

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

Комплект контрольно-оценочных средств
по ПМ. 01 «Разработка программных модулей программного обеспечения для
компьютерных систем»
специальности **09.02.03 Программирование в компьютерных системах**
базовой подготовки

УТВЕРЖДЕНО

Директор Колледжа ФГБОУ ВО «МАГУ»



_____/ Козлова Н.В./
Ф.И.О.

Мурманск
2019 г.

КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО МОДУЛЮ

Общие сведения

1	Специальность	09.02.03 Программирование в компьютерных системах
2	Форма обучения	очная
3	Профессиональный модуль	ПМ.01. Разработка программных модулей программного обеспечения для компьютерных систем
4	Форма промежуточной аттестации	Дифференцированный зачет МДК 01.02 Экзамен МДК 01.01 Экзамен (квалификационный) по ПМ.01

Перечень формируемых знаний, умений и компетенций

	Условное обозначение знаний, умений, компетенций	Элементы оценивания
Практический опыт	П.1	разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования;
	П.2	разработки кода программного продукта на основе готовой спецификации на уровне модуля;
	П.3	использования инструментальных средств на этапе отладки программного продукта;
	П.4	проведения тестирования программного модуля по определенному сценарию;
Умения	У1	осуществлять разработку кода программного модуля на современных языках программирования;
	У2	создавать программу по разработанному алгоритму как отдельный модуль;
	У3	выполнять отладку и тестирование программы на уровне модуля;
	У4	оформлять документацию на программные средства;
	У5	использовать инструментальные средства для автоматизации оформления документации;
Знания	З1	основные этапы разработки программного обеспечения;
	З2	основные принципы технологии структурного и объектно-ориентированного программирования;
	З3	основные принципы отладки и тестирования программных продуктов;
	З4	методы и средства разработки технической документации.
Общие компетенции	ОК.1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
	ОК.2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
	ОК.3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
	ОК.4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

	ОК.5	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
	ОК.6	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
	ОК.7	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
	ОК.8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
	ОК.9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
Профессиональные компетенции	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

1.3. Показатели оценки результата освоения общих и профессиональных компетенций

Результаты (освоенные общие компетенции)	Основные показатели результатов подготовки	Формы и методы контроля и оценки
ПК 1.1. Выполнять разработку спецификаций отдельных компонент.	Точность определения этапов разработки ПО. Правильность и точность разработки алгоритма поставленной задачи.	Текущий контроль знаний в форме: - в форме опросов; - самостоятельных работ по темам МДК;
ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.	Реализация всех функций программного продукта, представленных в спецификациях, оформленных в соответствии с требованиями ГОСТ ЕСПД и ГОСТ кл. 34 по разработке информационных систем, в среде программирования.	- защиты практических работ. Выполнение индивидуальных заданий.
ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.	Реализация процесса отладки программы на уровне модуля.	Контроль знаний в форме дифференцированного зачета по МДК.
ПК 1.4. Выполнять тестирование программных модулей.	Обоснование выбора методики тестирования программного продукта. Проведение тестирования в соответствии с правилами выбранной методики.	Дифференцированный зачет по учебной практике и по разделу профессионального модуля.
ПК 1.5. Осуществлять оптимизацию программного кода модуля.	Реализация основных принципов технологии объектно-ориентированного	Экзамен квалификационный по профессиональному

	программирования. Осуществление контроля объема памяти и времени обработки результатов.	модулю.
ПК 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.	Оформление документации в соответствии с ГОСТ ЕСПД и ГОСТ кл. 34 по разработке информационных систем	

Формы и методы контроля и оценки результатов обучения должны позволять проверять у обучающихся не только сформированность профессиональных компетенций, но и развитие общих компетенций и обеспечивающих их умений.

Результаты (освоенные общие компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	– демонстрация интереса к будущей профессии	Наблюдение за поведением обучающегося в аудитории и учебной лаборатории, анализ текущей успеваемости. Наблюдение за формированием у обучающегося устойчивого интереса к специальности.
ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.	– обоснование выбора методов и способов решения профессиональных задач в области разработки информационных систем; – демонстрация эффективности и качества выполнения профессиональных задач.	Интерпретация результатов наблюдений за обучающимся к самоорганизации. Оценка поведения обучающегося в конкретных ситуациях.
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.	– самостоятельность в принятии решения при выполнении учебных и профессиональных задач и нести за них ответственность	Анализ выполнения практических работ. Поиск и нахождение решений для нестандартных ситуаций.
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	– нахождение и использование информации для эффективного выполнения профессиональных задач, профессионального и личностного развития.	Наблюдение за способностью обучающегося пользоваться специальной литературой, справочниками. Подготовка сообщений, использование электронных источников информации.
ОК 5. Использовать информационно – коммуникационные технологии в профессиональной деятельности.	– демонстрация навыков использования информационно-коммуникационные технологии в профессиональной деятельности.	Наблюдение за навыками работы с автоматизированными программами, Интернет-ресурсами.
ОК 6. Работать в коллективе и в команде, эффективно общаться	– взаимодействие с обучающимися,	Наблюдение за коммуникабельностью

