

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет  
им. Г. И. Носова»  
Многопрофильный колледж



**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ  
ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ  
АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

**ПМ. 01. РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ  
СИСТЕМ**


**программы подготовки специалистов среднего звена  
по специальности СПО**

**09.02.03 Программирование в компьютерных системах  
базовой подготовки**

Магнитогорск, 2017

## ОДОБРЕНО

Предметно-цикловой комиссией  
«Информатики и вычислительной  
техники»

Председатель  /И.Г. Зорина  
Протокол № 7 от 14 марта 2017  
Методической комиссией  
Протокол №4 от 23 марта 2017г

## СОГЛАСОВАНО

Председатель экспертной комиссии:

Начальник отдела  
ООО «ММК-Информсервис»

  
(подпись)

А.Ю.Газыгин

## Разработчик (и):

преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» Многопрофильный  
колледж Людмила Александровна Фетисова  
преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» Многопрофильный  
колледж Влада Дилиуровна Тутарова

Комплект контрольно-оценочных средств составлен на основе ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах базовой подготовки, утвержденного от 28 июля 2014 г. № 804, и рабочей программы профессионального модуля ПМ. 01. Разработка программных модулей программного обеспечения для компьютерных систем и предназначен для контроля и оценки образовательных достижений обучающихся.

## **СОДЕРЖАНИЕ**

1. Общие положения	4
2. Оценка освоения теоретического курса профессионального модуля	12
3. Контроль приобретения практического опыта. Оценка по учебной и (или) производственной практике	83
4. Контрольно-оценочные средства для экзамена (квалификационного)	94
Лист регистрации изменений и дополнений	98

## 1 ОБЩИЕ ПОЛОЖЕНИЯ

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности и составляющих его профессиональных компетенций, а также общие компетенции, формирующиеся в процессе освоения программы подготовки специалистов среднего звена в целом.

Формой аттестации по профессиональному модулю является экзамен (квалификационный). Экзамен (квалификационный) проводится в форме выполнения практико-ориентированного задания.

Итогом экзамена является однозначное решение: «вид профессиональной деятельности освоен / не освоен».

### 1.1 ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

Таблица 1.1

Элементы модуля	Формы промежуточной аттестации
1	2
МДК.01.01. Системное программирование	Дифференцированный зачет
МДК.01.02. Прикладное программирование	Дифференцированный зачет
Учебная практика	зачет
Производственная практика (по профилю специальности)	зачет
ПМ	Экзамен (квалификационный)

### 1.2 РЕЗУЛЬТАТЫ ОСВОЕНИЯ МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

#### 1.2.1 Профессиональные и общие компетенции

В результате контроля и оценки по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций

Таблица 1.2

<b>Профессиональные компетенции</b> (должны быть сформированы в полном объеме)	<b>Основные показатели оценки результата</b>
ПК 1.1. Выполнять разработку спецификаций отдельных компонент.	-Умение описывать внешнюю информационную среду, к которой должны применяться программы разрабатываемой ПС -Умение определять функции ПС, определенных на множестве состояний этой информационной среды (такие функции будем называть внешними функциями ПС); - Владение навыками описания нежелательных (исключительных) ситуаций, которые могут возникнуть при выполнении программ ПС, и реакций на эти ситуации, которые должны обеспечить соответствующие программы
ПК.1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля	- Умение правильно и обоснованно выбирать алгоритм с использованием средств автоматизированного проектирования -. Владение навыками корректного построения алгоритма в соответствии с заданной задачей -Умение разрабатывать код программного модуля на современных языках программирования.
ПК.1.3. Выполнять отладку программных модулей с использованием	-Умение разрабатывать спецификации этапов жизненного цикла программы в соответствии с

специализированных программных средств.	<p>поставленной задачей;</p> <ul style="list-style-type: none"> <li>- Умение использовать инструментальные средств на этапе отладки программного продукта;</li> <li>- Владеть навыками отладки программы на уровне модуля</li> </ul>
ПК.1.4. Выполнять тестирование программных модулей.	<ul style="list-style-type: none"> <li>- Владеть навыками тестирования программного модуля по разработанному сценарию;</li> <li>- Владеть навыками тестирования программы на уровне модуля;</li> <li>- Владеть навыками тестирования программы с использованием инструментальных средств</li> </ul>
ПК.1.5. Осуществлять оптимизацию программного кода модуля.	<ul style="list-style-type: none"> <li>- Умение осуществлять оптимизацию программного кода модуля по определенному сценарию;</li> <li>- Умение использовать методы для получения кода с заданной функциональностью и степенью качества;</li> <li>- Умение использовать инструментальные средства для оптимизации программного кода модуля</li> </ul>
ПК.1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.	<ul style="list-style-type: none"> <li>- Умение правильно определять и использовать методы и средства разработки технической документации;</li> <li>- Умение применять инструментальные средства для автоматизации оформления документации;</li> <li>- Умение разрабатывать отдельные компоненты документации на программные средства с использованием графических языков спецификаций</li> </ul>
ПК.1.7. Осуществлять работу с	<ul style="list-style-type: none"> <li>- Умение правильно применять</li> </ul>

системой контроля версий.	методы контроля качества объектно-ориентированного программирования; - Умение осуществлять рефакторинг и оптимизацию программного кода; - Умение работать с системой контроля версий.
---------------------------	---

Таблица 1.3

<b>Общие компетенции</b> <i>(возможна частичная сформированность)</i>	<b>Основные показатели оценки результата</b>
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес	Демонстрация интереса к будущей профессии
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 разработки автоматизированных информационных систем; оценка эффективности и качества выполнения;
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность	Решение стандартных и нестандартных профессиональных задач в области разработки автоматизированных информационных систем;
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития	Эффективный поиск необходимой информации; использование различных источников, включая электронные
ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности	Использование возможностей, предоставляемых информационно-коммуникационными технологиями для получения новых знаний и реализации умений в своей профессиональной деятельности

ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями	- наблюдение и оценка решения профессиональных задач на практических занятиях; - оценка результатов самостоятельной работы;
ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), 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 профессионального модуля
ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности	Анализ инноваций в области разработки информационных систем

Таблица 1. 4

<b>Профессиональные и общие компетенции, которые возможно сгруппировать для проверки</b>	<b>Основные показатели оценки результата</b>	<b>№ заданий для проверки</b>
ПК.1.1. ОК 2. ОК. 4.	-Умение описывать внешнюю информационную среду, к которой должны применяться программы разрабатываемой ПС -Умение определять функции ПС, определенных на множестве состояний этой информационной среды (такие функции будем называть внешними функциями ПС); - Владение навыками описания нежелательных	<i>МДК 01.01 МДК 01.02 1-54</i>



	(исключительных) ситуаций, которые могут возникнуть при выполнении программ ПС, и реакций на эти ситуации, которые должны обеспечить соответствующие программы	
ПК 1.2. ОК 2. ОК 3.	<ul style="list-style-type: none"> <li>- Умение правильно и обоснованно выбирать алгоритм с использованием средств автоматизированного проектирования</li> <li>-. Владение навыками корректного построения алгоритма в соответствии с заданной задачей</li> <li>-Умение разрабатывать код программного модуля на современных языках программирования.</li> </ul>	МДК 01.01 МДК 01.02 1-54
ПК 1.3. ОК 2. ОК 3. ОК 4.	<ul style="list-style-type: none"> <li>-Умение разрабатывать спецификации этапов жизненного цикла программы в соответствии с поставленной задачей</li> <li>-Умение использовать инструментальные средств на этапе отладки программного продукта</li> <li>- Владеть навыками отладки программы на уровне модуля</li> </ul>	МДК 01.01 МДК 01.02 1-34,40-54
ПК 1.4. ОК 2. ОК 4. ОК 5.	<ul style="list-style-type: none"> <li>- Владеть навыками тестирования программного модуля по разработанному сценарию</li> <li>- Владеть навыками</li> </ul>	МДК 01.01 МДК 01.02 1-54

	<p>тестирования программы на уровне модуля</p> <p>- Владеть навыками тестирования программ с использованием инструментальных средств</p>	
<p>ПК 1.5.</p> <p>ОК5.</p> <p>ОК 9.</p>	<p>- Умение осуществлять оптимизацию программного кода модуля по определенному сценарию</p> <p>- Умение использовать методы для получения кода с заданной функциональностью и степенью качества</p> <p>- Уметь использовать инструментальные средства для оптимизации программного кода модуля</p>	<p><i>МДК</i> <i>01.01</i> <i>МДК</i> <i>01.02 1-54</i></p>
<p>ПК 1.6.</p> <p>ОК 9.</p>	<p>- Умение правильно определять и использовать методы и средства разработки технической документации</p> <p>- Умение применять инструментальные средства для автоматизации оформления документации</p> <p>- Умение разрабатывать отдельные компоненты документации на программные средства с использованием графических языков спецификаций</p>	<p><i>МДК</i> <i>01.01</i> <i>МДК</i> <i>01.02 1-34,40-54</i></p>
<p>ПК.1.7.</p> <p>ОК 2.</p> <p>ОК 3.</p>	<p>- Умение правильно применять методы контроля качества объектно-</p>	<p><i>МДК</i> <i>01.01</i> <i>МДК</i></p>

ОК 4.	<p>ориентированного программирования;</p> <ul style="list-style-type: none"> <li>- Умение осуществлять рефакторинг и оптимизацию программного кода;</li> <li>- Умение работать с системой контроля версий.</li> </ul>	01.02 35- 39
-------	---	-----------------

## **2 ОЦЕНКА ОСВОЕНИЯ ТЕОРЕТИЧЕСКОГО КУРСА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

Основной целью оценки МДК.01.01.Системное программирование и МДК.01.02. Прикладное программирование является оценка умений и знаний.

Обучающийся в ходе освоения профессионального модуля должен:

### **уметь:**

- осуществлять разработку кода программного модуля на современных языках программирования;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- оформлять документацию на программные средства;
- использовать инструментальные средства для автоматизации оформления документации;

### **знать:**

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- основные принципы отладки и тестирования программных продуктов;
- методы и средства разработки технической документации.

Оценка теоретического курса профессионального модуля осуществляется с использованием следующих форм и методов контроля: входной контроль, контрольная работа, практическая работа, рубежный контроль, экзамен.

