

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Мурманский арктический государственный университет»

Комплект контрольно-оценочных средств
по учебной дисциплине «Основы программирования»
специальности **09.02.03 Программирование в компьютерных системах**
базовой подготовки

УТВЕРЖДЕНО

Директор Колледжа ФГБОУ ВО «МАГУ»



/ Козлова Н.В./
Ф.И.О.

Мурманск

2019 г.

КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Общие сведения

1.	Специальность	09.02.03 Программирование в компьютерных системах
2.	Форма обучения	очная
3.	Дисциплина	ОП.05. Основы программирования
4.	Форма аттестации по учебной дисциплине	экзамен

Перечень формируемых знаний, умений и компетенций

Код ПК, ОК	Умения	Знания
ОК 1 - 9 ПК 1.1. ПК 1.2. ПК 1.3. ПК 1.4. ПК 1.5. ПК 3.1.	У.1 Работать в среде программирования У.2 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования	3.1 Этапы решения задачи на компьютере 3.2 Типы данных 3.3 Базовые конструкции изучаемых языков программирования 3.4 Принципы структурного и модульного программирования 3.5 Принципы объектно-ориентированного программирования

Показатели оценки результата освоения общих компетенций (ОК) по УД

Результаты (освоенные общие компетенции)	Основные показатели результатов подготовки
ОК.1 Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	демонстрация интереса к будущей профессии через: – повышение качества обучения по ПМ; – участие в студенческих олимпиадах, научных конференциях; – участие в органах студенческого самоуправления; – участие в социально-проектной деятельности; – портфолио студента
ОК.2 Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.	– выбор и применение методов и способов решения профессиональных задач в области эксплуатации компьютерных систем и комплексов; – оценка эффективности и качества выполнения профессиональных задач
ОК.3 Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	– решение стандартных и нестандартных профессиональных задач в области технологических процессов технического обслуживания компьютерных систем и комплексов
ОК.4 Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	– получение необходимой информации с использованием различных источников, включая электронные.
ОК.5 Использовать информационно-коммуникационные технологии в профессиональной деятельности.	– оформление результатов самостоятельной работы с использованием ИКТ; – работа с АРМами, Интернет
ОК.6 Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.	– взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения и практики; – умение работать в группе; – наличие лидерских качеств; – участие в студенческом самоуправлении;
ОК.7 Брать на себя ответственность за работу членов команды (подчиненных),	– проявление ответственности за работу подчиненных, результат выполнения заданий;

результат выполнения заданий.	– самоанализ и коррекция результатов собственной работы
ОК.8 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	– организация самостоятельных занятий при изучении дисциплин профессиональных модулей; – самостоятельный, профессионально-ориентированный выбор тематики творческих и проектных работ (курсовых, рефератов, докладов и т.п.); – составление резюме; – посещение дополнительных занятий;
ОК.9 Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	– анализ инноваций в области разработки технологических процессов; – использование «элементов реальности» в работах обучающихся (курсовых, рефератов, докладов и т.п.).

Результаты освоения учебной дисциплины, подлежащие проверке

Раздел Тема	Коды компетенций, формированию которых способствует элемент программы	Результаты обучения: умения, знания	Показатели оценки результата	Вид и формы контроля	Задания № приложения (УМК)
<p>Раздел 1. Современные технологии разработки программного обеспечения.</p> <p>Тема 1.1. Современные технологии разработки программного обеспечения.</p>	ОК. 1, 4, 5, 9	3.1, 3.5	<p>Знает назначение этапов решения задачи на компьютере:</p> <ul style="list-style-type: none"> – постановка задачи; – формализация задачи; – построение алгоритма; – составление программы на языке программирования; – отладка и тестирование программы; – проведение расчетов и анализ полученных результатов. <p>Знает объектно-ориентированные принципы построения программы в конкретной языковой среде программирования (инкапсуляция, полиморфизм, наследование).</p>	<p><i>Текущий</i></p> <p>Устный опрос</p> <p>Решение задач</p> <p>Тестирование</p> <p>Внеаудиторная самостоятельная работа</p>	<p>Дидактические материалы к уроку. Тема 1.1 (УМК)</p> <p>Дидактические материалы к уроку. Задачи. Тема 1.1 (УМК)</p> <p>БТЗ. Тест №1 «Современные технологии и среды разработки программного обеспечения»</p> <p>Сайт по дисциплине myvbnet.wix.com/myvbnet</p>
<p>Раздел 2. Язык программирования Visual Basic .NET.</p> <p>Тема 2.1. Основы языка программирования Visual Basic .NET.</p> <p>Тема 2.2. Объектно-ориентированное программирование.</p> <p>Тема 2.3. Работа с файловой системой.</p>	ОК. 2, 3, 4, 5, 6, 8, 9 ПК 1.1-1.5, ПК 3.1	У.1, У.2	<p>Умеет выполнять запись и отладку программного кода на конкретном языке программирования в современной интегрированной среде разработки программного обеспечения. Использует эффективно:</p> <ul style="list-style-type: none"> – редактор кода, технологию IntelliSense; – инструменты отладки; – информационные окна; – справочную систему. <p>Умеет:</p> <ul style="list-style-type: none"> – составлять алгоритмы типовых учебных задач; 	<p><i>Текущий</i></p> <p>Устный опрос</p> <p>Решение задач</p> <p>Тестирование</p>	<p>Дидактические материалы к уроку. Тема 2.1, 2.2, 2.3, 2.4 (УМК)</p> <p>Дидактические материалы к уроку. Задачи. Тема 2.1, 2.2, 2.3, 2.4 (УМК)</p> <p>БТЗ. Тест №2 «Основы языка программирования Visual Basic .NET», Тест №3 «Объектно-ориентированное программирование», Тест №4 «Программирование коллекций».</p>

Тема 2.4. Коллекции.			<ul style="list-style-type: none"> – записывать программу по составленному алгоритму посредством операторов конкретного языка программирования (сопоставлять базовые алгоритмические конструкции с операторами; выстраивать последовательность блоков, определять структуру модулей, параметры операторов). Реализует этапы решения задачи на компьютере: – составление программы на языке программирования; – отладка и тестирование программы; – проведение расчетов и анализ полученных результатов. 	<p>Внеаудиторная самостоятельная работа</p> <p><i>Промежуточный</i></p> <p>Практические работы 1 – 12</p>	<p>Сайт по дисциплине myvbnnet.wix.com/myvbnnet</p> <p>МУ к выполнению практических работ:</p> <p>Практическая работа №1 "Расчёт переменных";</p> <p>Практическая работа №2 "Алгоритм ветвления";</p> <p>Практическая работа №3 "Обработка ошибок времени выполнения";</p> <p>Практическая работа №4 "Обработка одномерных массивов";</p> <p>Практическая работа №5 "Обработка ошибок времени выполнения при работе с одномерными массивами";</p> <p>Практическая работа №6 "Вычисление корней уравнения";</p> <p>Практическая работа №7 "Вычисление суммы числового ряда";</p> <p>Практическая работа №8 "Обработка матрицы";</p> <p>Практическая работа №9 "Подпрограммы";</p> <p>Практическая работа №10 "Обработка строк";</p> <p>Практическая работа №11 "Перечисляемый тип";</p> <p>Практическая работа №12 "Структуры"; (УМК)</p>
		3.1, 3.2, 3.3, 3.4, 3.5,	<p>Знает основные типы данных:</p> <ul style="list-style-type: none"> – обозначение и характеристики; – формы объявления и инициализации переменных; – порядок применения и высвобождения переменных; – назначение основных свойств и методов объектных типов; – правила преобразования типов. <p>Знает формы записи операторов:</p> <ul style="list-style-type: none"> – ветвления (If... Then... Else; Select Case; GoTo); – циклов (For... Next; For Each... Next; While; Until). – обработки ошибок (Try... Catch... Finally); – организации подпрограмм (Funciton, Sub, Property). <p>Знает основные принципы структурного и модульного программирования:</p>		