с коллегами, руководством, потребителями.	преподавателями и мастерами в ходе обучения	обучающегося. Наблюдение за поведением обучающегося в группе. Оценка развития ответственности и доброжелательности.
ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.	– проявление ответственности за работу подчиненных, результат выполнения заданий.	Моделирование социальных и профессиональных ситуаций на занятиях и во время прохождения учебной практики.
ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.	– результативность информационного поиска с целью самообразования	Контроль выполнения индивидуальной самостоятельной работы обучающегося. Сдача зачетов, квалификационных экзаменов
ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	– проявление интереса к инновациям в области разработки информационных технологий, операционных систем и автоматизации функций управления предприятием, фирмой, подразделением организации	Интерпретация наблюдения за деятельностью обучающегося по подготовке и выполнению практической деятельности.

1.4. Порядок и условия организации экзамена (квалификационного)

Экзамен (квалификационный) представляет собой выполнение комплексной практической работы.

Задания и показатели оценки результатов освоения программы модуля

Номер и содержание задания	Оцениваемые компетенции	Показатели оценки результата (критерии оценки)
Задание № 1 Выполнение проекта в среде разработки программного обеспечения (индивидуальные задания, выдаются до начала учебной и производственной практики).	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.	1. Правильность и точность разработки алгоритма поставленной задачи. 2. Корректность и работоспособность проекта. 3. Правильность оформления проекта
Задание № 2 Выполнение тестирования, отладки разработанного продукта или разработка компонента проектной и технической документации (индивидуальные задания, выдаются до начала учебной и производственной практики).	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. ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый	Правильность и рациональность принятых решений по тестированию и отладке программы на уровне модуля; составлению проектной и технической документации.

Задание № 3 Выполнение теста на ПК	<p>интерес</p> <p>ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.</p> <p>ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.</p> <p>ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.</p> <p>ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.</p>	Не менее 61% выполнения теста
------------------------------------	--	-------------------------------

1.5. Критерии оценки

Освоенные ПК и ОК	Показатель оценки результата (критерии оценки)	Соответствует/ Не соответствует
ПК 1.1. ОК 1 - 9	Правильность и точность разработки алгоритма поставленной задачи	Соответствует/ Не соответствует
ПК 1.2. ОК 1 - 9	Корректность и работоспособность проекта. Правильность оформления проекта.	Соответствует/ Не соответствует
ПК 1.3. ОК 1 - 9	Правильность и рациональность принятых решений по тестированию и отладке программы на уровне модуля.	Соответствует/ Не соответствует
ПК 1.4. ОК 1 - 9	Правильность и рациональность принятых решений по тестированию и отладке программы на уровне модуля.	Соответствует/ Не соответствует
ПК 1.5. ОК 1 - 9	Корректность и работоспособность проекта.	Соответствует/ Не соответствует
ПК 1.6. ОК 1 - 9	Правильность составления проектной и технической документации.	Соответствует/ Не соответствует

ОЦЕНОЧНАЯ ВЕДОМОСТЬ по профессиональному модулю

_____,
Фамилия, имя, отчество студента

обучающийся (аяся) на 3 курсе по специальности 09.02.03 Программирование в компьютерных системах освоил (а) программу профессионального модуля «Разработка программных модулей программного обеспечения для компьютерных систем» в объеме 910 час.