Таблица 2.1

**Паспорт оценочных средств**

№	Контролируемые разделы (темы) МДК*	Контролируемые умения, знания	Контролируемые компетенции	Наименование оценочного средства	
				Текущий контроль	Промежуточная аттестация
МДК.01.01. Системное программирование					Дифференциро- ванный зачет
1	Тема 01.01.02 Машинное представление данных	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Тестовый контроль	
2	Тема 01.01.04 Основные машинные команды Ассемблер	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Тестовый контроль	
3	Тема 01.01.05 Написание программ на языке Ассемблер	31-4,У1-5	ПК 1.1-1.7	Контрольная работа	
4	Тема 01.01.06 Управление системными ресурсами компьютера	31-4,У1-5	ПК 1.1-1.7	Практико- ориентированное задание	
МДК.01.02. Прикладное программирование					дифференцир- ованный зачет
1	Тема 01.02.02 Разработка консольного приложения	31-4,У1-5	ПК 1.1-1.7	Контрольная работа	
2	Тема 01.02.04. Введение в объектно-ориентированное программирование	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Тестовый контроль Практико- ориентированное задание	
3	Тема 01.02.07 Расширения управляемого C++ (Managed	31-4,У1-5	ПК 1.1-1.7	Контрольная работа	

	C++)			Тестовый контроль	
4	Тема 01.02.08 Работа со строками в Windows	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Практико-ориентированное задание	
5	Тема 01.02.09 Принципы разработки Windows приложений в .NET (Windows Forms Application)	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Практико-ориентированное задание	
6	Тема 01.02.10 Окна инструментов среды разработки Visual Studio	31-4,У1-5	ПК 1.1-1.7	Контрольная работа	
7	Тема 01.02.11 Перемещение и изменение размеров окон инструментов	31-4,У1-5	ПК 1.1-1.7	Контрольная работа	
8	Тема 01.02.12 Контроль версий GIT	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Практико-ориентированное задание	
9	Тема 01.02.14 Регистры накопления. Макеты. Перечисления. Периодические регистры сведений	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Тестовый контроль	
	Тема 01.02.15 Перечисления. Обратные регистры накопления. План видов характеристик.	31-4,У1-5	ПК 1.1-1.7	Контрольная работа Тестовый контроль	

## 2.1 Задания для оценки освоения МДК 01.01 и МДК 01.02

### 2.1.1 ВХОДНОЙ КОНТРОЛЬ

#### Спецификация

Входной контроль проводится с целью определения готовности обучающихся к освоению учебной дисциплины, базируется на дисциплинах, предшествующих изучению данной учебной дисциплины:

- «Элементы высшей математики»;
- «Элементы математической логики»;
- «Теория вероятностей и математическая статистика»;
- «Операционные системы»;
- «Архитектура компьютерных систем»;
- «Технические средства информатизации»;
- «Информационные технологии»;
- «Основы программирования»;
- «Теория алгоритмов».

По результатам входного контроля планируется осуществление в дальнейшем дифференцированного и индивидуального подхода к обучающимся. При низком уровне знаний проводятся корректирующие курсы, дополнительные занятия, консультации.

#### Примеры заданий входного контроля МДК.01.01. Системное программирование

1. Перечислите составляющие системного блока.
2. Перечислите виды памяти компьютера
3. Переведите числа в десятичную систему, а затем проверьте результаты, выполнив обратные переводы:
  - а) 10110112;      е) 5178;      л) 1F16;
  - б) 101101112;    ж) 10108;      м) ABC16;
  - в) 0111000012;    з) 12348;      н) 101016;
  - г) 0,10001102;    и) 0,348;      о) 0,4A16;
  - д) 110100,112;    к) 123,418;    п) 1DE,C816.

#### Критерии оценки

За каждый правильный ответ – 1 балл.

За неправильный ответ – 0 баллов.

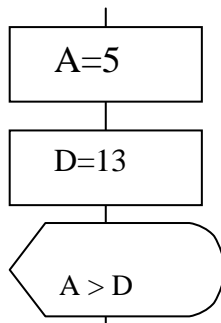
Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог

$90 \div 100$	5	отлично
$80 \div 89$	4	хорошо
$70 \div 79$	3	удовлетворительно
менее 70	2	не удовлетворительно

## МДК.01.02. Прикладное программирование

### Вариант- В1

- Определить, какого типа будет выражение -  $15/3$ 
  - Логического
  - Символьного
  - Вещественного
  - Целого
- Любая блок - схема начинается с блока:
  - Начало
  - Вывода
  - Вычисления
  - Ввода
- Какое значение переменной S получится при вычислении  $S = 20 \% 6$ 
  - 3.3
  - 3
  - 2
  - 3.33
- Определить, какой оператор является оператором цикла:
  - While do
  - If then
  - Switch of
  - Main return
- Определить, что будет напечатано на экране в результате выполнения программы:
  - 0
  - 13
  - 5
  - 1





## Вариант- В2

1. Определить, какого типа будет выражение - 1+0.0
  - A. Логического
  - B. Символьного
  - C. Вещественного
  - D. Целого
2. Какое значение переменной S получится при вычислении  $S = 20 / 6$  (целые числа)
  - A. 3.3
  - B. 3
  - C. 2
  - D. 3.33
3. Любая блок - схема начинается с блока:
  - A. Ввода
  - B. Начало
  - C. Условия
  - D. Вычисления
4. Определить, какое значение переменной Y получится при вычислении логического выражения –  $Y = \text{odd}(\text{trunc}(10 * 0.182))$ 
  - A. 0.182
  - B. 0
  - C. 1.82
  - D. 1
5. Определить, как будет выглядеть запись арифметического выражения при использовании стандартных функций –  $Y = X^4$ 
  - A.  $Y = \text{sqrt}(\text{sqr}(x))$
  - B.  $Y = \text{sqr}(\ln(x))$
  - C.  $Y = \exp(\ln(x))$
  - D.  $Y = \text{pow}(x, 4)$
6. Определить, какой оператор является оператором цикла:
  - A. If then
  - B. Case of
  - C. Do while
  - D. Then else

## Критерии оценки

За каждый правильный ответ – 1 балл.

За неправильный ответ – 0 баллов.

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

### 2.1.2 Текущий контроль

Текущий контроль успеваемости осуществляется в ходе повседневной учебной работы по междисциплинарному курсу. Данный вид контроля должен стимулировать стремление к систематической самостоятельной работе по изучению программы курса, овладению профессиональными и общими компетенциями, позволяет отслеживать положительные/отрицательные результаты и планировать предупреждающие/корректирующие мероприятия.

### Формы текущего контроля

### 1 ТЕСТОВЫЙ КОНТРОЛЬ

#### МДК.01.01.Системное программирование

#### Тема 01.01.02 Машинное представление данных

№ п/п задания	Содержание тестового задания	Варианты ответов
1.	Язык Ассемблера – это	1) символическое представление машинного языка 2) числовое представление машинного языка 3) знаковое представление машинного языка 4) смешанное представление

		машинного языка
2.	Языком Ассемблера является:	1) Лисп (List Information Symbol Processing) 2) Фортран (FORMulaeTRANslation) 3) Бейсик (Beginner's All-purpose Symbolic Instruction Code) 4) ни один из языков не является языком Ассемблера
3.	Сколько существовало языков низкого уровня, последовательно сменивших друг друга	1) 1 2) 2 3) 3 4) 4
4.	Вычислительная система – это	1) совокупность не только аппаратных, но и программных средств 2) совокупность только аппаратных 3) совокупность только программных средств 4) ни один из предложенных вариантов неверен
5.	Ассемблер — это	1) компилятор с языка ассемблера в команды машинного языка 2) интерпретатор с языка ассемблера в команды машинного языка 3) транслятора с языка ассемблера в команды машинного языка 4) деинтерпритатор с языка ассемблера в команды машинного языка
6.	Что не является недостатком языков ассемблера	1) большие объемы кода

		2) трудоемкость чтения и поиска ошибок 3) непереносимость на другие платформы 4) небольшой набор символов по сравнению с языками высокого уровня
7.	Комментарий описывается следующей диаграммой	1) : 2) ; 3) . 4) ,
8.	Директива — это	1) команда, выполняемая транслятором во время обработки программы 2) документация, которой придерживается программист 3) комментарии, описанные диаграммой 4) указание команды (инструкции) процессора
9.	Основная единица ассемблерной программы — это	1) символ 2) строка 3) блок 4) число
10.	Описание сегмента оканчивается директивой	1) segment 2) segment ends 3) segment end 4) ends
11.	Максимальный размер сегмента ограничен и составляет	1) $2^{16}$ 2) $2^8$ 3) $2^{32}$ 4) $2^4$
12.	Директивы определения общего режима трансляции определяют глобальные особенности трансляции, которые должны учитываться при обработке текста программы	1) режимов трансляции 2) секционирования программы 3) макросредств 4) резервирования

		данных
13.	Директивы предназначенные для оформления логически законченных участков программы	1) режимов трансляции 2) секционирования программы 3) макросредств 4) резервирования данных
14.	Директивы позволяющие при написании программы оперировать заранее заготовленными фрагментами текстов	1) режимов трансляции 2) секционирования программы 3) макросредств 4) резервирования данных
15.	Директивы размещающие в случае необходимости, инициализирующие области памяти под данные программы	1) режимов трансляции 2) секционирования программы 3) макросредств 4) резервирования данных
16.	Директивы размещающие в случае необходимости, инициализирующие области памяти под данные программы	1) режимов трансляции 2) секционирования программы 3) макросредств 4) резервирования данных
17.	В программной модели процессора x86 имеется:	1) 31 регистр 2) 21 регистр 3) 32 регистра 4) 22 регистра
18.	В программной модели процессора x86 имеется регистров прикладного программиста:	1) 13 2) 14 3) 15 4) 16
19.	В программной модели процессора x86 имеется регистров системного программиста:	1) 13 2) 14 3) 15 4) 16
20.	Сколько используется регистров общего назначения	1) 4 2) 8 3) 16 4) 32
21.	Сколько бит содержат регистры общего	1) 4

	назначения	2) 8 3) 16 4) 32
22.	Какие регистры используются как указательные	1) EBP, ESI, EDI и ESP 2) EAX, EBX, ECX, EDX, EBP, ESI, EDI и ESP. 3) AX, BX, CX, DX, BP, SI, DI и SP. 4) AH и AL
23.	Что означает системный флажок VM	1) виртуальный режим 2) защищенный режим 3) возобновление 4) низший приоритет
24.	Что означает системный флажок R	1) виртуальный режим 2) защищенный режим 3) возобновление 4) низший приоритет
25.	Что означает системный флажок NT	1) виртуальный режим 2) вложенная задача 3) текущая задача 4) низший приоритет
26.	Что означает системный флажок IOPL	1) виртуальный режим 2) прерывание 3) трассировка 4) уровень привилегии ввода/вывода
27.	Что означает системный флажок T	1) виртуальный режим 2) прерывание 3) трассировка 4) уровень привилегии ввода/вывода
28.	Флажки действуют только на циклические команды	1) флажки управления 2) флажки состояния 3) системные флажки 4) прикладной флажки
29.	Содержимое одного из регистров общего назначения — это	1) база 2) смещение 3) индекс 4) масштаб
30.	Адрес ячейки в сегменте памяти — это	1) база 2) смещение

		3) индекс 4) масштаб
31.	База + (Индекс * Масштаб) + Смещение=	1) EA 2) AE 3) CE 4) EC
32.	Индекс заключается в	1) () 2) {} 3) [] 4) ""
33.	Сегменты памяти, определяемые только одними сегментными регистрами, не имеют атрибута	1) начальный адрес 2) конечный адрес 3) максимальный размер 4) адрес, находящийся на границе параграфа
34.	Как размещаются сегменты в памяти	1) произвольно 2) в строгом порядке 3) систематически 4) программно

#### Тема 01.01.04 Основные машинные команды Ассемблер

№ п/п задания	Содержание тестового задания	Варианты ответов
1.	Какая из команд пересылает один элемент данных из одного места в другое	1) MOV 2) XCHG 3) PUSH 4) POP
2.	Какая из команд обменивает содержимое двух регистров или содержимое регистра и ячейки памяти	1) MOV 2) XCHG 3) PUSH 4) POP
3.	Какая из команд копирует операнд-источник в вершину стека	1) MOV 2) XCHG 3) PUSH 4) POP
4.	Какая из команд берет верхний элемент из стека и пересылает его в операнд-получатель	1) MOV 2) POP 3) PUSHA и PUSHAD 4) POPAи POPAD