			<ul style="list-style-type: none"> – отказ от использования оператора безусловного перехода GoTo; – любая программа строится из трёх базовых управляющих конструкций: последовательность, ветвление, цикл; – базовые управляющие конструкции могут быть вложены друг в друга произвольным образом; – повторяющиеся фрагменты программы можно оформить в виде подпрограмм (процедур и функций); – каждую логически законченную группу инструкций следует оформить как блок; – все перечисленные конструкции должны иметь один вход и один выход; – разработка программы ведётся пошагово, методом "сверху вниз". – программное средство разделяется на отдельные именованные сущности модули. <p>Знает концепции основных принципов объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.</p>		
<p>Раздел 3. Windows Forms</p> <p>Тема 3.1. Формы</p> <p>Тема 3.2. Элементы управления</p>	<p>ОК. 2, 3, 4, 5, 6, 8, 9</p> <p>ПК 1.1-1.5</p> <p>ПК 3.1</p>	<p>У.1, У.2</p>	<p>Умеет создавать с помощью инструментов интегрированной среды программирования типовые экранные формы со стандартными элементами управления.</p> <p>Умеет реализовывать рациональный, дружелюбный графический интерфейс пользователя приложения,</p>	<p><i>Текущий</i></p> <p>Устный опрос</p> <p>Решение задач</p> <p>Тестирование</p>	<p>Дидактические материалы к уроку. Тема 3.1, 3.2 (УМК)</p> <p>Дидактические материалы к уроку. Задачи. Тема 3.1, 3.2 (УМК)</p> <p>БТЗ. Тест №5 «Элементы управления формы»</p>

			обеспечивающий: ввод и отображение данных, обработку событий приложения.	Внеаудиторная самостоятельная работа <i>Промежуточный</i>	Сайт по дисциплине myvbnet.wix.com/myvbnet
		3.1, 3.2, 3.3, 3.4, 3.5	<p>Знает реализацию этапов решения задачи на компьютере:</p> <ul style="list-style-type: none"> – составление программы на языке программирования; – отладка и тестирование программы; – проведение расчетов и анализ полученных результатов. <p>Знает назначение и характеристики объектных типов основных элементов управления: Form, Label, Button, RadioButton, CheckBox, ComboBox, TextBox, ListBox.</p> <p>Знает методы использования базовых конструкций конкретного изучаемого языка программирования для реализации алгоритмов обработки событий основных элементов управления графического интерфейса.</p> <p>Знает методы использования принципов структурного и модульного программирования:</p> <ul style="list-style-type: none"> – реализует структуру программы в виде модулей – использует принципы структурного программирования. <p>Знает концепции основных принципов объектно-ориентированного программирования реализованных в стандартных элементах управления формы.</p>	Практические работы 13-16	МУ к выполнению практических работ: Практическая работа №13 "Объектная модель классов"; Практическая работа №14 "Подсистема ввода/вывода"; Практическая работа №15 "Windows Forms Расчет переменных"; Практическая работа №16 "Windows Forms Обработка одномерных массивов"; (УМК)
				Промежуточный контроль	экзамен

Порядок и условия организации промежуточной аттестации по дисциплине

- 1) Форма проведения аттестации *экзамен*
- 2) Требования к студенту по допуску к промежуточной аттестации:
наличие положительных оценок за все практические работы
- 3) Количество вариантов заданий 25 (по количеству студентов)
- 4) Время выполнения задания 45 мин.
- 5) Оборудование персональный компьютер.
- 6) Литература для студентов, использование которой разрешено на экзамене: не допускается.

Типовой образец экзаменационного билета

1. Дать характеристику этапов решения задач на компьютере.
2. Создать консольное приложение, в котором необходимо:
 - 1.1. Объявить переменные для хранения сведений об аппаратных характеристиках компьютера: модель процессора, объем оперативной памяти, дата изготовления BIOS.
 - 1.2. Обеспечить консольный ввод переменных с пояснениями к вводимым значениям.

Типовые теоретические вопросы к экзамену

1.	Дать характеристику этапов решения задач на компьютере.
2.	Современные языки программирования.
3.	История языка программирования Visual Basic .NET.
4.	Интегрированная среда разработки.
5.	Программная платформа .NET Framework.
6.	Структура программы на Visual Basic .NET.
7.	Объявление переменных и констант.
8.	Операторы присваивания.
9.	Арифметические операторы.
10.	Операторы сравнения. Простые условия.
11.	Логические операторы. Составные условия.
12.	Условные конструкции. Оператор If...Then...Else.
13.	Условные конструкции. Оператор Select...Case.
14.	Оператор безусловного перехода GoTo.
15.	Ошибки времени выполнения. Исключения.
16.	Структурная обработка исключений.
17.	Объявление и использование одномерных массивов.
18.	Операторы циклических алгоритмов с заданным числом повторений.
19.	Объявление и использование многомерных массивов.
20.	Динамическая размерность массивов.
21.	Операторы циклов с предусловием и постусловием.
22.	Подпрограммы-методы. Сигнатура подпрограмм функций и процедур.
23.	Список параметров подпрограммы.
24.	Операторы и методы для обработки строковых типов данных (String, Char).
25.	Основные понятия и принципы объектно-ориентированного программирования.

Критерии оценки устных ответов обучающихся

Оценка «5» ставится, если обучающийся: 1) полно и аргументировано отвечает по содержанию задания; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры; 3) излагает материал последовательно и правильно.

Оценка «4» ставится, если обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.

Оценка «3» ставится, если обучающийся обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки.

Оценка «2» ставится, если обучающийся обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.

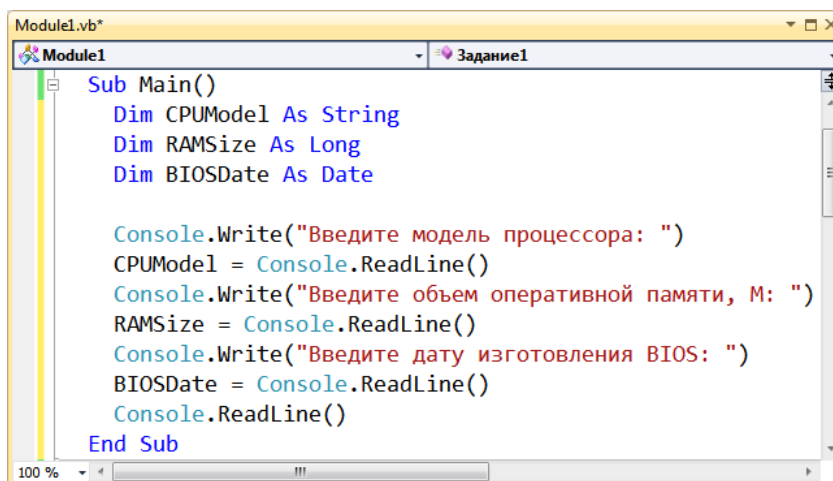
Типовые примеры практических заданий к экзамену

Задание 1.

Создать консольное приложение, в котором необходимо:

1. Объявить переменные для хранения сведений об аппаратных характеристиках компьютера: модель процессора, объем оперативной памяти, дата изготовления BIOS.
2. Обеспечить консольный ввод переменных с пояснениями к вводимым значениям.

Решение



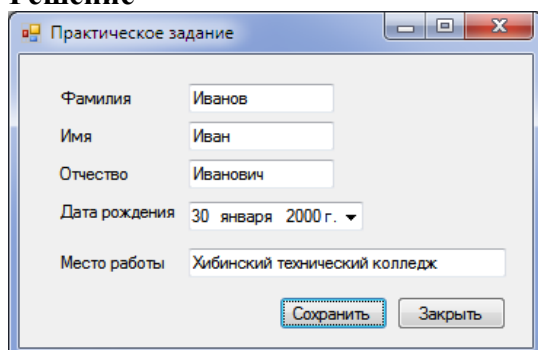
```
Module1.vb*
Module1
Задание1
Sub Main()
    Dim CPUModel As String
    Dim RAMSize As Long
    Dim BIOSDate As Date

    Console.WriteLine("Введите модель процессора: ")
    CPUModel = Console.ReadLine()
    Console.WriteLine("Введите объем оперативной памяти, М: ")
    RAMSize = Console.ReadLine()
    Console.WriteLine("Введите дату изготовления BIOS: ")
    BIOSDate = Console.ReadLine()
    Console.ReadLine()
End Sub
```

Задание 2.