№	Профессиональные и общие компетенции	Критерии оценки	Соответствует	Не соответствует	Замечания
1	ПК 1.1. Выполнять разработку спецификаций отдельных компонент ОК 6, ОК 7	-Правильность и точность разработки алгоритма поставленной задачи (на основании отчетов по учебной и производственной практике) -Демонстрация умения работать в команде, совместного выполнения принятого решения, приоритетное использование делового стиля общения (по заключению руководителя производственной практики) -Правильность выполнения тестового задания не менее 61%			
2	ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля ОК 1, ОК 3	-Корректность и работоспособность выполненного проекта. -Правильность оформления проекта (на основании отчетов по учебной и производственной практике) -Правильность выполнения тестового задания не менее 61%			
3	ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств	-Правильность и рациональность принятых решений по тестированию и отладке программы на уровне модуля (на основании отчетов по учебной и производственной практике) -Правильность выполнения тестового задания не менее 61%			
4	ПК 1.4. Выполнять тестирование программных модулей ОК 2, ОК 4	-Правильность и рациональность принятых решений по тестированию и отладке программы на уровне модуля (на основании отчетов по учебной и производственной практике) -Правильность выполнения тестового задания не менее 61%			
5	ПК 1.5. Осуществлять оптимизацию программного кода модуля ОК 8, ОК 9	-Корректность и работоспособность проекта (на основании отчетов по учебной и производственной практике) -Положительная характеристика руководителя производственной практики -Правильность выполнения тестового задания не менее 61%			
6	ПК 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций ОК4, ОК 5	-Правильность составления проектной и технической документации (на основании отчетов по учебной и производственной практике) -Правильность выполнения тестового задания не менее 61%			

Заключение экзаменационной (аттестационной комиссии):

Вид профессиональной деятельности «Разработка программных модулей программного обеспечения для компьютерных систем» _____

освоен (не освоен)

Подписи членов экзаменационной комиссии:

_____	_____	_____
<i>должность</i>	<i>подпись</i>	<i>Ф.И.О.</i>
_____	_____	_____
<i>должность</i>	<i>подпись</i>	<i>Ф.И.О.</i>
_____	_____	_____
<i>должность</i>	<i>подпись</i>	<i>Ф.И.О.</i>

Дата _____ . _____ .20_____

1.6. Типовые контрольные задания и методические материалы для промежуточной аттестации по междисциплинарным курсам

В ходе освоения профессионального модуля применяются следующие формы контроля:

- устный опрос;
- защита практической работы;
- решение задач по теме;
- терминологический диктант;
- тестирование.

Критерии оценки:

Опроса

Оценка «отлично» ставится, если: обучающийся полно излагает материал, дает правильное определение основных понятий; обнаруживает понимание материала, используя терминологический аппарат, может обосновать свои суждения, привести необходимые примеры, делать выводы и обобщения; излагает материал последовательно и правильно с точки зрения норм литературного языка.

«Хорошо» – обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для отметки «5», но допускает 1 – 2 ошибки, которые сам же исправляет, и 1 – 2 недочета в последовательности и логичности излагаемого.

«Удовлетворительно» – обучающийся обнаруживает знание и понимание основных положений данной темы, но: излагает материал неполно и допускает неточности в определении понятий или их трактовке; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.

Оценка «2» ставится, если обучающийся обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и терминов, искажающие их смысл, беспорядочно и неуверенно излагает материал.

Выполнения практической работы

Оценка «отлично» ставится, если обучающийся выполнил работу в полном объеме с соблюдением необходимой последовательности действий; соблюдает правила техники безопасности; правильно и в соответствии с требованиями (ГОСТ) оформляет программный код и соответствующую документацию; правильно выполняет отладку программы и анализ ошибок; предложенное решение задачи аргументированно и обоснованно с использованием терминологического аппарата.

Оценка «хорошо» ставится, если обучающийся выполнил требования к оценке «5», но допущены 2 – 3 недочета.

Оценка «удовлетворительно» ставится, если работа выполнена не полностью, но реализованной части достаточно для получения правильных результатов и выводов; в ходе проведения работы были допущены ошибки, частично соблюдены требования по оформлению программного кода и документации.

Оценка «неудовлетворительно» ставится, если работа выполнена не полностью или объем выполненной части работы не позволяет сделать правильных выводов и получить правильных результатов (программа не работает); обучающийся не может ответить ни на один из поставленных вопросов, связанных с написанием программы и выбором решения поставленной задачи.

Решения задач

Оценка «отлично» ставится, если обучающийся решил задачу в полном объеме с соблюдением необходимой последовательности действий; правильно и в соответствии с требованиями (ГОСТ) оформляет блок-схему и программный код; предложенное решение задачи аргументированно и обоснованно с использованием терминологического аппарата.

Оценка «хорошо» ставится, если обучающийся выполнил требования к оценке «5», но допущены 2 – 3 недочета.