5.	Какая из арифметических команд суммирует два операнда, помещая результат в первый операнд	1) ADD 2) SUB 3) INC 4) DEC
6.	Какая из арифметических команд умножает беззнаковые целые числа	1) MUL 2) SUB 3) DIV 4) DEC
7.	Какая из арифметических команд умножает знаковые целые числа	1) MUL 2) IMUL 3) DIV 4) DEC
8.	Какая из арифметических команд изменяет знак операнда, находящегося в регистре или памяти	1) MUL 2) IMUL 3) NEG 4) ADC
9.	Что не является командой преобразования данных	1) MOVSX 2) IMUL 3) CBW 4) MOVZX
10.	Какая команда заполняет регистр EDX знаковым битом регистра EAX	1) CBW 2) IMUL 3) CBW 4) CDQ
11.	Какая из команд не относится к логическим командам	1) MOVSX 2) OR 3) TEST 4) AND
12.	Какая из логических команд выполняет инвертирует биты своего операнда	1) MOVSX 2) XOR 3) NOT 4) AND
13.	Какая из логических команд выполняет логическую функцию ИСКЛЮЧАЮЩЕГО ИЛИ для двух операндов	1) MOVSX 2) XOR 3) NOT 4) AND
14.	Какая из логических команд выполняет логическую функцию ИЛИ для двух операндов	1) MOVSX 2) XOR 3) OR 4) NOR
15.	К командам сдвига и циклического сдвига не относятся команды	1) SHR, SHL 2) SAR, SAL



		3) ROR, ROL 4) RAR, RAL
16.	Вывод целых чисел производится в диапазоне	1) от – 32768 до 32767 2) от – 32767 до 32766 3) от 0 до 255 4) от – 32769 до 32768
17.	Вывод беззнаковых чисел производится в диапазоне	1) от 0 до 9 2) от 0 до 36 3) от 0 до 255 4) от 0 до 65535
18.	Совокупность файлов, системных таблиц, методов доступа к файлам, способов организации файлов, процедур доступа к файлам — это	1) Файловая таблица 2) Файловая система 3) Файловая директива 4) Файловая оболочка
19.	Вывод беззнаковых чисел производится в диапазоне	1) от 0 до 9 2) от 0 до 36 3) от 0 до 255 4) от 0 до 65535
20.	В любой файловой системе запись о файле не содержит:	1) имя и расширение имени файла; 2) атрибуты файла 3) методы доступа к файлам 4) размер (длину) файла
21.	Минимальный объем внешней памяти, выделяемый под файл	1) блок 2) участок 3) сегмент 4) сектор
22.	Стандартный ввод только для чтения обозначается цифрой	1) 0 2) 1 3) 2 4) 3
23.	Стандартный вывод только для записи обозначается цифрой	1) 4 2) 3 3) 2 4) 1
24.	Любая операционная система поддерживает следующие запросы на доступ к файлам	1) Создание 2) Удаление 3) Запись 4) Ничего из вышеперечисленного

25.	Сколько двухсловных векторов располагается в оперативной памяти	1) 8 2) 32 3) 256 4) 1024
26.	Самотестирование основных узлов компьютера после включения питания называется	1) UMB 2) POST 3) GET 4) BIOS
27.	Структура данных длиной в 1 параграф, используемая системой для отслеживания состояния памяти	1) UMB 2) MCB 3) MBC 4) UBM
28.	Реакция вычислительной системы на некоторое асинхронное событие называется	1) прерывание 2) ошибка 3) звуковой сигнал 4) выключение компьютера
29.	Узел компьютера, логически располагающийся между процессором и внешними устройствами	1) контроллер прерываний 2) контроллер ошибок 3) контроллер звуковых сигналов 4) контроллер выключения компьютера
30.	Команда близкого или внутрисегментного вызова	1) call far 2) call near 3) ret near 4) retfar
31.	В состав обработчика прерываний не входит	1) вход в драйвер 2) инициализация выполнения запроса на внешнем устройстве 3) инициализация выполнения запроса на внутреннем устройстве 4) выход из драйвера
32.	Описание работы, которую должна выполнить система — это	1) задание 2) процесс 3) действие 4) условие
33.	Последовательность команд,	1) алгоритм

	выполняемых процессором над определенным набором данных, принадлежащих одному процессу	2) оператор 3) процесс 4) поток
34.	Простые объекты ядра, у которых нет специальных условий, при которых они переключаются в сигнализирующее состояние	1) мьютексы 2) события 3) семафоры 4) ждущие таймеры
35.	Совокупность конструкций языка, заставляющих транслятор генерировать и/или модифицировать исходный текст программы	1) макросредства 2) макрокоманды 3) микросредства 4) микрокоманды
36.	При использовании какой директивы макрогенератор транслятора будет повторять последовательность строк до тех пор, пока значение выражение не станет равно нулю	1) REPT 2) UNT 3) WHILE 4) FOR

## МДК.01.02. Прикладное программирование

### Тема 01.02.04. Введение в объектно-ориентированное программирование

1. Отметьте все утверждения, которые считаете верными:

- 1) нельзя с помощью шаблона создать функцию с таким же именем, как у явно определенной функции;
- 2) цель введения шаблонов – создание функций, которые могут обрабатывать однотипные данные;
- \*3) при использовании шаблонов функций возможна перегрузка как с помощью шаблонов, так и функций;
- \*4) в качестве описания шаблона функции используется прототип шаблона:

*template список \_ параметров \_ шаблона*

2. Можно ли задать шаблон со значением параметра по умолчанию?

*Варианты ответа:*

\*1) да; 2) нет.

3. Каков правильный заголовок шаблона?

*Варианты ответа:*

\*1) `template<class t1,class t2>;` 2) `template <class t1,t2>;` 3) `template <class t,class t>;`

4. Сколько параметров может быть у шаблона при определении шаблона функции?

*Варианты ответа:*

1) 1; 2) столько, сколько аргументов у функции; \*3) столько, сколько типов используется для параметризации.

5. Отметьте правильный вариант описания шаблона семейства функций:

1) `template(class T)`

`void func(T* p1, T* p2){...}`

2) `template <class T>;`

`void func(T* p1, T* p2){...}`

\*3) `template<class T>`

`void func(T* p1, T* p2){...}`

4) `template<class T>`

`void func(T* p1, T* p2){...}`

6. Можно ли использовать класс-шаблон в качестве базового класса?

*Варианты ответа:*

\*1) да; 2) нет.

### **Тема 01.02.07 Расширения управляемого C++ (Managed C++)**

Какая из записей является правильной записью абстрактного класса?

*Варианты ответа:*

1) `abstract class A{virtual f()=0;};`

\*2) `class A{virtual f()=0;};`

3) `class A{virtual f();};`

2. Абстрактный класс – это класс, в котором:

1) есть хотя бы один виртуальный метод;

2) есть виртуальный конструктор;

3) есть виртуальный деструктор;

\*4) есть чисто виртуальный метод.

3. Основная проблема множественного наследования состоит в:

1) замедлении выполнения программ;

2) необходимости явно указывать, к какому базовому классу принадлежит метод;

\*3) возможности существования двух экземпляров объекта базового класса;

4) неэкономном расходовании памяти.

4. Если записано

`class A{public:virtual void f(){cout<<1;}};`

`class B:public A {public:virtual void f(){cout<<2;}};`

то что будет напечатано, если

`B b;A &a=b; a.f(); ?`

*Варианты ответа:*

\*1) 2; 2) 21; 3) 12; 4) 1; 5) ошибка.

5. Если записано

`class A {public:void f(){cout<<1;}};`

```
class B:public A{public: void f(){cout<<2;}};
```

то что будет напечатано, если

B b; b.f(); ?

*Варианты ответа:*

\*1) 2; 2) 21; 3) 12; 4) 1.

6. Если записано

```
class A{public:virtual void f(){cout<<1;}};
```

```
class B:public A{public:virtual void f(){cout<<2;}};
```

то что будет напечатано, если

B b; b.f(); ?

*Варианты ответа:*

\*1) 2; 2) 21; 3) 12; 4)1; 5)ошибка.

## **Тема 01.02.14 Регистры накопления. Макеты. Перечисления.**

### **Периодические регистры сведений**

#### **1. Из чего состоит конфигурация**

1. Объекты конфигурации
2. Объекты встроенного языка
3. Объекты информационной базы
4. Верны ответы 1 и 3
5. Верны все варианты

#### **2 Как можно разделить список пользователей на группы?**

- 1.Нет такой возможности.
- 2.В списке пользователей можно создать группу пользователей.
3. Разделение списка осуществляется с помощью общих реквизитов.

#### **3 Как разделить механизм нумерации документов по филиалам для многофирменного учета?**

- 1.Только программно.
- 2.С помощью механизма разделения данных.
3. Разделение нумератора для многофирменного учета не поддерживается.

#### **4 Какие варианты подчинения существуют в системе**

##### **1С:Предприятие 8**

1. группам
2. элементам
3. группам и элементам
4. Верны ответы 1 и 3
5. Верны ответы 2 и 3
6. Верны все указанные ответы

##### **5 Какие виды иерархии существуют в системе 1С:Предприятие 8**

1. Иерархия групп
2. Иерархия элементов
3. Иерархия групп и элементов

4. Верны ответы 1 и 3
5. Верны ответы 2 и 3
6. Верны все указанные ответы

**16 Какие компоненты типовой поставки в системе 1С:Предприятие 8 защищены аппаратным ключом**

1. Клиентское приложение
2. Типовое типовое решение
3. Сервер 1С:Предприятие
4. Правильны ответы 1 и 3
5. Правильны ответы 1 и 2
6. Верны все варианты

**7 Какие объекты используются при описании алгоритма?**

1. Объекты конфигурации
2. Объекты встроенного языка
3. Объекты информационной базы
4. Верны ответы 1 и 3
5. Верны все варианты

**8 Какими данными и с какой целью обеспечивает пользователя «Граница последовательности документов»?**

1. Моментом времени, начиная с которого перепроведение документов последовательности в хронологическом порядке, восстановит правильность (актуальность) учета, контролируемого последовательностью
2. Датой, начиная с которой перепроведение всех документов в хронологическом порядке, восстановит правильность (актуальность) учета, контролируемого последовательностью
3. Ссылкой на документ последовательности, начиная с которого перепроведение документов последовательности в хронологическом порядке, восстановит правильность (актуальность) учета, контролируемого последовательностью

**9 Каким может быть тип данных ресурса у регистра сведений?**

1. Один из примитивных типов данных
2. Только ссылочные типы
3. Хранилище значений
4. Составной тип данных
5. Верны все перечисленные ответы
6. Верны ответы 1 и 2

**10 Какого вида клиентского приложения не существует в системе 1С:Предприятие 8?**

1. Отладочный клиент
2. Толстый клиент
3. Тонкий клиент

4. Веб-клиент
5. Не существует 2 и 3 вариантов

**Тема 01.02.15 Перечисления. Оборотные регистры накопления. План видов характеристик.**

**1 Какое максимальное количество видов документов позволяет создать система 1С:Предприятие 8.0 в процессе конфигурирования?**

1. Неограниченное
2. Ограничивается свойством конфигурации "Количество видов документов"
3. 256, т.к. система не позволяет обрабатывать более 256 таблиц объектов одного типа
4. 50, т.к. максимальная длина номера документа равна 50 символов

**2 Какое максимальное количество реквизитов документа позволяет определить система 1С:Предприятие 8.0?**

1. Количество реквизитов документа не ограничено
2. Максимальное количество реквизитов ограничивается в свойстве "Количество реквизитов"
3. Количество реквизитов документа не ограничено, но при этом реквизитов ссылочного типа не может быть больше 10
4. Количество реквизитов зависит от варианта работы с информационной базой. В файловом - до 50 реквизитов, в клиент-серверном - неограниченно

**3 Какое максимальное количество табличных частей документа позволяет определить система 1С:Предприятие 8.0?**

1. Количество табличных частей документа не ограничено
2. Максимальное количество табличных частей ограничивается в свойстве "Количество табличных частей"
3. У документа может быть только одна табличная часть
4. Количество табличных частей зависит от варианта работы с информационной базой. В файловом - до 50, в клиент-серверном - неограниченно

**4 Какую конфигурацию можно изменять интерактивно в конфигураторе?**

1. Основная конфигурация
2. Конфигурация базы данных
3. Конфигурация поставщика
4. Внешние конфигурации

**5 Картинки для прикладного решения могут браться из...**

1. библиотеки картинок технологической платформы
2. библиотеки картинок конфигурации
3. из файловой системы

4. верны варианты 1 и 2

5. верны все варианты

**6. Можно ли в системе 1С: Предприятия определить свой тип данных, например "ЦенаНоменклатуры"?**

1. Да, для этого есть определяемые типы.