Создать графическое приложение для запроса сведений о персоне: фамилия, имя, отчество, дата рождения, место работы. На форме разместить две кнопки: 1) сохранить строки данных в структуре; 2) закрыть программу.

Решение



Фамилия	Иванов
Имя	Иван
Отчество	Иванович
Дата рождения	30 января 2000 г.
Место работы	Хибинский технический колледж

Сохранить Закрыть

```

Form1.vb*
Form1 (Объявления)
Structure ТипПерсона
    Dim Фамилия As String
    Dim Имя As String
    Dim Отчество As String
    Dim ДатаРождения As Date
    Dim МестоРаботы As String
End Structure

Private Персона As ТипПерсона

Private Sub КнопкаСохранить_Click(sender As System.Object,
    Персона.Фамилия = TextBoxФамилия.Text
    Персона.Имя = TextBoxИмя.Text
    Персона.Отчество = TextBoxОтчество.Text
    Персона.ДатаРождения = DateTimePickerДатаРождения.Value
    Персона.МестоРаботы = TextBoxМестоРаботы.Text
End Sub

Private Sub КнопкаЗакрыть_Click(sender As System.Object, e
    Close()
End Sub
End Class
100 %

```

Критерии оценки:

- 1) полнота и правильность решения задания;
- 2) последовательность и логичность выполнения действий при выполнении действий;
- 3) владение терминологическим аппаратом и использование его при ответе (пояснение при демонстрации работы).

Оценка «отлично» ставится, если обучающийся выполнил работу в полном объеме с соблюдением необходимой последовательности действий; соблюдает правила техники безопасности; правильно выполняет анализ возможных ошибок; обнаруживает понимание материала, излагает материал последовательно и правильно с точки зрения норм литературного языка.

Оценка «хорошо» ставится, если обучающийся выполнил требования к оценке «5», но допускает 1 – 2 ошибки, которые сам же исправляет, и 1 – 2 недочета в последовательности и логичности излагаемого.

Оценка «удовлетворительно» ставится, если работа выполнена не полностью, но реализованной части достаточно для получения правильных результатов и выводов; обучающийся обнаруживает знание и понимание основных положений данной темы, но: излагает материал неполно и допускает неточности в определении понятий или их трактовке.

Оценка «2» ставится, если работа выполнена не полностью или объем выполненной части работы не позволяет сделать правильных выводов и получить правильных результатов; обучающийся обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и терминов, искажающие их смысл, беспорядочно и неуверенно излагает материал.

Типовые контрольные задания и методические материалы для текущего и промежуточного контроля

Примерные устные вопросы

по теме «Современные технологии разработки программного обеспечения»

№	Вопрос	Ответ
1.	Что такое интегрированная среда разработки?	Интегрированная среда разработки (англ. IDE, Integrated development environment) — система программных средств, используемая программистами для разработки программного обеспечения.
2.	Какие инструменты входят	В интегрированную среду разработки входят инструменты:

	в интегрированную среду разработки?	редактор кода; компилятор и/или интерпретатор; отладчик; инструменты для конструирования графического интерфейса пользователя.
3.	Visual Studio .NET это?	Интегрированная среда разработки программного обеспечения компании Microsoft позволяющая разрабатывать: консольные приложения; приложения с графическим интерфейсом; веб-сайты; веб-приложения; веб-службы.
4.	Назначение программной платформы?	Программная платформа предназначена для предоставления среды исполнения прикладным программам. Например, организует порядок запуска программы, схему использования ею адресного пространства, взаимодействие с файловой системой.
5.	Что такое .NET Framework?	.NET Framework — программная платформа компании Microsoft. Она обеспечивает общеязыковую среду исполнения Common Language Runtime (CLR), для языков программирования (Visual Basic, C#, F# и других).

Критерии оценки устных ответов обучающихся

Оценка «5» ставится, если обучающийся: 1) полно и аргументировано отвечает по содержанию задания; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры; 3) излагает материал последовательно и правильно.

Оценка «4» ставится, если обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.

Оценка «3» ставится, если обучающийся обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки.

Оценка «2» ставится, если обучающийся обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.

Типовые задачи по теме «Основы языка программирования Visual Basic .NET»

№	Задача	Решение
1.	Объявить переменную для имени группы.	Dim GroupName As String Dim Группа As String = "2ПКС-14-дХТК"
2.	Объявить константу для логотипа языка программирования и даты работы	Const Soft As String = "VB.NET" Const ДатаРаботы As Date = #1/31/2016#
3.	Определить значение переменной x в результате работы программы: Dim x As Byte = 1 x = 2 x = x + 1 x = 2 + x * 2 x = x / 4 x += 2	x = 5

	<pre>x -= 3 x *= 5</pre>	
4.	<p>Определить результат составного условия $x > y$ И $x \neq z$ ИЛИ $y = z$, если $x = 2, y = 5,9, z = 7,1$.</p>	ЛОЖЬ
5.	<p>Запишите шаблон для оператора Like фильтрующего строки "1ГЭМ-15", "2ПКС-14".</p>	"[12]???-1[45]"
6.	<p>Определить значение переменной x в результате выполнения программного кода: Dim x As Byte = 3, y As Byte = 4, z As Byte = 5 Select Case x Case 4 To 6, Is > 7, y + z x = 10 Case Is < 0, 0 To 2, z - y + 1 x = 20 Case Is >= 3 + 1, y * 2 - z x = 30 Case 3 x = 40 End Select</p>	x = 30
7.	<p>Объявите одномерный массив с элементами целого типа.</p>	Dim A(4) As Integer
8.	<p>Инициализируйте одним оператором пять элементов одномерного массива.</p>	A = {0.2, 1, 11, 7, 13}
9.	<p>Добавьте три элемента к одномерному массиву и сохраните "старые" значения элементов.</p>	<pre>Dim A(4) As Integer '... ReDim Preserve A(UBound(A) + 3)</pre>
10.	<p>Записать фрагмент кода для блок-схемы:</p>	<pre>If x > y Then If y <> z Then If b And c Then c = Not a Else c = a Or b a = b Or c End If Else b = a Or c End If Else a = b And c If b And c Then c = Not a Else c = a Or b a = b Or c End If End If</pre>

Критерии оценки

Оценка «5» ставится, если в логических рассуждениях и решении нет ошибок, задача решена рациональным способом, правильно оформлена.

Оценка «4» ставится, если в логических рассуждениях и решении нет существенных ошибок, но задача решена нерациональным способом, либо допущено не более двух несущественных ошибок.

Оценка «3» ставится, если допущена существенная ошибка в математических расчётах.

Оценка «2»: имеются существенные ошибки в логических рассуждениях и в решении, либо отсутствует ответ на задание.

Примерные тестовые вопросы по теме «Основы языка программирования Visual Basic.NET»

1	Название языка программирования Basic происходит от выражения	* Beginner's All-purpose Symbolic Instruction Code Beginner Drivers Of All Car Brands Beginner Player In Computer Games A Person Begins To Speak The Language
2	Модуль (Module)	* Место для определения переменных, подпрограмм (функций, процедур, свойств), событий. Оператор для объявления переменных. Точка входа для начала выполнения программы.
3	Процедура Main()	* Точка входа для начала выполнения программы. Место для определения переменных, подпрограмм (функций, процедур, свойств), событий. Оператор для объявления переменных.
4	Переменная	* Поименованная область памяти служащая для размещения данных. Подпрограмма специального вида, которая всегда должна возвращать результат. Не изменяемая область памяти служащая для хранения данных.
5	Константа	* Не изменяемая область памяти служащая для хранения данных. Диапазон значений которой состоит из адресов ячеек памяти. Поименованная область памяти служащая для размещения данных.
6	Выберите правильные имена переменных	* X1 * Долгота_20 * _КолиныДеньги * SuperPrice * X_2_0_Y 2MyVariable
7	Объявление переменных целого типа	* Dim RealVar As Integer * Dim MyByte As Byte * Dim MaxValue As Long Dim Float As Double Const MyFlag As Boolean = True
8	Объявление вещественных	* Dim Количество As Single