Оценка «удовлетворительно» ставится, если работа выполнена не полностью, но реализованной части достаточно для получения правильных результатов и выводов; в ходе проведения работы были допущены ошибки, частично соблюдены требования по оформлению блок-схем и программного кода.

Оценка «неудовлетворительно» ставится, если задание выполнено не полностью или объем выполненной части работы не позволяет сделать правильных выводов и получить правильных результатов; обучающийся не может ответить ни на один из поставленных вопросов, связанных с составлением программы и построением блок-схемы.

Выполнения терминологического диктанта

Оценка «отлично» ставится, если процент правильных ответов 91-100.

Отметка «хорошо» ставится, если процент правильных ответов 81-90 с незначительными ошибками, т.е. использованы синонимы, не искажающие смысл терминов.

Отметка «удовлетворительно» ставится, если процент правильных ответов 71-80.

Отметка «неудовлетворительно» ставится, если процент правильных ответов менее 71.

При оценивании учитывается полнота и логика изложения. Допускается давать определение словами-синонимами, неискажающими смысловое значение определения.

Тестовых форм контроля

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	неудовлетворительно

Примеры заданий

МДК.01.01 Системное программирование

Типовые вопросы для устного опроса по теме «Основные структуры языка: циклы. Цикл с условием»

1. Что такое цикл?
2. Какие средства предоставляются языком C# для организации циклов?
3. Что такое цикл с параметром?
4. Приведите общие формы записи арифметических операторов цикла и объясните, как они действуют.
5. Что такое итерационный цикл?
6. Представьте общую форму записи оператора цикла итерационного типа с предусловием и объясните, как он действует.
7. Представьте общую форму записи оператора цикла итерационного типа с постусловием и объясните, как он действует.
8. Когда применяются операторы Break и Continue? Можно ли выйти из цикла, не завершая его?
9. Как нужно организовать цикл, чтобы можно было войти в тело цикла?

Типовая практическая работа по теме «Составление программ разветвляющейся структуры»

Цель: изучить операторы, используемые для организации ветвления в программе.
Познакомится с логическими выражениями и операциями.

Ход работы:

1. Ознакомиться с кратким теоретическим материалом (+ конспект лекций).
2. Выполнить индивидуальное задание:
 - составить блок-схему решения задачи;
 - создать проект в среде программирования;
 - написать программу с использованием оператора if ... else, выполнить отладку;
 - подобрать наборы исходных данных для тестирования всех ветвей программы.
3. Сформулировать выводы по проделанной работе.

Методические указания по выполнению работы

1. Теоретический материал

Операторы ветвления управляют потоком выполнения программы. К основным операторам ветвления в C# относятся: условный оператор if ... else и переключатель switch.

Условные операторы позволяют выбрать один из вариантов выполнения действий в зависимости от каких-либо условий. Условие – это логическое выражение, т.е. выражение, результатом которого является логическое значение истина или ложь. Выражение может иметь арифметический тип. Если оно не равно 0, то условие считается истинным. Если равно 0, то условие считается ложным.

Оператор if (условный оператор) выбирает один из двух вариантов последовательности вычислений.

Синтаксис условного оператора вид:

```
if (выражение_1)
    {оператор_выбора_1};
[else
    {оператор_выбора_2;}]
```

Это полная форма оператора (рисунок 1), программирующая структуру полного ветвления. Обычно *выражение* – это некоторое условие, содержащее операции отношения и логические операции. Значение выражения приводится к целому и интерпретируется в соответствии с правилом: равно нулю – ложь, не равно нулю – истина (т.е. если условие в «выражение_1» истинно, то выполняется «оператор_выбора_1», если условие ложно, то выполняется «оператор_выбора_2»).

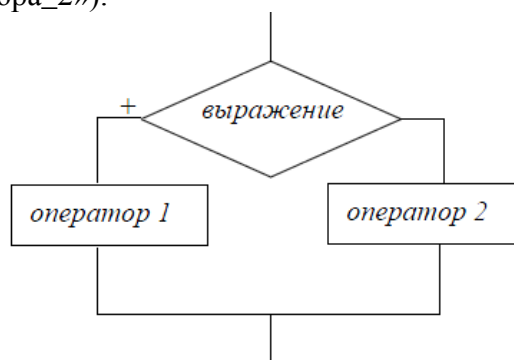


Рисунок 1 – Блок-схема оператора ветвления

В качестве операторов нельзя использовать описания и определения.

Если в случае истинности условия необходимо выполнить несколько операторов, то их заключают в фигурные скобки (т.е. использовать составные операторы и блоки).