2. Да, для этого есть общие реквизиты

3. Такой возможности в платформе нет.

**7 На компьютерах с какой операционной системой могут работать пользователи системы 1С:Предприятие 8.3?**

1. На Windows

2. На Windows и Linux

3. На Windows - с использованием толстого, тонкого, веб- клиентов, на Linux - только Веб-клиент

4. На windows только Толстый клиент, на Linux - Тонкий и Веб-клиент

**8 Назовите основное назначение объекта «Последовательность документов»?**

1. Автоматизация контроля над хронологическим порядком проведения документов тех видов, которые указаны в последовательности

2. Чтобы запрещать пользователю непоследовательное проведение документов

3. Чтобы запрещать пользователю проведение документов "задним числом"

4. Автоматизация разрешения коллизий при одновременной записи в информационную базу нескольких документов, принадлежащих последовательности

5. Этот объект позволяет вести список тех документов, которые были проведены "задним числом"

**9 Назовите основное назначение объектов типа «Документ»?**

1. Предназначены для хронологического отражения в системе событий предметной области, например, хозяйственных операций предприятия, контактов с покупателями

2. Предназначены для отражение в системе условно-постоянной информации, например, карточек контрагентов

3. Предназначены только для отражения хозяйственных операций в регистрах учета, например, в регистрах бухгалтерии

4. Предназначены только для печати на бумажных носителях унифицированных форм, например, счетов-фактур, расходных накладных

5. Предназначены только для обработки больших объемов данных в целях получения информации нового качества, например, формирование книги покупок, книги продаж по учету НДС

**10 Можно ли в 1С:Предприятии создавать общий реквизит?**



1. Да, такая возможность присутствует
2. Механизмы для создания общих реквизитов в платформе отсутствуют.

## 2.1.1 КОНТРОЛЬНЫЕ РАБОТЫ

### МДК.01.01. Системное программирование

#### Тема 01.01.02 Машинное представление данных

##### Вариант 1

1. Как представлено число  $141_{10}$  в двоичной, восьмеричной и шестнадцатеричной системах счисления?
2. Перевести число  $111000111$  из двоичной системы счисления в десятичную, в восьмеричную и шестнадцатеричную.
3. Чему равна сумма чисел  $57_8$  и  $46_{16}$ ? Ответ представить в двоичной системе счисления.
4. Перевести восьмеричное число  $732$  в десятичную СС.
5. Перевести шестнадцатеричное число  $AB2$  в десятичную СС.
6. Перевести восьмеричное число  $1126$  в шестнадцатеричную СС.
7. Перевести шестнадцатеричное число  $F205$  в восьмеричную СС.
8. Перевести вещественное десятичное число  $21,3$  в двоичную СС с точностью до трех знаков после запятой.
9. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа  $22$  оканчивается на  $4$ .
10. Третья цифра шестнадцатеричного четырехзначного числа равна  $9$ . Первую цифру переставили в конец числа. полученное число оказалось на  $2B7A_{16}$  меньше исходного. Найти исходное число, записанное в системе счисления по основанию  $16$ .
11. Трехзначное число, записанное в одиннадцатеричной системе счисления, уменьшается вдвое от перестановки первой цифры в конец числа. Определить максимальное из таких чисел, записанное в системе счисления по основанию  $11$ .

##### Вариант 2

1. Как представлено число  $144_{10}$  в двоичной, восьмеричной и шестнадцатеричной системах счисления?
2. Перевести число  $110011001$  из двоичной системы счисления в десятичную, в восьмеричную и шестнадцатеричную.
3. Дано  $N = 227_8$ ,  $M = 99_{16}$ . Какое из чисел  $K$ , записанных в двоичной системе, отвечает условию  $N < K < M$ ?
4. Перевести восьмеричное число  $532$  в десятичную СС.
5. Перевести шестнадцатеричное число  $65$  в десятичную СС.
6. Перевести восьмеричное число  $2205$  в шестнадцатеричную СС.
7. Перевести шестнадцатеричное число  $ADD6$  в восьмеричную СС.

8. Перевести вещественное десятичное число 28,5 в двоичную СС с точностью до трех знаков после запятой.
9. В системе счисления с некоторым основанием число 12 записывается в виде 110. Укажите это основание.
10. Шестнадцатеричное четырехзначное число начинается цифрой 8. Первую цифру переставили в конец числа. Полученное число оказалось на  $685B_{16}$  меньше исходного. Найти исходное число, записанное в системе счисления по основанию 16.
11. Трехзначное число, записанное в тринадцатеричной системе счисления, увеличивается вчетверо от перестановки первой цифры в конец числа. Определить максимальное из таких чисел, записанное в системе счисления по основанию 13.

### Вариант № 3

1. Как представлено число  $143_{10}$  в двоичной, восьмеричной и шестнадцатеричной системах счисления?
2. Перевести число 101010101 из двоичной системы счисления в десятичную, в восьмеричную и шестнадцатеричную.
3. Дано  $A=A7_{16}$ ,  $B=251_8$ . Какое из чисел  $C$ , записанных в двоичной системе, отвечает условию  $A < C < B$ ?
4. Перевести восьмеричное число 645 в десятичную СС.
5. Перевести шестнадцатеричное число 3BC в десятичную СС.
6. Перевести восьмеричное число 3145 в шестнадцатеричную СС.
7. Перевести шестнадцатеричное число 239F в восьмеричную СС.
8. Перевести вещественное десятичное число 42,3 в двоичную СС с точностью до трех знаков после запятой.
9. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 39 оканчивается на 3.
10. Трехзначное число, записанное в шестнадцатеричной системе счисления, увеличивается втрое от перестановки первой цифры в конец числа. Найти максимальное из таких чисел, записанное в системе счисления по основанию 16.
11. Трехзначное число, записанное в одиннадцатеричной системе счисления, уменьшается вдвое от перестановки первой цифры в конец числа. Определить максимальное из таких чисел, записанное в системе счисления по основанию 11.

### Вариант № 4

1. Как представлено число  $132_{10}$  в двоичной, восьмеричной и шестнадцатеричной системах счисления?

2. Перевести число 101110011 из двоичной системы счисления в десятичную, в восьмеричную и шестнадцатеричную.
3. Чему равна сумма чисел  $34_8$  и  $CC_{16}$ ? Ответ представить в двоичной системе счисления.
4. Перевести восьмеричное число 640 в десятичную СС.
5. Перевести шестнадцатеричное число 5F в десятичную СС.
6. Перевести восьмеричное число 5647 в шестнадцатеричную СС.
7. Перевести шестнадцатеричное число 254A в восьмеричную СС.
8. Перевести вещественное десятичное число 35,8 в двоичную СС с точностью до трех знаков после запятой.
9. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 29 оканчивается на 5.
10. Шестнадцатеричное четырехзначное число заканчивается цифрой 9. Первую цифру переставили в конец числа. Полученное число оказалось на  $2EA4_{16}$  меньше исходного. Найти исходное число, записанное в системе счисления по основанию 16.
11. Трехзначное число, записанное в тринадцатеричной системе счисления, увеличивается вчетверо от перестановки первой цифры в конец числа. Определить максимальное из таких чисел, записанное в системе счисления по основанию 13.

#### **Тема 01.01.04 Основные машинные команды Ассемблер**

1. Составить программу для подсчета двухбайтовой контрольной суммы области памяти внутри одного сегмента DS от адреса A1 до A2.

*Рекомендации* Адреса A1, A2 - двухбайтовые. Начальный адрес A1 поместите в регистр SI, конечный A2 – в регистр DI. В качестве примера возьмите A1=1000h, A2=10FFh. Для проверки работы программы обнулите эту область и запишите туда несколько байт, чтобы их можно было для проверки было легко сложить.

2. Составить программу, которая копирует свой код в другую область памяти в том же кодовом сегменте, например, сразу же за текстом программы. Можно ли создать программу, которая копирует свой код N раз?

#### **Тема 01.01.05 Написание программ на языке Ассемблер**

1. Изучить ассемблерный код программы сложения двух чисел, реализованной на рисунке. На основе представленного кода написать программу сложения трех чисел.

```

pr1_1.asm — Блокнот
Файл  Правка  Формат  Вид  Справка
;Программа сложения двух чисел:
s_s      segment stack "stack"      ;адрес начала сегмента стека
        dw 12 dup(?)                ;зарезервировано в памяти 12 ячеек с любым
        ;значением
s_s      ends                        ;адрес  конца сегмента стека

d_s      segment                    ;адрес начала сегмента данных
aa       dw 45h,23h                 ;данные т.е. числа 45h и 23h записаны по адресу aa
        ;и aa+2 соответственно т.к. они определены
        ;как слова
sum      dw 0                       ;данное т.е. число 0 записано по адресу sum
        ;(это метка)
d_s      ends                        ;адрес  конца сегмента данных

c_s      segment                    ;начало сегмента кода
        assume ss:s_s,ds:d_s,cs:c_s ;псевдооператор, определяющий каким
        ;сегментным регистром соответствуют
        ;назначенные метками адреса  начала
        ;сегментов
beg:     mov ax,d_s;                 ;начало программы - имя и после него двоеточие
        ;2 команды для пересылки в регистр сегмента данных (ds) адреса
        ;начала сегмента данных т.е. метки начала сегмента данных

        mov ds,ax                   ;пересылку в ds можно сделать только через
        ;промежуточную пересылку в рабочий регистр,
        ;например ax т.к. из памяти в сегментный регистр
        ;записывать нельзя

        mov ax,aa                   ;пересылка в регистр ax содержимого, находящегося
        ;по адресу aa т.е. числа 45h

        add ax,aa+2                  ;сложить число, которое находится в регистре ax с
        ;содержимым, находящимся по адресу aa+2
        ;т.е. числа 23h и результат записывается в регистр ax

        mov sum,ax                  ;переслать содержимое ax т.е. результат в ячейку
        ;памяти по  адресу sum

        mov ah,4ch                  ;для правильного завершения программы необходи-
        ;мо переслать в регистр ah - байт равный 4ch

        int 21h                    ;и вызвать прерывание равное 21h

c_s      ends                        ;конец сегмента кода

        end beg                     ;конец программы. Метки начала и конца програм
        ;мы должны совпадать. По этой метке транслятор
        ;установит регистры CS и IP.