	переменных	* Dim Производство As Double Dim Сумма As Byte Dim Product As String Dim Объем As Long
9	Укажите правильно объявленные и инициализированные переменные и константы	* Dim Bound As Single * Dim GoodDay As Date * Dim ДатаРождения As Date = #31/12/2015# * Const ОднаПословица As String = "На юге теплее, зато на севере холоднее!" * Const Постоянная_Величина As Double = 1.2 Dim b Byte
10	Определите соответствие между типами данных и размерами занимаемой памяти	Byte 1 байт Integer 4 байта Double 8 байт Char 2 байта
11	Какие из операторов выполняют операцию присваивания?	* = * += * -= * *= * /= * \ * ^= <> >= >=
12	Какие из операторов являются операторами сравнения?	* > * > * <> * >= * <= Mod Not OrElse AndAlso
13	Установите соответствие между оператором и названием арифметического действия	+ Сложение - Вычитание * Умножение / Деление \ Целочисленное деление Mod Остаток от деления ^ Возведение в степень
14	Укажите приоритет выполнения операций. Большее число соответствует высшему приоритету.	1 +, - 2 Mod 3 \ 4 *, / 5 ^
15	Выражение для логической функции: f = НЕ ((а ИЛИ b) И (с ИЛИ d)) ИЛИ НЕ(НЕ e ИЛИ g)	* f = Not ((a Or b) And (c Or d)) Or Not (Not e Or g) * f = Not ((a OrElse b) AndAlso (c OrElse d)) OrElse Not (Not e

	<pre> OrElse g) f = Not ((a And b) Or (c And d)) And Not (Not e And g) f = Not (a Or b) And (c Or d) Or Not Not e Or g f = Not a Or b And c Or d Or Not Not e Or g </pre>
--	---

Критерии оценки

Процент результативности (правильных ответов)	Оценка
95 ÷ 100	5
82 ÷ 94	4
61 ÷ 81	3
60% и менее	2

Темы для дискуссий:

1. Язык программирования INTERCAL.
2. Язык программирования CaneCode.
3. Язык программирования Ook!
4. Язык программирования COW.
5. Язык программирования Whitespace.
6. Язык программирования Spoon.
7. Язык программирования LOLCODE.
8. Язык программирования Byter.
9. Язык программирования Befunge.
10. Язык программирования Befunge-93.
11. Язык программирования Unefunge.
12. Язык программирования Trefunge.
13. Язык программирования 4DL.
14. Язык программирования Piet.
15. Язык программирования Thue.
16. Язык программирования Unlambda.
17. Язык программирования Smetana.
18. Язык программирования SMITH.
19. Язык программирования Muriel.
20. Язык программирования Malbolge.
21. Язык программирования ALPACA.
22. Язык программирования Chef.
23. Язык программирования Shakespeare.
24. Язык программирования Haifu.
25. Язык программирования ArnoldC.
26. Язык программирования FiM++.
27. Язык программирования Var'aq.
28. Язык программирования P''.
29. Язык программирования QuakeC.
30. Язык программирования MC++.
31. Язык программирования Автокод «Инженер».
32. Язык программирования Cool.
33. Язык программирования GOTO++.
34. Язык программирования Lithp.
35. Язык программирования paranoid.
36. Язык программирования SARTRE.
37. Язык программирования SIMPLE.
38. Язык программирования HQ9+/HQ9++.
39. Язык программирования Whenever.
40. Язык программирования smilesript.

Кейс-задания

1. Описать класс, реализующий стек. Написать программу, использующую этот класс для моделирования T-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.
2. Описать класс, реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа ко всем элементам. Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.
3. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.
4. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность отдельного изменения составных частей адреса, создания и уничтожения объектов этого класса. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
5. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
6. Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
7. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
8. Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта, возможность выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, вывода на экран элемента массива по заданному индексу, вывода на экран всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
9. Составить описание класса для определения одномерных массивов строк фиксированной длины. Предусмотреть возможность обращения к отдельным строкам массива по индексам, контроль выхода за пределы массива, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, вывод на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
10. Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения, вычитания и умножения многочленов с получением нового объекта-многочлена, вывод на экран описания многочлена. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
11. Составить описание класса одномерных массивов строк, каждая строка задается длиной и указателем на выделенную для нее память. Предусмотреть возможность обращения к отдельным

- строкам массива по индексам, контроль выхода за пределы массивов, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, вывод на экран элемента массива и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
12. Составить описание класса, обеспечивающего представление матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
 13. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
 14. Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
 15. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
 16. Описать класс «множество», позволяющий выполнять основные операции — добавление и удаление элемента, пересечение, объединение и разность множеств. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
 17. Описать класс, реализующий стек. Написать программу, использующую этот класс для отыскания прохода по лабиринту. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. После отыскания прохода программа печатает найденный путь в виде координат квадратов.
 18. Описать класс «предметный указатель». Каждый компонент указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до десяти. Предусмотреть возможность формирования указателя с клавиатуры и из файла, вывода указателя, вывода номеров страниц для заданного слова, удаления элемента из указателя. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.
 19. Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.
 20. Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие заданное с клавиатуры слово.

Пример практической работы №1 по теме «Расчёт переменных»

Цели

Практическое знакомство с интегрированной средой разработки программных продуктов Visual Studio и языком программирования Visual Basic. Получение основных практических навыков программирования алгоритмов линейной структуры (создание, отладка и выполнение программы).

Задание для практической работы

Составить программу для расчета значений переменных по их выражениям. Определить значения переменных по заданным исходным данным. В программе обеспечить ввод данных и отображение результатов.

Порядок выполнения

1. Составить алгоритм решения задачи.
2. Определить значения результатов по тестовому набору исходных данных.
3. Создать проект программы в среде программирования.
4. Выполнить отладку и запуск программы с тестовыми исходными данными.

Краткое изложение темы

Программа алгоритма линейной структуры – это программа, в которой блоки выполняются последовательно друг за другом.

В настоящей работе программа линейной структуры реализует действия для расчета переменных по математическим формулам. В коде программы выделяются следующие элементы: объявление переменных; ввод исходных данных; вычисление по формулам; вывод результата.

При объявлении переменной необходимо решить, как она будет называться, и какой тип данных будет ей присвоен. Имя переменной может быть любым, но оно должно начинаться с буквы или знака подчеркивания. Если используется имя, описывающее содержащееся в переменной значение, то код будет легче читать. Например, переменная, позволяющая отслеживать число леденцов в банке, может называться КоличествоЛеденцов.

Объявляется переменная, при помощи ключевых слов **Dim** и **As**, например, **Dim** Number **As Integer**. Эта строка кода сообщает программе, что требуется использовать переменную с именем Number, и что она должна хранить целые числа (тип данных **Integer**)

Для считывания исходных данных в переменную используется метод Console.ReadLine().

Методы класса Math предоставляют тригонометрические, логарифмические и другие общие математические функции для записи формул (Math.Sin, Math.Sqrt, Math.Pow).

Для вывода результатов работы программы используется метод Console.WriteLine, аргументы которого формируют строку, отображающуюся на экране.