Особенности синтаксиса условного оператора:

- выражение записывается в круглых скобках;

– точка с запятой после оператора 1 ставится обязательно.

Условный оператор можно расширить для проверки нескольких условий:

```
if (x < 0)
    { Console.WriteLine("Отрицательная величина"); }
else if (x > 0)
    { Console.WriteLine("Положительная величина"); }
else Console.WriteLine("Ноль");
```

Конструкций `else if` может быть несколько.

Допускается сокращенная форма условного оператора, в которой отсутствует `else` и оператор_2. Например,

```
if (x > 0) x = -x;
```

Оператор_1 и оператор_2 могут также быть условными. Каждое `else` соответствует ближайшему `if`.

Пример выполнения задания.

Условие: с клавиатуры вводятся два числа. Если их произведение отрицательно, то умножить его на -2 и вывести на экран; в противном случае увеличить его в три раза и вывести на экран.

Замечание: 0 не является как положительным, так и отрицательным числом, следовательно, для проверки произведения на равенство нулю, должен использоваться еще один оператор проверки условия.

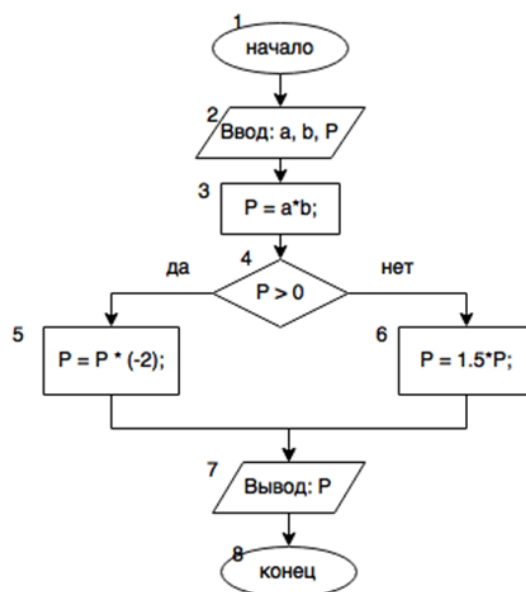
Исходный код программы представлен в листинге 1.

```
1 // *****
2 /* Практическая работа № 2 *
3 /* Выполнил: Иванов И.И., группа 2ПКС *
4 /* Задание: составить программу работы алгоритма ветвления *
5 //*****
6 using System;
7 using System.Collections.Generic;
8 using System.Linq;
9 using System.Text;
10
11 namespace ConsoleApplication2
12 {
13     class Program
14     {
15         static void Main(string[] args)
16         {
17             int a, b, P = 0; // объявление переменных
18
19             Console.WriteLine("Практическая работа № 2 ");
20             Console.Write("Введите два числа: \n");
21             Console.Write("a = "); // ввод исходных данных
22             a = Convert.ToInt32(Console.ReadLine());
23             Console.Write("b = ");
24             b = Convert.ToInt32(Console.ReadLine());
25
26             P = a * b; // расчет произведения
27             // проверка условия
28             if (P < 0) // если P < 0, то
29             {
30                 P = P * (-2);
31                 Console.WriteLine("Произведение отрицательное, результат расчета " + P);
32             }
33             else if (P > 0) // иначе
34             {
35                 P = P * 3;
36                 Console.WriteLine("Произведение положительное, результат расчета " + P);
37             }
38             else Console.WriteLine("Произведение нулевое, результат расчета " + P);
39
40             Console.ReadKey();
41         }
42     }
43 }
```

Листинг 1 – Исходный код

Типовые примеры задач по теме «Решение задач на построение алгоритма простой и усложненной разветвляющейся структуры».

Вводятся 2 числа. Если их произведение отрицательно, умножить его на -2 и вывести результат, в противном случае увеличить его в $1,5$ раза и вывести результат (начертить блок-схему и определить типы используемых графических нотаций, написать программный код на основе блок-схемы).



```
1 using System;
2
3 namespace ConsoleApp1
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             int a, b;
10            double P = 1;
11            Console.Write("Введите a = ");
12            a = Convert.ToInt32(Console.Read());
13            Console.Write("Введите b = ");
14            b = Convert.ToInt32(Console.Read());
15
16            P = a * b;
17            if (P > 0) {P = -2 * P;}
18            else
19            { P = 1.5 * P};
20            Console.WriteLine("Результат: " + P.ToString());
21            Console.ReadKey();
22        }
23    }
24 }
```

Типовые задания для тестирования по темам «Основные структуры языка: циклы. Цикл с параметром. Цикл с условием».