```

2. Транслировать ассемблерный код. В случае нахождения ошибок – устранить их и повторить трансляцию.
3. Скомпоновать программу редактором связей.
4. Запустить программу в отладчике, в случае обнаружения ошибок в работе программы исправить их (при необходимости повторить п.п.3, 4)
5. Показать работу программы преподавателю

**МДК.01.02. Прикладное программирование**

## Тема 01.02.02 Разработка консольного приложения

Создать новый проект приложения согласно варианту, написать программу для вычисления значений выражений для вводимых с клавиатуры исходных данных. Вывести на экран значения исходных данных и результатов вычислений, сопровождая вывод именами переменных и комментариями.

ТАБЛИЦА 4.1. Параметры задания

1	$A = \frac{2 \cos(X - \pi/6)}{1/2 + \sin^2 Y}$	2	$\gamma = \left  x^{y/x} - \sqrt[3]{y/x} \right $
3	$\varphi = (y - x) \frac{y - z/(y - x)}{1 + (y - x^2)}$	4	$\varphi = X(\sin X^3 + \cos^2 Y)$
5	$s = b \sin(at^2 \cos 2t) - 1$	6	$s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$
7	$Q = \frac{bx^2 - a}{e^{ax} - 1}$	8	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a }$
9	$S = \sqrt{xb/a} + \sqrt{a^2 + b^2} (x + b)^3$	10	$\omega = \sqrt{x^2 + b} - b^2 \sin^3(x + a)/x$
11	$y = \sin^3(x^2 + a)^2 - \sqrt{x/b}$	12	$s = x^3 \operatorname{tg}^2(x + b)^2 + a/\sqrt{x + b}$
13	$f = \sqrt[3]{\operatorname{mtgt} +  \operatorname{c} \sin t }$	14	$y = \cos^2 x^3 - x/\sqrt{a^2 + b^2}$
15	$z = m \cos(bt \sin t) + c$	16	$R = x^2(x + 1)/b - \sin^2(x + a)$
17	$y = b \operatorname{tg}^2 x - \frac{a}{\sin^2(x/a)}$	18	$S = \sqrt{xb/a} + \cos^2(x + b)^3$
19	$d = ae^{-\sqrt{a}} \cos(bx/a)$	20	$R = \sqrt{x^2 + b} - b^2 \sin^3(x + a)/x$

## Тема 01.02.04. Введение в объектно-ориентированное программирование

В рамках консольного приложения создать класс `A` с полями `a` и `b` и свойством `c`. Свойство – значение выражения над полями `a` и `b` (выражение и типы полей – см. вариант в таблице 1). Поля инициализировать при объявлении класса. Конструктор оставить по умолчанию. Проследить, чтобы поля `a` и `b` напрямую в других классах были недоступны. Создать класс `Programm` с одним методом – точкой входа. В теле метода создать объект класса `A`, вывести на экран значение свойства `c`.

### Таблица 1. Варианты заданий

Вариант	Тип полей a b,	Выражения
---------	----------------	-----------

1	decimal	++(префиксный),-
2	decimal	--(префиксный),+
3	int	*=,(постфиксный)++,-
4	int	+=,--(префиксный),-
5	int	-=,(постфиксный)++,-
6	float	/=,*
7	float	*=,/
8	decimal	(постфиксный)--,+
9	decimal	-=,+
10	decimal	*=,-
11	decimal	+=,-
12	float	%=,/
13	float	%=,*
14	int	--(префиксный)
15	byte	(постфиксный)--

## Тема 01.02.07 Расширения управляемого C++ (Managed C++)

### Вариант 1

Создать класс Point, разработав следующие элементы класса:

b. Поля:

- int x, y;

c. Конструкторы, позволяющие создать экземпляр класса:

- с нулевыми координатами;
- с заданными координатами.

d. Методы, позволяющие:

- вывести координаты точки на экран;
- рассчитать расстояние от начала координат до точки;
- переместить точку на плоскости на вектор (a, b).

e. Свойства:

- получить -установить координаты точки (доступное для чтений и записи);
- позволяющие умножить координаты точки на скаляр (доступное только для записи).

### Вариант 2.

Создать класс Triangle, разработав следующие элементы класса:

e. Поля:

- int a, b, c;

f. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.

g. Методы, позволяющие:

- вывести длины сторон треугольника на экран;

- рассчитать периметр треугольника;
- рассчитать площадь треугольника.

h. Свойства:

- позволяющее получить -установить длины сторон треугольника (доступное для чтения и записи);
- позволяющее установить, существует ли треугольник с данными длинами сторон (доступное только для чтения).

Вариант 3.

Создать класс Rectangle, разработав следующие элементы класса:

a. Поля:

- int a, b;

b. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.

c. Методы, позволяющие:

- вывести длины сторон прямоугольника на экран;
- рассчитать периметр прямоугольника;
- рассчитать площадь прямоугольника.

d. Свойства:

- получить -установить длины сторон прямоугольника (доступное для чтения и записи);
- позволяющее установить, является ли данный прямоугольник квадратом (доступное только для чтения).

Вариант 4.

Создать класс Money, разработав следующие элементы класса:

Поля:

- int first; //номинал купюры
- int second; //количество купюр

b. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.

c. Методы, позволяющие:

- вывести номинал и количество купюр;
- определить, хватит ли денежных средств на покупку товара на сумму N рублей.
- определить, сколько шт товара стоимости n рублей можно купить на имеющиеся денежные средства.

d. Свойства:

- позволяющее получить - установить значение полей (доступное для чтения и записи);
- позволяющее рассчитать сумму денег (доступное только для чтения).

Вариант 5.



Создать класс для работы с одномерным массивом целых чисел. Разработать следующие элементы класса:

a. Поля:

- `int [] IntArray;`
- `int n.`

b. Конструктор, позволяющий создать массив размерности n.

c. Методы, позволяющие:

- ввести элементы массива с клавиатуры;
- вывести элементы массива на экран;
- отсортировать элементы массива в порядке возрастания.

d. Свойства:

- возвращающее размерность массива (доступное только для чтения);
- позволяющее помножить все элементы массива на скаляр (доступное только для записи).

### **Тема 01.02.08 Работа со строками в Windows**

Вариант 1

Дана строка слов, разделенных пробелами. Сформируйте новую строку, вставив перед каждым вхождением слова "and" запятую. Определите, сколько в строке симметричных слов.

Вариант 2.

Дана строка слов, разделенных пробелами. Сформируйте новую строку, вставив перед каждым вхождением слова "no" запятую. Подсчитайте количество подстрок между запятыми. Определите, сколько в этой строке слов, у которых первая буква содержится в слове более одного раза.

Вариант 3.

Дана строка слов. Сформируйте новую строку, удалив пробелы, с которых может начинаться строка, а каждую внутреннюю группу пробелов замените одним пробелом. Подсчитайте количество слов в данной строке и количество слов, у которых первая и последняя буквы совпадают.

Вариант 4.

Дана строка слов, разделенных пробелами. Определите количество слов, которые встречаются более одного раза. Сформируйте строку из неповторяющихся слов.

Вариант 5.

Дана строка слов, разделенных пробелами. Сформируйте строку из неповторяющихся слов, расположив их в алфавитном порядке.

## **Тема 01.02.09 Принципы разработки Windows приложений в .NET (Windows Forms Application)**

### **Вариант1**

Создать класс для работы с одномерным массивом целых чисел.  
Разработать следующие элементы класса:

a. Поля:

- `int [] IntArray;`
- `int n.`

b. Конструктор, позволяющий создать массив размерности n.

c. Методы, позволяющие:

- ввести элементы массива с клавиатуры;
- вывести элементы массива на экран;
- отсортировать элементы массива в порядке возрастания.

d. Свойства:

- возвращающее размерность массива (доступное только для чтения);
- позволяющее домножить все элементы массива на скаляр (доступное только для записи).

### **Вариант2.**

Создать класс для работы с двумерным массивом целых чисел.  
Разработать следующие элементы класса:

a. Поля:

- `int [,] IntArray;`
- `int n.`

b. Конструктор, позволяющий создать массив размерности  $n \times n$ .

c. Методы, позволяющие:

- ввести элементы массива с клавиатуры;
- вывести элементы массива на экран;
- вычислить сумму элементов  $i$  -того столбца.

d. Свойства:

- позволяющее вычислить количество нулевых элементов в массиве (доступное только для чтения);
- позволяющее установить значение всех элементы главной диагонали массива равное скаляру (доступное только для записи).

### **Вариант3.**

Создать класс для работы с двумерным массивом вещественных чисел.  
Разработать следующие элементы класса:

a. Поля:

- `double [][] DoubelArray;`

- `int n, m.`
- b. Конструктор, позволяющий создать массив размерности  $n \times m$ .
- c. Методы, позволяющие:
  - ввести элементы массива с клавиатуры;
  - вывести элементы массива на экран;
  - отсортировать элементы каждой строки массива в порядке убывания.
- d. Свойства:
  - возвращающее общее количество элементов в массиве (доступное только для чтения);
  - позволяющее увеличить значение всех элементов массива на скаляр (доступное только для записи).

#### Вариант4.

Создать класс для работы с двумерным массивом вещественных чисел. Разработать следующие элементы класса:

- a. Поля:
  - `double [][] DoubleArray;`
  - `int n, m.`
- b. Конструктор, позволяющий создать массив размерности  $n \times m$ .
- c. Методы, позволяющие:
  - ввести элементы массива с клавиатуры;
  - вывести элементы массива на экран;
  - отсортировать элементы каждой строки массива в порядке убывания.
- d. Свойства:
  - возвращающее общее количество элементов в массиве (доступное только для чтения);
  - позволяющее увеличить значение всех элементов массива на скаляр (доступное только для записи).

#### Вариант 5.

Создать класс для работы с двумерным массивом целых чисел. Разработать следующие элементы класса:

- a. Поля:
  - `int [,] IntArray;`
  - `int n.`
- b. Конструктор, позволяющий создать массив размерности  $n \times n$ .
- c. Методы, позволяющие:
  - ввести элементы массива с клавиатуры;
  - вывести элементы массива на экран;
  - вычислить сумму элементов  $i$ -того столбца.
- d. Свойства:

- позволяющее вычислить количество нулевых элементов в массиве (доступное только для чтения);
- позволяющее установить значение всех элементов главной диагонали массива равное скаляру (доступное только для записи).

## **Тема 01.02.10 Окна инструментов среды разработки Visual Studio**

### **Вариант 1**

Создать класс для работы со строками. Разработать следующие элементы класса:

#### **a. Поля:**

- `StringBuilder Line`;
- `int n`.

#### **b. Конструктор, позволяющий создать строку из n символов.**

#### **c. Методы, позволяющие:**

- подсчитать количество пробелов в строке;
- заменить в строке все прописные символы на строчные;
- удалить из строки все знаки препинания.

#### **d. Свойства:**

- возвращающее общее количество элементов в строке (доступное только для чтения);
- позволяющее установить значение поля, в соответствии с введенным значением строки с клавиатуры, а также получить значение данного поля (доступно для чтения и записи)

### **Вариант 2.**

Создать класс для работы с датой. Разработать следующие элементы класса:

#### **a. Поле**

- `DateTime data`.

#### **b. Конструкторы, позволяющие установить:**

- заданную дату
- дату 1.01.2009

#### **c. Методы, позволяющие:**

- вычислить дату предыдущего дня;
- вычислить дату следующего дня;
- определить сколько дней осталось до конца месяца.

#### **d. Свойства:**

- позволяющее установить или получить значение поля класса (доступно для чтения и записи)
- позволяющее определить год високосным (доступно только для чтения)

### Вариант 3.

Создать класс для работы со строками. Разработать следующие элементы класса:

#### а. Поля:

- `StringBuilder line`;
- `int n`.