Пример выполнения задания

Даны выражения для расчёта переменных

$$\Psi = \sqrt{\frac{x+y}{y+z}} + \sin^3\left(\frac{\pi}{2} + 1\right)^2 + \operatorname{ctg}^2 x^3 + e^{x \cdot (-y)}$$

$$\omega = \sqrt[3]{|x^y - 0,5|} + \lg \left| \frac{x+1}{y+1} \right| + \log_2 x$$

Решение

```

Module1.vb
Module1
Main
'*****
'* Практическая работа №1
'* Выполнил: студент группы 2ПКС, Иванов И.И.
'* Условие.
'* Записать программу для расчета переменных.
'* *****
Module Module1
    Dim aNumber As Integer
    Sub Main()
        'Объявление констант
        Const pi As Double = 3.1415926535897931
        'Объявление переменных и ввод исходных данных
        Dim x As Double = Console.ReadLine(),
            y As Double = Console.ReadLine(),
            z As Double = Console.ReadLine()
        'Дополнительные переменные
        Dim v1, v2, v3, v4 As Double
        'Расчет значений
        v1 = (x + y) / (y + z)
        v1 = Math.Sqrt(v1)

        v2 = pi / 2 + 1
        v2 = Math.Pow(v2, 2)
        v2 = Math.Sin(v2)
        v2 = Math.Pow(v2, 3)

        v3 = Math.Pow(x, 3)
        v3 = 1 / Math.Tan(v3)
        v3 = Math.Pow(v3, 2)

        v4 = Math.Exp(x * (-y))

        Dim Кси As Double = v1 + v2 + v3 + v4

        v1 = Math.Pow(x, y) - 0.5
        v1 = Math.Abs(v1)
        v1 = Math.Pow(v1, 1.0 / 3.0)

        v2 = (x + 1) / (y + 1)
        v2 = Math.Abs(v2)
        v2 = Math.Log10(v2)
        v3 = Math.Log(x, 2)

        Dim Омега As Double = v1 + v2 + v3

        'Вывод на экран значений
        Console.WriteLine("Кси = {0:#####}, Омега = {0:#####}",
            Кси, Омега)

        Console.ReadKey()
    End Sub
End Module

```

Тестовый расчет

x	y	z	Ψ	ω
1,0	2,0	3,0	1,355	0,618

Пример практической работы №2 по теме «Программирование алгоритмов ветвления»

Цели

Получение основных практических навыков по работе с программами разветвляющейся структуры на языке программирования Visual Basic (создание, отладка, выполнение).

Задание для практической работы

Составить программу для задачи алгоритма разветвляющейся структуры. Подобрать необходимые и достаточные наборы исходных данных для тестирования всех ветвей программы. Обеспечить ввод исходных данных и отображение результатов.

Порядок выполнения

1. Составить алгоритм решения задачи.
2. Подобрать необходимые и достаточные наборы исходных данных для тестирования всех ветвей алгоритма и определить по ним результаты.
2. Создать проект программы в среде программирования.
3. Выполнить отладку и запуски программы со всеми тестовыми наборами исходных данных.

Краткое изложение темы

Программа алгоритма разветвляющейся структуры – это программа, в которой в зависимости от истинности или ложности некоторого условия выбирается одна из двух серий команд.

Для реализации программ с ветвлением в Visual Basic имеются специальные операторы: If...Then...Else и Select...Case, которые в зависимости от значений условного выражения передают управление в программе на тот или иной блок команд.

Очень важно при отладке программы с ветвлением пройти по всем веткам алгоритма, для этого необходимо составить такие наборы исходных данных, которые реализовывали бы все комбинации исходов условных выражений.

Пример выполнения задания

Условие.

Реализовать эпизод применения компьютера в книжном магазине: компьютерная программа запрашивает общую стоимость книг и сумму, внесённую покупателем, если сдача не требуется, на экран выводится сообщение "Спасибо за покупку.", если денег внесено больше – "Получите, пожалуйста, сдачу" и указывается ее сумма, если денег недостаточно – "Извините, Вы не доплатили." и на экране печатается недостающая сумма.

Решение

```

Module1.vb
(Общие) (Объявления)
'*****
'* Практическая работа №2 *
'* Выполнил: студент группы 2ПКС, Иванов И.И. *
'* Условие. *
'* Реализовать эпизод применения компьютера в книжном магазине: *
'* компьютерная программа запрашивает общую стоимость книг и сумму, *
'* внесённую покупателем, если сдача не требуется, на экран выводится *
'* сообщение "Спасибо за покупку.", если денег внесено больше - *
'* "Получите, пожалуйста, сдачу" и указывается ее сумма, если денег *
'* недостаточно - "Извините, Вы не доплатили." и на экране печатается *
'* недостающая сумма. *
'*****
Module Module1
Sub Main()
    Console.WriteLine("Здравствуйте!" & ControlChars.CrLf)

    Повторить:
    Console.Write("Введите общую стоимость книг, руб.: ")
    'Объявление переменной и присвоение значения
    Dim СтоимостьКниги As Single = Convert.ToSingle(Console.ReadLine())

    Console.Write("Введите Вашу сумму оплаты, руб.: ")
    Dim СуммаОплаты As Single = Convert.ToSingle(Console.ReadLine())

    Dim x As Single = СуммаОплаты - СтоимостьКниги,
        y As Single = Math.Abs(x)

    'Проверка условий и определение направления вычисления алгоритма
    If x < 0 Then
        Console.WriteLine("Извините, Вы не доплатили: {0:#.##} руб.", y)
    ElseIf x > 0 Then
        Console.WriteLine("Получите, пожалуйста, сдачу: {0:#.##} руб.", y)
    Else
        Console.WriteLine("Спасибо за покупку.")
    End If

    Console.WriteLine(ControlChars.CrLf &
        "Если Вы хотите продолжить работу нажмите клавишу (д)," &
        ControlChars.CrLf &
        "выйти из программы - любая другая клавиша.")
    Dim c As ConsoleKeyInfo = Console.ReadKey()

    If c.Key = ConsoleKey.L Then
        Console.Clear()
        GoTo Повторить
    End If
End Sub
End Module

```

Тестовый расчет

№	Стоимость книг	Внесенная сумма оплаты	Сообщение на экране
1	110	110	"Спасибо за покупку."
2	100,2	110	"Получите, пожалуйста, сдачу: 9,8 руб."
3	100,2	90	"Извините, Вы не доплатили: 10,2 руб."

Критерии оценки:

- 1) полнота и правильность решения задания;
- 2) последовательность и логичность выполнения действий при выполнении действий;
- 3) владение терминологическим аппаратом и использование его при ответе (пояснение при демонстрации работы).

Оценка «отлично» ставится, если обучающийся выполнил работу в полном объеме с соблюдением необходимой последовательности действий; соблюдает правила техники безопасности; правильно выполняет анализ возможных ошибок; обнаруживает понимание материала, излагает материал последовательно и правильно с точки зрения норм литературного языка.

Оценка «хорошо» ставится, если обучающийся выполнил требования к оценке «5», но допускает 1 – 2 ошибки, которые сам же исправляет, и 1 – 2 недочета в последовательности и логичности излагаемого.

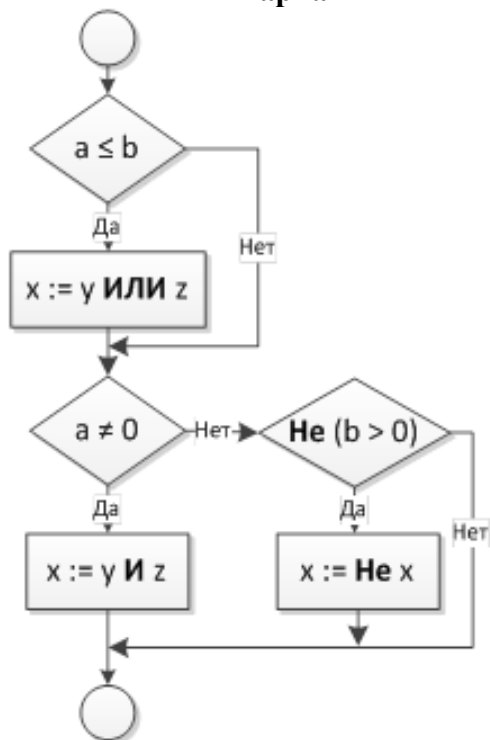
Оценка «удовлетворительно» ставится, если работа выполнена не полностью, но реализованной части достаточно для получения правильных результатов и выводов; обучающийся обнаруживает знание и понимание основных положений данной темы, но: излагает материал неполно и допускает неточности в определении понятий или их трактовке. Оценка «2» ставится, если работа выполнена не полностью или объем выполненной части работы не позволяет сделать правильных выводов и получить правильных результатов; обучающийся обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и терминов, искажающие их смысл, беспорядочно и неуверенно излагает материал.

