

```
Dim streamWriter2 As New StreamWriter(fileStream)
```

Открытие и создание файла для чтения и записи.

Для получения объектов классов FileStream, StreamReader и StreamWriter можно воспользоваться перечисленными в табл. 7.8 методами класса File.

Класс FileInfo имеет аналогичные методы за тем исключением, что имя файла задается при создании экземпляра класса, и поэтому в методах параметр Path отсутствует.

6. ПРИЛОЖЕНИЕ-ВАРИАНТЫ БИЛЕТОВ

Министерство образования МО

Государственное бюджетное образовательное учреждение среднего профессионального образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 1 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Алгоритмический язык как средство выражения и записи алгоритмов.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО

Государственное бюджетное образовательное учреждение среднего профессионального образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 2 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Алгоритмы, их свойства, исполнители алгоритмов.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО

Государственное бюджетное образовательное учреждение среднего профессионального образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 3 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Линейный алгоритм, их свойства, способы описания, примеры.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 4 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Разветвляющийся алгоритм, полная и неполная ветка.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____ /Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 5 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Циклический алгоритм, цикл, тело цикла.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____ /Гусев И.Е.

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 6 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Типы переменных величин, их представления в ПЭВМ.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____ /Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 7 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Виды величин. Переменная величина, ее имя и значение.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____ /Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 8 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Команда присваивания. Алгоритмические выражения, их запись через команду присваивания.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____ /Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 9 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Алгоритм Поиск. Примеры.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____ /Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 10 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Описание массива. Способы ввода массива в ПЭВМ.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 11 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Стандартные функции языка. Оператор функции пользователя.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 12 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Основной алгоритм работы процессора. Считывание слова из памяти.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 13 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Команда повторения с параметром (FOR-TO-STEP-NEXT). Цикл, тело цикла, параметры цикла.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 14 по дисциплине «Основы программирования» Специальность «09.02.03» Курс III группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	---	---

1. Литерные величины, операции склеивания, вырезки текста.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 15 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Команды условного и безусловного перехода (IF-THEN, GOTO).
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 16 по дисциплине «Основы программирования» Специальность «09.02.03» Курс III группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	---	---

1. Операторы ввода данных (INPUT, DATA-READ).
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 17 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Файловая структура ОС. Команды ОС. Команды ОС для открытия и закрытия файлов (OPEN, CLOSE).
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 18 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Стандартные и специальные функции языка
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 19 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Алгоритм Минэлемент. Примеры.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 20 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Алгоритм Упорядочение. Примеры.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 21 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Вспомогательные алгоритмы (на примере БИТ и БИД).
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 22 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-1	Утверждено: зам. директора по учебной работе _____20__г. _____
---	---	---

1. Алгоритм Схема Горнера. Примеры.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 23 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Отношения между величинами в качестве условий исполнения алгоритма по таблице значений. Составные условия (И, ИЛИ, НЕ).
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 24 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Табличные величины. Одномерные и двумерные таблицы.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 25 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-31	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Принцип структурного подхода при разработке алгоритма. Виды алгоритмов. Понятие блок-схемы.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 26 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Вспомогательные алгоритмы. Подпрограммы (GOSUB-RETURN).
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____ 20__ г. _____	Экзаменационный билет № 27 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____ 20__ г. _____
---	--	---

1. Этапы решения задач на ЭВМ. Понятие программы и языка программирования.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 28 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	---	---

1. Арифметическое выражение. Примеры.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 29 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	---	---

1. Исполнение программы на ПЭВМ. Трансляция методом интерпретации и компиляции.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 30 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	---	---

1. Обзор современных языков программирования.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

*Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»*

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 31 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Оператор печати данных PRINT.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

*Министерство образования МО
Государственное бюджетное образовательное учреждение среднего профессионального
образования Московской области «Ореховский индустриальный техникум»*

Рассмотрено предметной комиссией _____20__г. _____	Экзаменационный билет № 32 по дисциплине «Основы программирования» Специальность «09.02.03» Курс II группа В-21	Утверждено: зам. директора по учебной работе _____20__г. _____
---	--	---

1. Программное обеспечение ПЭВМ. Системное и прикладное ПО.
2. Задача
 - а) алгоритм (блок-схема)
 - б) программа

Преподаватель: _____/Гусев И.Е./

Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

- [1] Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования: Учебное пособие (М. Форум Инфра-М, 2004).
- [2] Н. Угринович «Информатика и информационные технологии», (Москва, Бином. Лаборатория знаний, 2005 г.)
- [3] Сафронов И.К. «Задачник – практикум по информатике» (Санкт-Петербург, «БХВ-Петербург», 2002).
- [4] Князева М.Д. «Алгоритмы» (Москва, «Новый город», 2000).
- [5] Фролов В.В. «Основы программирования», Москва, «Высшая школа», 2001.
- [6] Шафрин Ю.А. «Основы компьютерных технологий», Ефимова О.В. «Практикум по компьютерным технологиям» (Москва, АБФ, 1997).
- [7] Брябрин И.Б. «Основы программирования», Москва «Наука», 1990

Дополнительные источники: конспект лекций

Профильные периодические издания:

1. Информатика
2. Информатика и образование