#### б. Конструктор, позволяющий создать строку из `n` символов.

#### с. Методы, позволяющие:

- подсчитать количество пробелов в строке;
- заменить в строке все прописные символы на строчные;
- удалить из строки все знаки препинания.

#### д. Свойства:

- возвращающее общее количество элементов в строке (доступное только для чтения);
- позволяющее установить значение поля, в соответствии с введенным значением строки с клавиатуры, а также получить значение данного поля (доступно для чтения и записи)

### Вариант 4.

Создать класс для работы с регулярными выражениями. Разработать следующие элементы класса:

#### а. Поля:

- `Regex r`;
- `string text`;

#### б. Методы, позволяющие:

- определить, содержит ли текст фрагменты, соответствующие шаблону поля;
- вывести на экран все фрагменты текста, соответствующие шаблону поля;
- удалить из текста все фрагменты, соответствующие шаблону поля;

#### д. Свойства:

- позволяющее установить или получить строковое поле класса (доступно для чтения и записи)
- позволяющее установить или получить регулярное выражение, хранящееся в соответствующем поле класса (доступно для чтения и записи)

### Вариант 5.

Создать класс для работы с датой. Разработать следующие элементы класса:

#### а. Поле

- `DateTime data`.

b. Конструкторы, позволяющие установить:

- заданную дату
- дату 1.01.2009

c. Методы, позволяющие:

- вычислить дату предыдущего дня;
- вычислить дату следующего дня;
- определить сколько дней осталось до конца месяца.

d. Свойства:

- позволяющее установить или получить значение поле класса (доступно для чтения и записи)
- позволяющее определить год высокосным (доступно только для чтения)

## **Тема 01.02.11 Перемещение и изменение размеров окон инструментов**

Вариант 1:

Создание SDI –приложения

Приложение содержит следующие формы:

- Главная форма с меню.
- Форма для определения режимов работы программы.
- Форма «О программе» (About).
- Форма для ввода исходных данных и вывода результата.

В момент запуска приложения на экране появляются первые две формы, остальные появляются при выборе соответствующего пункта меню.

Меню имеет следующую структуру:

File Windows Help

Quit New About

Open Dialog

Главная форма содержит только меню

Вариант 2:

Создание MDI –приложения

В MDI –приложении существует одна главная форма и множество подчиненных. Чтобы окно Form1 стала главным, поменяем значение его свойства IsMdiContainer на True. Разрешается и создание окон, которые открывают из главного окна, но не являются подчиненными. Но такими окнами нельзя управлять из главного окна. В нашем приложении будут:

- Главное окно с меню.

- Окно «О программе».
- Динамически создаваемое диалоговое окно.
- Окно, которое можно создавать многократно из главного.
- Окно, которое можно создать из главного в единственном экземпляре.

### **Тема 01.02.12 Контроль версий GIT**

1. Как проиндексировать несколько файлов одной командой?
  - A. `git add TEXT1.txt, TEXT2.txt, TEXT3.txt`
  - B. `git add TEXT1.txt TEXT2.txt TEXT3.txt`
  - C. `git add TEXT1.txt ADD TEXT2.txt ADD TEXT3.txt`
2. Какой командой можно загрузить с GitHub репозиторий на свой компьютер?
  - A. `git pull`
  - B. `git fetch`
  - C. `git clone`
  - D. `git push`
3. Как создать новую ветку с именем dev?
  - A. `git branch new dev`
  - B. `git branch dev`
  - C. `git new dev`
  - D. `git create dev`
  - E. `git create subtree dev`
4. Как посмотреть список меток?
  - A. `git label`
  - B. `git labels`
  - C. `git tag`
  - D. `git show --labels`
5. Как удалить ветку night?
  - A. `git delete night`
  - B. `git branch --delete night`
  - C. `git branch -d night`
  - D. `git checkout --delete night`
6. Как посмотреть id коммита?
  - A. `git commit id`
  - B. `git id`
  - C. `git log`
  - D. `git info`
7. Какой текстовый редактор используется по умолчанию в git?
  - A. Установленный по умолчанию в системе

- B. notepad
- C. vim
- D. emacs

8. Какова минимальная длина SHA-1 хэша должна быть, чтобы можно было просмотреть информацию о коммите?

- A. 3 символа
- B. 5 символов
- C. 4 символа
- D. 6 символов

9. Как перейти из ветки master в ветку dev?

- A. git checkout dev
- B. git change master dev
- C. git branch master dev
- D. git move dev

10. Как вывести список удалённых репозиториях с именем и url?

- A. git remote -v
- B. git repository
- C. git show
- D. git remote

### **Тема 01.02.14 Регистры накопления. Макеты. Перечисления. Периодические регистры сведений**

Цель данной работы заключается в разработке нового документа, структура которого приведена в вариантах задания.

Выполнение работы состоит из следующих этапов:

1. Разработать новую подсистему «ФИО. № группы. № отделения». Все новые объекты данного задания поместить в данную подсистему.
2. Для обеспечения работы документа сформировать дополнительные объекты данных (константы, справочники, перечисления).
3. В форме документа необходимо организовать две закладки: «Шапка» (содержит реквизиты шапки) и «Таблица» (содержит табличную часть документа.).
4. Ввести в конфигурацию общий реквизит для всех документов конфигурации. Название и тип- см. варианты заданий.
5. В форме документа должно быть поле, показывающее итоговую сумму по какому-либо числовому полю табличной части документа.
6. Сформировать печатную форму документа. Вызов – по кнопке «Печать», расположенной в форме документа. Использовать конструктор запроса с обработкой результата.



7. Сформировать обработку «Работа с документами». Задачи, возложенные на данную обработку – см. варианты заданий. Поместить в подсистему «ФИО. № группы. № отделения».

8. В пользовательском режиме сформировать и записать несколько документов.

9. С помощью механизма функциональных опций обеспечить при внедрении приложения на предприятии возможность его настройки на особенности ведения учета. Настраиваемая особенность - см. варианты заданий.

Вариант 1

1. Общий реквизит – ДатаСозданияДокумента, тип «Дата». Автоматически означивать при вводе нового документа текущей датой.

2. Документ «Расходы по кассе »

Касса \_\_\_\_\_

Корресп.

От кого

Сумма

счет

☐ Поле «**От кого**» при вводе новой строки означает по умолчанию выбранным значением из справочников: Если Кор счет – 71, то из справочника сотрудников, если 60 или 62, то из справочника контрагентов. (Считать, что в качестве корреспондирующего счета можно брать только один из трех счетов).

3. Задача, возложенная на обработку «Работа с документами»:

Создать новый документ разработанного вида программным путем. Значения полей шапки документа брать произвольно. Сформировать две строки табличной части документа.

4. Настраиваемая особенность ведения учета в приложении:

Возможность исключения из документооборота учета кассовых оборотов по кассам.

Вариант 2

1. Общий реквизит – Автор, тип «СправочникСсылка.Пользователи». Означивается автоматически при открытии нового документа именем пользователя, с которым зашли в задачу.

2. Документ «Обороты по кассе »

Касса \_\_\_\_\_

Тип операции

Корресп. Счет

От кого/кому

Сумма

(приход или расход)

☐ Поле «**От кого/кому**» при вводе новой строки означает по умолчанию выбранным значением из справочников: Если Кор счет – 71, то из справочника сотрудников, Если 60 или 62, то из справочника контрагентов. (Считать, что в качестве корреспондирующего счета можно брать только один из трех счетов).

Вывести в окно сообщений список всех видов документов, расположенных в конфигурации.

Возможность исключения из документооборота учета кассовых оборотов по кассам.

1. Общий реквизит –Комментарий, тип строка, 100 символов.

Со Склада

Материал	Закупочная цена	Количество	Сумма Продукция, для которой отпускается материал	Корр. счет
----------	--------------------	------------	---	---------------

Код Продукции	Наименование	Плановая Себестоимость	Корр.счет, на котором накапливаются затраты
---------------	--------------	---------------------------	--

□ При вводе нового документа поля «Подразделение» , «Склад» означиваются по умолчанию выбранными значениями из соответствующих справочников.

Вывести в окно сообщений даты всех документов разработанного вида за указанный период.

Возможность исключения из документооборота учета движения материалов по подразделениям.

1.	Общий реквизит	–	Организация,	тип
	«СправочникСсылка.Организации»			

Покупатель

Товар	Цена	Количество	% НДС	Сумма НДС	Стоимость с НДС
-------	------	------------	-------	-----------	--------------------

□ Поля «шапки» документов при вводе нового документа означаются по умолчанию выбранными значениями из соответствующих справочников.

☐ Поля «Сумма НДС» и «Стоимость с НДС» означиваются автоматически.

3. Задача, возложенная на обработку «Работа с документами»:

Пометить все введенные документы разработанного вида на удаление.

4. Настраиваемая особенность ведения учета в приложении:

Возможность исключения из документооборота учета движения товаров по складам.

Вариант 5

1. Общий реквизит – ОтветственныйСотрудник, тип «СправочникСсылка.Сотрудники»

2. Документ: «Требование на списание материалов в производство»

Склад \_\_\_\_\_

Материал	Продукция, для которой отпускается материал	Корр. счет	Количество
----------	--	---------------	------------

Замечание .Для ведения документа необходимо создать 2 справочника:

Материалов; готовой продукции .

Справочник готовой продукции должен содержать следующие поля:

Код Продукции	Наимен	Плановая	Корр.Счет, на	Сумма затрат
	ование	себестоимость	котором	накапливаются
			затраты	

☐ Поле «КоррСчет» документа должен автоматически означиваться после выбора продукции.

3. Задача, возложенная на обработку «Работа с документами»:

Отменить проведение всех введенных документов разработанного вида.

4. Настраиваемая особенность ведения учета в приложении:

Возможность исключения из документооборота учета движения материалов по складам.

### **Тема 01.02.15 Перечисления. Оборотные регистры накопления. План видов характеристик.**

Выполнение задания состоит из следующих этапов:

- Создать новую подсистему «ФИО. Группа№». Все новые объекты, созданные в данной работе, поместить в данную подсистему.
- В справочниках один из реквизитов объявлен периодическим, т.е. система должна хранить его историю изменения во времени (такой реквизит помечен \* - см. ниже). Необходимо сформировать новый справочник с одноименным названием (добавить в конце названия справочника цифру 1). Для хранения истории изменения поля, помеченного \*, использовать

периодический регистр сведений. Структуру регистра сведений разработать самостоятельно. Из реквизитов справочника данное поле удалить. В форму элемента справочника ввести одноименный реквизит формы.

- При открытии формы элемента справочника на экране в данном поле ввода должно появиться последнее значение данного реквизита, взятое из регистра сведений.
- При вводе изменений в данном поле ввода это изменение должно сохраняться в регистре сведений.
- Включить в форму списка это поле и обеспечить при открытии формы списка означивание этого поля последним значением из регистра сведений на текущую дату.
- Сформировать отчет – справку, вызываемую из формы списка справочника следующей формы :

*История изменения реквизита < > для объекта < >*

Дата

Значение реквизита

**Замечание.** Объект – это элемент справочника, на котором стоит курсор в форме списка.

- Разработать печатную форму вывода элементов справочника. Периодический реквизит включить в печатную форму справочника. В качестве значения помеченного «\*» реквизита из регистра сведений брать значение на выбранную дату. Таким образом, при запуске печатной формы предварительно должен происходить запрос даты. Использовать конструктор запроса с обработкой результата.