Пример заданий для внеаудиторной самостоятельной работы

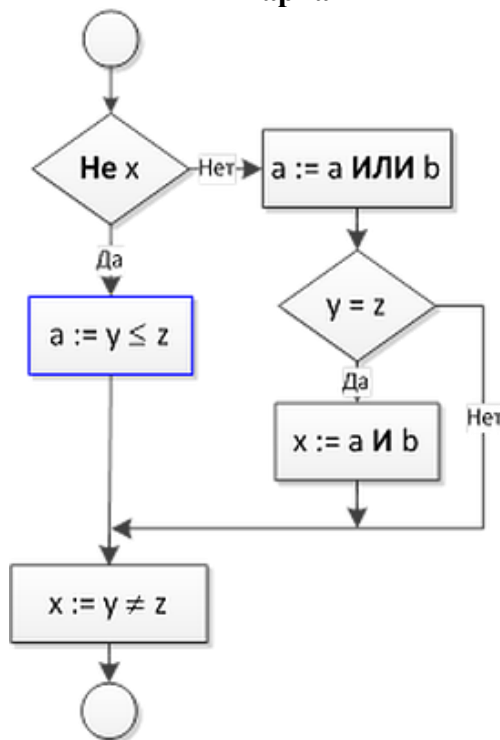
Условие задания

Создать консольную программу, работающую по алгоритму. В программе обеспечить ввод исходных данных и отображение результатов. Произвести пошаговую отладку и запуск программы в среде программирования.

Вариант 1



Вариант 2



Решение.

Вариант 1

```

Module1.vb*
Module1 (Объявления)
Sub Main()
    Dim a, b As Double
    Dim x, y, z As Boolean

    a = Console.ReadLine()
    b = Console.ReadLine()
    y = Console.ReadLine()
    z = Console.ReadLine()

    If a <= b Then x = y OrElse z
    If a <> b Then
        x = y AndAlso z
    Else
        If Not b > 0 Then x = Not x
    End If
End Sub
    
```

Вариант 2

```

Module1.vb*
Module1 (Общие) (Объявления)
Sub Main()
    Dim a, b, x As Boolean
    Dim y, z As Double

    a = Console.ReadLine()
    b = Console.ReadLine()
    x = Console.ReadLine()
    y = Console.ReadLine()
    z = Console.ReadLine()

    If Not x Then
        a = y <= z
    Else
        a = a OrElse b
        If y = z Then x = a AndAlso
    End If
    x = y <> z
End Sub
    
```

Критерии оценки:

- 1) полнота и правильность решения задания;
- 2) последовательность и логичность выполнения действий при выполнении действий;
- 3) владение терминологическим аппаратом и использование его при ответе (пояснение при демонстрации работы).

Оценка «отлично» ставится, если обучающийся выполнил работу в полном объеме с соблюдением необходимой последовательности действий; соблюдает правила техники безопасности; правильно выполняет анализ возможных ошибок; обнаруживает понимание материала, излагает материал последовательно и правильно с точки зрения норм литературного языка.

Оценка «хорошо» ставится, если обучающийся выполнил требования к оценке «5», но допускает 1 – 2 ошибки, которые сам же исправляет, и 1 – 2 недочета в последовательности и логичности излагаемого.

Оценка «удовлетворительно» ставится, если работа выполнена не полностью, но реализованной части достаточно для получения правильных результатов и выводов; обучающийся обнаруживает знание и понимание основных положений данной темы, но: излагает материал неполно и допускает неточности в определении понятий или их трактовке.

Оценка «2» ставится, если работа выполнена не полностью или объем выполненной части работы не позволяет сделать правильных выводов и получить правильных результатов; обучающийся обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и терминов, искажающие их смысл, беспорядочно и неуверенно излагает материал.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.

Приступая к изучению дисциплины студенту необходимо внимательно ознакомиться с тематическим планом занятий, списком рекомендованной литературы.

Основными видами аудиторной работы студентов являются урок и практические занятия.

В ходе урока преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы. Во время занятий необходимо вести конспект. Преподаватель дает на уроке задания для закрепления пройденного материала, организует и оказывает студенту помощь в самостоятельной работе во время урока, дает рекомендации на подготовку к практической работе и указания на выполнение домашней работы. Во время урока преподаватель также проводит проверку теоретических знаний по теме

прошлого урока. Активное участие студента во всех этапах занятия, позволит ему качественно усвоить необходимый теоретический и практический материал, разобраться в основных вопросах и получить дополнительные необходимые для понимания и дальнейшей практической деятельности рекомендации преподавателя.

Целями выполнения практических работ является:

- 1) обобщение, систематизация, углубление, закрепление полученных теоретических знаний по конкретным темам;
- 2) формирование умений применять полученные знания на практике, реализация единства интеллектуальной и практической деятельности;
- 3) развитие интеллектуальных умений у будущих специалистов; аналитических, проектировочных, конструктивных и др.
- 4) выработку при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

Практические занятия вырабатывают у студентов навыки применения полученных знаний для решения профессиональных практических задач. На практических занятиях студенты выполняют тренировочные упражнения, решают задачи по созданию компьютерных программ и приложений, разбирают и моделируют производственные ситуации.

Для выполнения практических работ студентам выдается сборник практических работ. Каждая инструкция в сборнике содержит цель работы, перечень программного обеспечения, ход выполнения работы, обращающие внимание студентов на существенные стороны изучаемых тем. Вопросы помогают глубже осмыслить производимые действия и полученные результаты и на их основе самостоятельно сделать необходимые выводы.

В ходе работы необходимо строго соблюдать правила охраны труда; всю работу производить с максимальной тщательностью.

После окончания работы каждый студент представляет работающую программу, программный код которой должен быть отформатирован с отступами блоков и содержать подробные комментарии.

В конце занятия преподаватель ставит оценку, которая складывается из результатов наблюдения за выполнением практической части работы, проверки программы, беседы в ходе работы или после нее.

Типовая инструкция по охране труда для студентов

1. Будьте внимательны и дисциплинированы
2. Не приступайте к выполнению работы без разрешения преподавателя.
3. Размещайте устройства ввода/вывода персонального компьютера на своем рабочем месте таким образом, чтобы исключить их падение или опрокидывание.
4. Перед выполнением работы необходимо внимательно изучить ее содержание и ход выполнения.

Для успешной подготовки к практическим занятиям студенту необходима предварительная самостоятельная работа по теме планируемого занятия: работа над конспектом, учебником, учебным пособием, интернет -ресурсами, чтобы основательно овладеть теорией вопроса.

В ходе изучения дисциплины предусмотрена внеаудиторная (домашняя) самостоятельная работа в объеме 96 часов.

Внеаудиторная самостоятельная работа выполняется студентами в целях:

- углубление и расширение теоретических знаний;
- формирование умений использовать справочную документацию и специальную литературу;
- развитие познавательных способностей и активности обучающихся;
- формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации;
- развитие исследовательских умений;
- умение использовать материал, собранный и полученный в ходе самостоятельных занятий для решения практических задач.

Внеаудиторная самостоятельная работа дополняет содержание аудиторных занятий, способствует закреплению, обобщению и систематизации полученных на уроках теоретических знаний и совершенствованию практических умений, а также развитию таких качеств личности, как ответственность и организованность.

Объем времени для выполнения учебного задания определен эмпирически - на основании наблюдений за выполнением студентами аудиторной самостоятельной работы; на основе опроса студентов о затратах времени на выполнение того или иного внеаудиторного задания; на основе хронометража собственных затрат преподавателя на решение той или иной задачи с внесением поправочного коэффициента из расчета уровня знаний и умений студента по дисциплине.

Оценка за выполнение домашнего задания выставляется в журнал учебных занятий.

Дополнительные занятия и консультации позволяют студенту восполнить пробелы в знаниях под руководством преподавателя, выполнить пропущенную работу, за которую должна стоять оценка, повысить оценку, обсудить вопросы, направленные на углубленное изучение темы, получить консультацию преподавателя по теме научно-исследовательской работы.