Вариант 1

### **Справочник основных средств (ОС):**

- 1) Инвентарный номер ОС;
- 2) Наименование ОС;
- 3) Материально ответственное лицо (Сотрудник)\*;
- 4) Первоначальный износ;
- 5) Дата \_\_\_\_\_ ввода в эксплуатацию;
- 6) Год выпуска.
- 7) Срок эксплуатации ( в мес.)

Табличная часть элементов справочника содержит список подразделений, которым принадлежало ОС в течение времени эксплуатации и имеет следующую структуру:

- 1) Дата появления в подразделении
- 2) Подразделение.

Вариант 2

### **Справочник товаров:**

- 1) Код товара;

- 2) Наименование товара;
- 3) Основная единица измерения;
- 4) Закупочная цена\* ;
- 5) Розничная цена .

Табличная часть элементов справочника содержит список возможных единиц измерений каждого товара и имеет следующую структуру:

- 1) единица измерения;
  - 2) Коэффициент пересчета относительно основной единицы измерения
- Вариант 3

**1) Справочник сотрудников:**

- 2) Табельный номер сотрудника;
- 3) ФИО сотрудника;
- 4) Тип сотрудника (выбирается из списка);
- 5) Дата поступления на работу;
- 6) Оклад\*;
- 7) Признак членства в профсоюзе.

Табличная часть элементов справочника содержит список детей сотрудника и имеет следующую структуру:

- 1) Пол (выбирается из списка);
- 2) ФИО ребенка.
- 3) Дата рождения

Вариант 4

**Справочник контрагентов:**

- 1) Код контрагента;
- 2) Наименование контрагента;
- 3) Адрес контрагента;
- 4) Расчетный счет;
- 5) Телефон ;
- 6) Размер кредита.

Табличная часть элементов справочника содержит список договоров контрагента и имеет следующую структуру:

1. Код договора;
2. Наименование договора.
3. Дата заключения договора
4. условия договора (текст)

Вариант 5

**Справочник транспортных средств:**

1. Тип транспортного средства (выбирается из списка);
2. Наименование транспортного средства;
3. Объем двигателя;
4. Мощность двигателя;
5. Дата выпуска.

6. Водитель (Сотрудник)\*.

Табличная часть элементов справочника содержит график прохождения технического осмотра (ТО) транспортного средства и имеет следующую структуру:

1. Номер по порядку;
2. Дата прохождения ТО;
3. Примечание (текст).

**Критерии оценки**

За каждый правильный ответ – 1 балл.

За неправильный ответ – 0 баллов.

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

**2.1.2 ТИПОВЫЕ ПРАКТИКО-ОРИЕНТИРОВАННЫЕ  
ЗАДАНИЯ/СИТУАЦИОННЫЕ ЗАДАЧИ**

**Раздел 1. МДК.01.01.Системное программирование**

**Тема 01.01.06 Управление системными ресурсами компьютера**

1. Программа выводит на экран список файлов из корневого каталога. При нажатии клавиши S программа сортирует этот список по алфавиту.
2. Разработать программу рисования красного закрашенного квадрата в центре экрана.

**Раздел 2 МДК.01.02. Прикладное программирование**

**Тема 01.02.04. Введение в объектно-ориентированное программирование**

1. На форме расположен TextBox с именем textBox1. Класс создан в отдельном файле с именем Class1.cs. Необходимо определить, какое значение поле name будет иметь при выполнении

следующего кода и почему?

```

Form1.cs*  Class1.cs  Form1.cs [Конструктор]*
button.student
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace button
{
    class student
    {
        public string name = "Bob";
        public student(string name)
        {
            this.name = name;
        }
    }
}

```

```

Form1.cs*  Class1.cs  Form1.cs [Конструктор]*
button.Form1
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace button
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            student tom = new student("Mkk") { name = "Mon" };
            textBox1.Text = tom.name;
        }
    }
}

```

2. Дан листинг программы. Необходимо определить, что выведет на экран следующая программа и почему?

Class1.cs	Form1.cs
<pre> namespace button {     class Person </pre>	<pre> private void Form1_Load(object sender, EventArgs e) { </pre>

<pre> {     int age = 26;     string name = "Tom";      public Person(int age, string name)     {         this.age = age;         this.name = name;     } } </pre>	<pre> Person person = new Person(19, "Bob"); textBox1.Text = person.name;  } </pre>
--	---

3. Дан листинг программы. Что выведет на экран следующая программа и почему?

Class1.cs	Form1.cs
<pre> class Auto // легковой автомобиль {     public int Seats { get; set; } // количество сидений     public Auto(int seats)     {         Seats = seats;     } } class Truck : Auto    // грузовой автомобиль {     public decimal Capacity { get; set; } // грузоподъемность     public Truck(int seats, decimal capacity)     {         Seats = seats;         Capacity = capacity;     } } </pre>	<pre> Truck truck = new Truck(2, 1.1m); </pre>



4. Дан листинг программы. Что выведет на экран следующая программа и почему?

Class1.cs	Form1.cs
<pre> class Person {     public string Name { get; set; } = "Ben";      public Person(string name)     {         Name = "Tim";     } }  class Employee : Person {     public string Company { get; set; }      public Employee(string name, string company)         : base("Bob")     {         Company = company;     } } </pre>	<pre> Employee emp = new Employee("Tom", "Microsoft") { Name = "Sam" }; </pre>

5. Дан листинг программы. Что будет выведено на экран в результате выполнения следующей программы и почему?

Class1.cs	Form1.cs
<pre> namespace button {     class Person     {         public string Name { get; set; }         public virtual string Display()         { </pre>	<pre> Person person = new Manager { Name = "Bob", Company = "Microsoft" }; Employee employee = (Employee)person; textBox1.Text=employee.Display(); </pre>

<pre>         return ("Person"+ Name);     } } class Employee : Person {     public string Company { get; set; }     public override string Display()     {         return ("Employee" + Name);      } } class Manager : Employee {     public override string Display()     {         return ("Manager" + Name);      } } </pre>	
---	--

### Тема 01.02.08 Работа со строками в Windows

1. Написать программу, которая сравнивает две строки и выводит ту строку, которая больше.
2. Дана строка, состоящая из одного слова. Написать программу, которая будет проверять, является ли это слово палиндромом.
3. Дана строка произвольной длины. Необходимо отсортировать символы этой строки в алфавитном порядке.

### Тема 01.02.09 Принципы разработки Windows приложений в .NET (Windows Forms Application)

1. Необходимо создать приложение, в котором пользователю будет предложен список животных и птиц. При выборе животного вывести на экран его изображение.

2. Создать Windows-приложение. При запуске программы пользователю должна быть представлена форма авторизации. После успешного ввода логина и пароля, пользователь переходит на вторую форму, где ведется текущая успеваемость студента.

3. Создать карту звездного неба Солнечной системы. При наведении на планету должно показываться имя планеты (использовать элемент управления ToolTip). Если пользователь хочет просмотреть справочную информацию о выбранной планете, он должен щелкнуть по ней. При щелчке мыши открывается новая форма с информацией.

### Тема 01.02.12 Контроль версий GIT

Выполнить задания:

- Найти репозиторий –newgit (автор Zakirova Regina).
- Скопировать ссылку (clone)
- Создать новую папку
- Клонировать репозиторий newgit

### Критерии оценки

За каждый правильный ответ – 1 балл.

За неправильный ответ – 0 баллов.

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

## 2.1.3 ВОПРОСЫ ЭКЗАМЕНА/ЗАЧЕТА

### Раздел 1. МДК 01.01. Системное программирование

#### Вопросы к дифференцированному зачету

1. Процессор: микропроцессор и его назначение, состав микропроцессора, режимы работы микропроцессора.
2. Материнская плата
3. Память компьютера: внутренняя память компьютера, внешняя память компьютера
4. Периферийные устройства компьютера: клавиатура, манипулятор «мышь», цифровой аудиоканал.
5. Системы счисления.
6. Перевод чисел в различные системы счисления
7. Пользовательские регистры процессора
8. Сегментированная модель памяти
9. Организация адресного доступа к оперативной памяти
10. Стековая память
11. Описание интерфейса отладчика
12. Функциональные клавиши
13. Основные команды отладчика.
14. Синтаксис языка Ассемблер
15. Структура машинной команды
16. Способы задания операндов команды
17. Адресация к памяти
18. Команды передачи данных
19. Арифметические команды
20. Логические операторы и команды сдвига
21. Команды передачи управления
22. Команды цикла
23. Основные команды отладчика
24. Команды передачи данных
25. Арифметические команды
26. Логические операторы и команды сдвига
27. Команды передачи управления
28. Команды цикла
29. Обзор программ для компоновки и отладки программ Ассемблера: Tasm, Masm, Fasm
30. Стандарты языка Ассемблер
31. Структура программы
32. Данные в Ассемблере

33. Аппаратные прерывания. Приоритет прерываний. Запрет и маскирование аппаратных прерываний.
34. Работа со стеком в Ассемблере.
35. Подпрограммы
36. Макросы
37. Написание программ с использованием циклических конструкций
38. Написание программ с использованием подпрограмм
39. Написание программ с использованием макросов
40. Управление системными ресурсами IBM PC
41. Программный доступ к CMOS-памяти
42. Программирование клавиатуры
43. Микросхема таймера Intel 8253 и ее программирование
44. Работа системных часов
45. Задержка операций и контроль временных интервалов
46. Использование счетчика тактов процессора в качестве таймера
47. Управление звуком
48. Генерация случайных чисел
49. Программирование мыши
50. Написание резидентных программ
51. Структура резидентной программы
52. Проблема повторной загрузки и инициализации резидентной программы
53. Выгрузка резидентной программы из памяти компьютера
54. Организация текстового видеорежима
55. Управление видеоконтроллером в текстовом режиме
56. Средства управления шрифтами в текстовом режиме
57. Организация графического видеорежима
58. Управление видеоконтроллером в графическом режиме
59. Работа с цветовой палитрой

### **Критерии оценки**

Оценка «отлично» выставляется, если практические работы выполнены в объеме не менее 85% и зачтены с оценкой отлично, результаты ответа на устный вопрос – отлично или хорошо.

Оценка «хорошо» выставляется, если практические и лабораторные работы выполнены в объеме от 70% до 84 % и зачтены с оценкой хорошо или отлично, результаты ответа на устный вопрос – хорошо.

Оценка «удовлетворительно» выставляется, если практические и лабораторные работы выполнены в объеме от 55 % до 69% и зачтены с

оценкой удовлетворительно, результаты ответа на устный вопрос – удовлетворительно.

Оценка «неудовлетворительно» выставляется, если практические и лабораторные работы выполнены в объеме менее 55% и зачтены с оценкой удовлетворительно, результаты ответа на устный вопрос – неудовлетворительно.