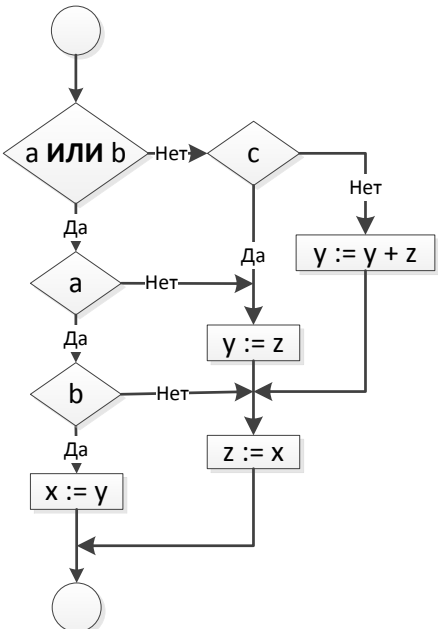
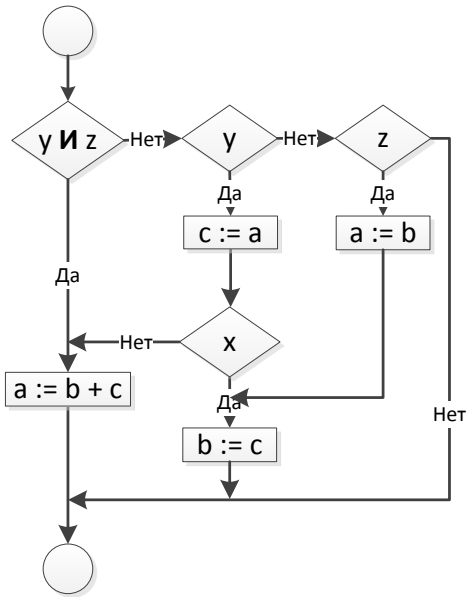
Технологическая карта практических работ

№ занятия	Тема практической работы	Кол. часов	Задание	Литература со стр.
1-2	Программирование алгоритмов линейной структуры.	4	Составить программу для расчета значений переменных по их выражениям. Определить значения переменных по заданным исходным данным. В программе обеспечить ввод исходных данных и отображение результатов.	[1, с. 3]
3-4	Программирование алгоритмов разветвляющейся структуры.	4	Составить программу для задачи алгоритма разветвляющейся структуры. Подобрать необходимые и достаточные наборы исходных данных для тестирования всех ветвей программы. Обеспечить ввод исходных данных и отображение результатов.	[1, с. 7,11]
5	Программирование с обработкой ошибок времени выполнения.	2	Внедрить в программу алгоритма разветвляющейся структуры код обработки ошибок времени выполнения. Реализовать структурную обработку ошибок для нескольких видов исключений: переполнение (OverflowException), преобразование типов (InvalidCastException), деление на ноль (DivideByZeroException).	[1, с. 14]
6-8	Программирование алгоритмов циклической структуры.	6	Составить программу алгоритма циклической структуры для обработки одномерного массива. Обеспечить ввод размерности и элементов массива, отобразить результаты. Подобрать варианты тестовых наборов исходных данных для проверки правильности работы программы.	[1, с. 19]
9-11	Программирование алгоритмов циклической структуры с обработкой ошибок времени выполнения.	6	Добавить в программу для одномерного массива код обработки ошибок времени выполнения. Подобрать варианты тестовых наборов исходных данных для тестирования процесса обработки ошибок времени выполнения в циклических алгоритмах.	[1, с. 23]
12	Программирование алгоритмов циклической структуры с условным числом повторений.	2	Составить программу для задачи нахождения корня уравнения методом половинного деления. Обеспечить ввод границ отрезка для нахождения корня и отображение, вычисленного с заданной точностью, корня.	[1, с. 26]
13	Программирование алгоритмов циклической структуры с условным числом повторений.	2	Составить программу для задачи вычисления суммы числового ряда. Обеспечить ввод исходных данных (аргументов) и отображение, вычисленной суммы с заданной точностью.	[1, с. 32]
14-17	Программирование алгоритмов со структурой вложенных циклов.	8	Составить программу алгоритма со структурой вложенных циклов для обработки двумерного массива. Обеспечить ввод размерностей и элементов массива, отобразить результаты. Подобрать варианты тестовых наборов исходных данных для проверки правильности работы программы.	[1, с. 36]
18-19	Программирование подчиненных алгоритмов.	4	Составить программу выполняющую подчиненный алгоритм обработки текстовых данных внутри подпрограммы. Подобрать варианты тестовых наборов исходных данных для проверки правильности работы программы.	[1, с. 41]
20-22	Программирование алгоритмов обработки текстовых данных.	6	Составить программу выполняющую обработку текстовых данных с использованием: специальных операторов, стандартных и пользовательских подпрограмм. Подобрать варианты тестовых наборов исходных данных для проверки правильности работы программы.	[1, с. 46]
23-24	Программирование перечисляемых типов.	4	Составить программу с использованием перечисляемых типов для логически связанных друг с другом значений. Объявить массив исходных данных с количеством элементов соответствующих перечисляемому типу. Выполнить обработку массива в соответствии с условием задачи.	[1, с. 54]
25-27	Программирование структур данных.	6	Составить программу с объявлением структуры (типа данных) описывающей суть элемента предметной области задачи. Структура должна содержать не менее пяти полей и трех методов. На основе типа структуры создать массив элементов и запрограммировать обработку данных.	[1, с. 59]
28-32	Объектно-ориентированное программирование. Программирование	10	Составить программу с описанием данных предметной области задачи в виде объектной модели классов. В описании использовать механизмы инкапсуляции и наследования. Запрограммировать	[1, с. 67]

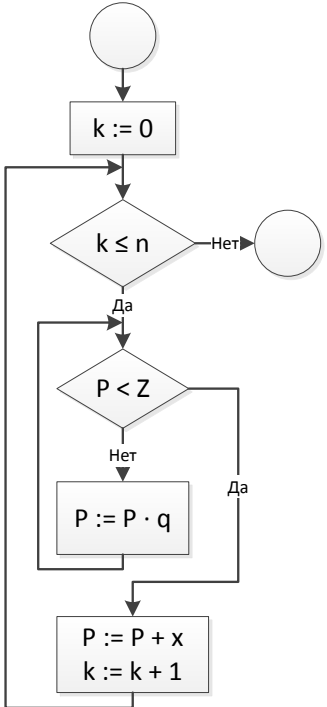
	классов.		методы для ввода полей классов модели. Решить задачу по обработке данных объектной модели. Количество классов в модели должно быть не менее трех, суммарное количество полей всех классов – не менее двенадцати.	
33-37	Программирование подсистемы потокового ввода/вывода.	10	Дополнить классы объектной модели, описывающей данные предметной области, методами чтения и записи данных в поток. Запрограммировать чтение и запись данных объектной модели в текстовый файл.	[1, с. 76]
38-40	Формы, элементы управления.	6	Создать графический интерфейс приложения для расчета числовых переменных. Запрограммировать события нажатия кнопок для выполнения расчета, очистки текстовых окон и выхода из приложения.	[1, с. 88]
41-43	Формы, элементы управления.	6	Создать графический интерфейс приложения для обработки одномерного массива. Запрограммировать события нажатия кнопок для обработки, очистки текстовых окон и выхода из приложения.	[1, с. 93]
44-46	Формы, элементы управления.	6	Создать графический интерфейс приложения для работы с массивом структур данных. Запрограммировать события нажатия кнопок для манипулирования массивом структур данных.	[1, с. 98]
47-49	Формы, элементы управления.	6	Расширить функциональные возможности графического приложения для работы с массивом структур данных. Запрограммировать события нажатия кнопок для выполнения заданий на обработку структур данных.	[1, с. 106]
50-51	Коллекции.	4	В коде графического приложения для обработки одномерного массива выполнить замену одномерного массива на соответствующую универсальную коллекцию типа List(Of T).	[1, с. 117]
52-54	Коллекции.	6	Модифицировать код графического приложения для обработки массива структур, выполнив замену одномерного массива структур на универсальную коллекцию типа List(Of T). В приложение добавить функции сортировки и поиска структур данных по предикату.	[1, с. 121]