## **2.2 Задания для оценки освоения МДК 01.02 Прикладное программирование**

1. Язык C#. Преимущества и особенности языка. Родительский класс. Что такое поля и методы?
2. Схема типов в языке C#. Что относится к значимым типам, к ссылочным.
3. Типы данных. Логический, арифметический целочисленный, с плавающей точкой, с фиксированной точкой, символьный тип.
4. Явное и неявное преобразование типов. Между какими типами существует неявное преобразование. Какие типы необходимо преобразовывать явно.
5. Increment, decrement. Арифметические операции. Операции отношения. Сдвиг. Логические операции.
6. Класс Random и его функции.
7. Массивы. Синтаксис объявления одномерных массивов в языке C#. Объявление константных массивов и с вызовом конструктора массивов.
8. Статические одномерные и многомерные массивы. Синтаксис о
9. Объявление массивов. Инициализация массивов. Заполнение массива случайными элементами.
10. Объявление динамические одномерные и многомерные массивы. В чем отличие от статических массивов?
11. Многомерные массивы. Объявление и инициализация многомерных массивов. Методы определения измерений.
12. Массив массивов. Объявление и инициализация «зубчатых» массивов».
13. Создание методов в языке C#. Синтаксис метода типа void. Синтаксис вызова метода.
14. Создание методов в языке C#. Синтаксис метода с указанием типа результата. Синтаксис вызова метода. Описание тела метода.
15. Список формальных аргументов метода. Соответствие фактических и формальных аргументов.
16. Операторы языка C#. Операторы выбора if, switch.

17. Операторы языка C#. Операторы условного перехода goto, break, continue, return.
18. Операторы языка C#. Операторы цикла for, while, do while, foreach.
19. Тип данных Char. Основные методы.
20. Тип данных String. Операции над строками. Основные методы и свойства. Метод ToCharArray.
21. Язык программирования 1С. Типы данных, операторы присваивания, встроенные процедуры и функции.
22. Выгрузка/загрузка файловой информационной базы.
23. Язык программирования 1С. Операторы ветвления.
24. Язык программирования 1С. Работа с циклами.
25. Процедуры и функции. Работа с параметрами.
26. Универсальные коллекции значений. Массивы.
27. Обзор основных объектов конфигурации, их назначение.
28. Создание констант, настройка их свойств.
29. Создание линейных справочников, настройка их свойств.
30. Создание документов, настройка их свойств.

#### Примеры типовых заданий

1. В программе генерируется случайным образом трехзначное число. Необходимо поменять местами 2 и 3 цифру в числе.
2. Пользователь с клавиатуры задает четырехзначное число. Необходимо поменять местами 2 и 4 цифру в числе.
3. Пользователь задает три строки. Необходимо вывести строку, которая больше (имеет большее количество символов). Две оставшиеся преобразовать в одну.
4. Даны n натуральных чисел. Вычислить сумму ряда. n вводится с клавиатуры.

$$\frac{2 - 1}{2 * (2 + 1)} + \frac{3 - 1}{3 * (3 + 1)} + \dots + \frac{(n - 1)}{n(n + 1)}$$

5. Подсчитать факториал числа при помощи рекурсии.
6. Посчитать сумму, произведение, разность, деление чисел, введенных с клавиатуры. Ввод данных осуществляется до тех пор, пока не введен 0. (решить задачу циклом с предусловием)
7. Посчитать сумму, произведение, разность, деление чисел, введенных с клавиатуры. Ввод данных осуществляется до тех пор, пока не введен 0. (решить задачу циклом с постусловием)
8. Дан одномерный массив из 100 элементов, заполненный случайными числами. Необходимо заменить все 0 в массиве на 1, и все четные элементы массива заменить на значение 5.

9. Дан одномерный массив из 20 элементов, заполненный случайными числами. Необходимо поменять местами минимальное и максимальное число данного массива.
10. Найти максимальный и минимальный элемент диагонали двумерного массива
11. Найти сумму элементов  $i$  строки двумерного массива
12. Найти количество повторений элементов массива. В ответ вывести два массива: первый – заполненный случайными значениями; второй – массив, содержащий повторения.
13. Написать метод, позволяющий посчитать значения уравнения:

$$y = \sqrt{(x^2 + 15)}$$

Выражение  $(x^2 + 15)$  должно вычисляться в первой функции. После подсчета результата функция должна вызвать второй метод, где происходит взятие корня.

14. Написать два метода. Первый метод генерирует случайные числа для матриц. Второй - рассчитывает сумму матриц.
15. Написать метод подсчета среднеарифметического значения двумерного массива.
16. Дана произвольная строка, заполненная символами. Определить, сколько раз встречается заданный символ в строке.
17. Дана строка произвольной длины. Заменить все четные символы строки на знак «\*». Вывести полученную строку в формате: первая буква заглавная, все остальные строчные.
18. Даны две строки. Необходимо скопировать подстроку первой строки, начиная с 3 символа, заканчивая 7. Например, дана строка «C# – объектно-ориентированный язык программирования.». Необходимо во вторую строку скопировать текст «– обь».
19. Дана строка, состоящая из одного слова. Необходимо проверить, является ли это слово палиндромом.
20. Дана строка произвольной длины. Необходимо отсортировать символы этой строки в алфавитном порядке.



### **3 КОНТРОЛЬ ПРИОБРЕТЕНИЯ ПРАКТИЧЕСКОГО ОПЫТА. ОЦЕНКА ПО УЧЕБНОЙ И ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

#### **3.1. Общие положения**

Предметом оценки по учебной и производственной практике являются:

- 1) профессиональные и общие компетенции;
- 2) практический опыт и умения.

Оценка по практике выставляется на основании Отчета по учебной и производственной практике, содержащем задание на практику и аттестационный лист с указанием видов работ, выполненных обучающимся во время учебной и производственной практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила учебная и производственная практика.

#### **3.2. Требования к Отчету по учебной и производственной практике**

Требования к Отчету по учебной и производственной практике представлены в Методических указаниях по практике.

## 4 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ЭКЗАМЕНА (КВАЛИФИКАЦИОННОГО)

### I ПАСПОРТ

#### Назначение:

КОС предназначены для контроля и оценки результатов освоения профессионального модуля *ПМ.01. Разработка программных модулей программного обеспечения для компьютерных систем* по специальности СПО: 09.02.03 Программирование в компьютерных системах.

### II ЗАДАНИЕ ДЛЯ ЭКЗАМЕНУЮЩЕГОСЯ. Типовой вариант

Коды проверяемых профессиональных и общих компетенций: в части освоения основного вида деятельности (ВД): Разработка программных модулей программного обеспечения для компьютерных систем и соответствующих профессиональных компетенций (ПК):

ПК.1.1. Выполнять разработку спецификаций отдельных компонент.

ПК.1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК.1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК.1.4. Выполнять тестирование программных модулей.

ПК.1.5. Осуществлять оптимизацию программного кода модуля.

ПК.1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

ПК.1.7. Осуществлять работу с системой контроля версий.

#### Инструкция

При выполнении задания квалификационного экзамена Вам необходимо ознакомиться с представленной документацией.

Задание для квалификационного экзамена состоит из следующей документации/файлов:

- |                  |                              |
|------------------|------------------------------|
| 1.data.xlsx      | (данные для импорта)         |
| 2.StyleGuide.pdf | (руководство по стилю)       |
| 3.prezent.ppsx   | (презентация работы системы) |
| 4.Logo/          | (папка с логотипами)         |

Вам необходимо разработать информационную систему «Библиотека». К окончанию квалификационного экзамена убедитесь, что

1. Вы следуете предоставленному руководству по стилю во всех частях системы.
2. Вы предоставляете соответствующие проверки и сообщения об ошибках во всех частях системы.
3. Все соответствующие кнопки/ссылки работают.

4. Вы используете соответствующие соглашения об именах для всех частей системы по мере необходимости.
5. Логика приложения соответствует варианту, изложенному в презентации.
6. Все элементы клиентского приложения отображаются на русском языке.
7. Код приложения сопровождается комментариями.

Для предоставления результата выполнения данного задания необходимо сохранить работу, оформленную в соответствии с требованиями:

- файл базы данных должен быть назван по шаблону kv\_exNN\_BD, где NN-номер компьютера в классе;
- проект приложения должен быть назван по шаблону kv\_exNN2, где NN-номер участника и сохранен в папке на рабочем столе под названием KV\_ex.

СУБД для реализации БД и среда разработки студентом выбирается самостоятельно.

Время выполнения 3 часа.

### Текст задания:

В рамках разработки информационной системы предлагается ER-модель Базы Данных учебного заведения.



### Задание. Этап 1.

Используя файл ER-model, необходимо выбрать средство хранения данных из предложенных на Вашем ПК.

1. Определить связи между сущностями.

2. Определить тип данных для каждого атрибута всех таблиц.
3. Заполнить таблицы БД для возможности проверки запросов (не менее 5 записей в каждой)
4. Реализовать запросы к БД:
  - a. Нужно получить информацию о студентах, зачисленных в первой половине августа 2018 года.
  - b. Получить информацию обо всех студентах, кроме тех, кто учится по специальности Программирование в компьютерных системах.
  - c. Узнать, на какую оценку студенты группы Пр-15-1 сдали экзамен по информатике.
  - d. Найти суммарные баллы, полученные студентами по результатам семестра. Итоговый набор должен содержать следующие сведения: код студента, его фамилию, код группы и итоговую сумму оценок по сданным экзаменам. Информация должна быть отсортирована по сумме набранных баллов.
  - e. Определить учебные группы, имеющие по информатике более высокий средний балл, чем на курсе в целом.

Используя любое инструментальное средство разработки оконных приложений, необходимо создать клиентское приложение с возможностью работы с хранилищем БД.

Необходимо:

1. Установить связи между таблицами БД.
2. Реализовать возможность ввода информации в БД через формы.
3. Заполнить таблицы БД, для возможности проверки запросов.
4. Реализовать отчетные формы на основании запросов к БД:
  - a. Нужно получить информацию о студентах, зачисленных в первой половине августа 2018 года.
  - b. Получить информацию обо всех студентах, кроме тех, кто учится по специальности Программирование в компьютерных системах.
  - c. Узнать, на какую оценку студенты группы Пр-15-1 сдали экзамен по информатике.
  - d. Найти суммарные баллы, полученные студентами по результатам семестра. Итоговый набор должен содержать следующие сведения: код студента, его фамилию, код группы и итоговую сумму оценок по сданным экзаменам. Информация должна быть отсортирована по сумме набранных баллов.

- е. Определить учебные группы, имеющие по информатике более высокий средний балл, чем на курсе в целом.
5. Подготовить отчеты (см.п.4), с возможностью импорта в MS Word или MS Excel.

### **III ПАКЕТ ЭКЗАМЕНАТОРА**

#### **IIIa УСЛОВИЯ**

Комплект представляет собой пакет документации/файлов, содержащий непосредственно задание квалификационного экзамена, выполнение которого рассчитано на 3 часа; условия его выполнения; критерии оценки; данные для импорта; руководство по стилю; презентация работы системы и папка с логотипами библиотеки.

Членам комиссии (экзаменаторам) доступна вся информация, студентам – задание квалификационного экзамена, данные для импорта, руководство по стилю, презентация работы системы и папка с логотипами библиотеки.

Для проведения квалификационного экзамена в структурном подразделении «Многопрофильный колледж» ФГБОУ ВО «Магнитогорский государственный технический университет им. Г.И.Носова» предусмотрена аккредитованная площадка для проведения чемпионата по стандартам WorldSkills, компетенция «Программные решения для бизнеса» с соответствующим техническим и программным обеспечением.

#### **IIIб КРИТЕРИИ ОЦЕНКИ**

Критерии сформированы на основе оценки конкурсных заданий чемпионатов WorldSkills по компетенции «Программные решения для бизнеса».

	Критерии	Оценка
A	Анализ и проектирование систем	12,20
B	Разработка программного обеспечения	19,00
C	Стандарты разработки	1,00
D	Системная документация	0,80
	Итого	33

Оценка «отлично» – 22 балла и выше;

Оценка «хорошо» – 15-21 баллов;

Оценка «удовлетворительно» 8-14 баллов;

Оценка «неудовлетворительно» – менее 8 баллов.