Задания для самостоятельной работы обучающихся

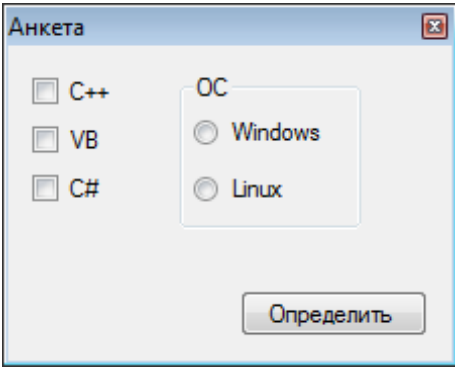
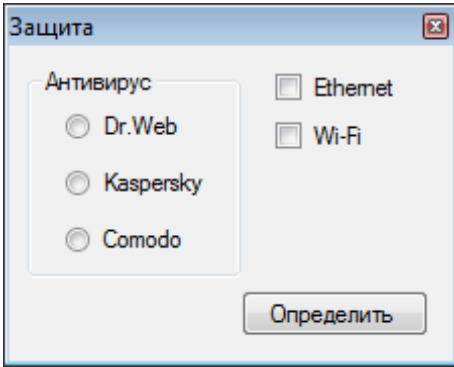
№ задания	Номер, наименование разделов, тем	Вид внеаудиторной самостоятельной работы	Задания для внеаудиторной самостоятельной работы	Примерный объем времени на выполнение, в час.
	Раздел 1. Современные технологии разработки программного обеспечения			
1	Тема 1.1. Современные технологии разработки программного обеспечения	Изучение этапов решения задач на компьютере.	Используя конспект и дополнительную литературу (см. список литературы) определить: 1) этапы решения задачи на компьютере, где определяется правильность полученных промежуточных и итоговых результатов; 2) этапы решения задачи на компьютере, когда происходит работа в текстовом редакторе и используются специальные инструменты.	2
	Раздел 2. Язык программирования Visual Basic .NET			

2	Тема 2.1. Основы языка программирования Visual Basic .NET	Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Объявление переменных и констант.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задания: 1) объявить переменную для записи данных о стоимости 1 кг продукта питания; 2) объявить константу для скорости света; 3) объявить переменную для записи сведений о возрасте человека; 4) объявить константу максимального количества курсов обучения группы 2ПКС-14.	4
3		Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Операторы сравнения, логические. Простые и составные условные выражения.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задания: 1) объявить переменную f и присвоить ей значение выражения $f = (x \neq y) \text{ ИЛИ } (z \geq 5)$, здесь x, y, z – целые числа от 0 до 100; 2) объявить переменную f и присвоить ей значение выражения $f = \text{НЕ} ((a < c) \text{ И } (a \neq b))$, здесь a, b, c – вещественные числа в интервале (-1000;500).	4
4		Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Операторы ветвления.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задание. Запрограммировать фрагмент алгоритма (не использовать оператор GoTo). 1.  2. 	8

5		Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Одномерные массивы.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задания: 1) объявить и инициализировать одномерный массив с элементами целого типа (длинное целое); 2) уменьшить массив на $x * 2$ элементов и сохранить оставшиеся в нем значения; 3) объявить и инициализировать одномерный массив с логическими элементами; 4) увеличить массив на $z \setminus 5$ элементов и сохранить оставшиеся в нем значения.	4
6		Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Операторы циклов с заданным числом повторений.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задание. Запрограммировать фрагмент алгоритма используя, только операторы циклов. 1.  <pre> graph TD Start(()) --> I0[i := 0] I0 --> IM{i <= m} IM -- Да --> SD[S := S + d i := i + 1] SD --> IM IM -- Нет --> DW[d := d^2] DW --> SW{S < W} SW -- Да --> IM SW -- Нет --> End(()) </pre> 2.  <pre> graph TD Start(()) --> K0[k := 0] K0 --> KN{k <= n} KN -- Да --> PZ{p < Z} PZ -- Да --> PQ[P := P * q] PZ -- Нет --> PX[P := P + x k := k + 1] PX --> PZ KN -- Нет --> End(()) </pre>	10

7		<p>Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Операторы циклов с условным числом повторений.</p>	<p>Используя конспект и дополнительную литературу (см. список литературы) выполнить задание. Запрограммировать фрагмент алгоритма используя, только операторы циклов (объявлять переменные не нужно).</p> <p>1.</p>  <pre> graph TD Start(()) --> I0[i := 0] I0 --> IM{i <= m} IM -- Да --> Splus[S := S + d i := i + 1] Splus --> IM IM -- Нет --> Dsq[d := d^2] Dsq --> SW{S < W} SW -- Да --> IM SW -- Нет --> End(()) </pre> <p>2.</p>  <pre> graph TD Start(()) --> K0[k := 0] K0 --> KN{k <= n} KN -- Да --> PZ{P < Z} PZ -- Да --> Pmult[P := P * q] Pmult --> PZ PZ -- Нет --> Pinc[P := P + x k := k + 1] Pinc --> KN KN -- Нет --> End(()) </pre>	10
8		<p>Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Многомерные массивы.</p>	<p>Используя конспект и дополнительную литературу (см. список литературы) выполнить задания:</p> <ol style="list-style-type: none"> 1) создать трехмерный массив с элементами дробного; 2) увеличить количество слоев приблизительно на ~25% с сохранением значений; 3) определить количество элементов в массиве, не используя свойство Length; 4) создать двумерный массив для хранения дат; 5) сократить количество столбцов в массиве приблизительно на одну треть с сохранением значений; 6) определить количество элементов в массиве, не используя свойство Length. 	4

9		Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Строковые типы данных.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задания. Объявить и присвоить переменной s1 правильно по смыслу составленное предложение из элементов (слов) массива. 1. <code>Dim a() As String = {"новаторов", "есть", "из", "величайший", "Время"}</code> Объявить и присвоить переменной s2 подстроку из s1 начиная с 11 позиции и заканчивая 20-ой. Ответьте, что будет в s2? 2. <code>Dim b() As String = {"механизм", "действие", "мысли", "приводит", "юмор", "в"}</code> Объявить и присвоить переменной s2 подстроку из s1 начиная с 5 позиции и заканчивая 12-ой. Ответьте, что будет в s2?	4
10		Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Перечисления и структуры.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задания: 1) объявить перечисляемый тип для части суток; 2) объявить тип структуры для описания сущности животного; 3) объявить перечисляемый тип для летних месяцев года; 4) объявить тип структуры для описания сущности растения.	6
11	Тема 2.2. Объектно-ориентированное программирование	Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка программирования. Объектно-ориентированное программирование.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задание. Провести абстрагирование объекта. Выделить его 5 значимых свойств для программирования предметной области. Указать назначение 1-го метода для обработки свойств. Построить двухуровневую иерархию наследования. 1. Объект человек. Предметная область пациент больницы. 2. Объект животное. Предметная область питомец зоопарка.	8
12	Тема 2.3. Работа с файловой системой	Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка. Объекты файловой системы	Используя конспект и дополнительную литературу (см. список литературы) выполнить задание, написать код: 1) создания новой папки "W:\Account\Students\FIO"; 2) переименования папки "W:\Account\Students\FIO" в папку с именем "W:\Account\Students\MyFIO"; 3) удаления пустой папки на томе C: и создания папки на томе D:. Имя папки "\All\Docs1"; 4) переименования папки "Y:\My\Docs1" в папку с именем "Y:\My\Docs2".	10

13	Тема 2.4. Коллекции	Подготовка к практической работе по пройденному теоретическому материалу. Изучение основ языка. Коллекции.	Используя конспект и дополнительную литературу (см. список литературы) решить задачи. 1) Создать и добавить в коллекцию значения: "\$", "Евро", "ПИ", "Дата рождения". 2) Вставить во 2-ю позицию коллекции слово "ЛОЖЬ". Удалить последний элемент в коллекции. 3) Создать и добавить в коллекцию значения: 22.04.1870, "Ленин", 110, True 4) Удалить первый элемент в коллекции. Добавить в предпоследнюю позицию слово "История".	6
	Раздел 3. Windows Forms			
14	Тема 3.1. Формы	Подготовка к практической работе по пройденному теоретическому материалу. Изучение технологии Windows Forms.	Используя конспект и дополнительную литературу (см. список литературы) выполнить задание. Записать код для кнопки "Определить". 1.  2. 	6
15	Тема 3.2. Элементы управления		Используя конспект и дополнительную литературу (см. список литературы) подготовить реферат-презентацию по описанию назначения свойств и методов определенного элемента управления формы.	10
			Всего	96