Инструкция к тесту:

На выполнение теста дается 7 минут. Символ «*» в конце вопроса подразумевает наличие нескольких правильных ответов. Не задерживайтесь слишком долго над одним заданием. Ответ на задание состоит из номера задания и одной или нескольких букв – вариантов ответа.

Прежде чем приступить к решению, удостоверьтесь, что вы правильно поняли, что от вас требуется.

Вопросы:

1. До каких пор будут выполняться операторы в теле цикла while ($x < 100$)?

- а) пока x равен 100;
- б) пока x меньше или равен 100;
- в) пока x строго меньше 100;
- г) пока x больше 100.

2. Структура объявления переменных в C#

- а) [=] <идент. 2>;
- б) [:=] <идент. 2>;
- в) [=]; <идент. 2>
- г) [=] <идент. 2>;

3. Укажите правильную форму записи цикла do while

а)

```
1 // форма записи оператора цикла do while:
2 do // начало цикла do while
3 {
4 /*блок операторов*/;
5 }
6 while (/*условие выполнения цикла*/); // конец цикла do while
```

б)

```
1 // форма записи оператора цикла do while:
2 do // начало цикла do while
3 {
4 /*блок операторов*/;
5 }
6 while (/*условие выполнения цикла*/) // конец цикла do while
```

в)

```
1 // форма записи оператора цикла do while:
2 do // начало цикла do while
3 {
4 /*блок операторов*/;
5 }
6 while {/*условие выполнения цикла*/} // конец цикла do while
```

4. Тело любого цикла выполняется до тех пор, пока его условие ...

- а) у цикла нет условия;
- б) ложно;
- в) истинно

5. Какой оператор вызывает немедленный выход из цикла?

- а) break;
- б) continue;
- в) Stop;
- г) ;

6. Цикл с предусловием?

- а) while;
- б) for;
- в) do while;
- г) switch

7. Оператор цикла*:

- а) if;
- б) switch;
- в) while;

- г) for
8. Оператор повторений:
- а) break;
 б) switch;
 в) while
 г) for
9. Разрешается ли опускать фигурные скобки в цикле for?
- а) Да, это можно делать всегда.
 б) Нет, этого делать нельзя.
 в) Да, это можно делать в том случае, если известно какое количество операций будет использовано.
 г) Да, это можно делать, если должна выполняться лишь одна операция.
10. Цикл for (int i = 0; i < 10; i --); будет выполняться бесконечно?
- а) да;
 б) нет.

Ключи:

Вопрос	Ответ	Вопрос	Ответ
1	в	6	а
2	г	7	в, г
3	а	8	б
4	в	9	г
5	а	10	а

Темы для дискуссий

1. Эволюция и классификация языков программирования.
2. Основные понятия языков программирования
3. Этапы решения задач на компьютерах.
4. Трансляция, компиляция и интерпретация.
5. Алгоритм и его свойства.
6. Способы записи алгоритма.
7. Стандарт оформления блок-схем.
8. Алгоритмы разветвляющейся структуры.
9. Алгоритмы циклической структуры.
10. Интегрированные среды программирования.
11. Скалярные типы данных в программе.
12. Структурные типы данных в программе. Массивы.
13. Организация ввода-вывода в программе.
14. Условные операторы.
15. Циклические операторы.
16. Критерии стиля программирования.
17. Оценка стиля программирования.

МДК.01.02. Прикладное программирование

Типовые понятия для терминологического диктанта

Инструкция:

Цель – выявить сформированность знаний по базовым терминам курса.

Запишите продиктованные термины, дайте им определение. Время проведения диктанта – 15 минут.

Тестирование ПО (software testing) – это процесс анализа и эксплуатации программного обеспечения (плановая и упорядоченная деятельность) с целью выявления дефектов.

Дефект – это ошибка/неточность, которая может быть (а может и не быть) следствием сбоя, т.е. под дефектом понимается невыполнение требования, связанного с предполагаемым или установленным использованием.

Доказательство (proof) – попытка найти ошибки в программе безотносительно к внешней для программы среде.

Контроль (verification) – попытка найти ошибки, выполняя программу в тестовой или моделируемой среде.

Испытание (validation) – попытка найти ошибки, выполняя программу в заданной реальной среде.

Аттестация (certification) – авторитетное подтверждение правильности программы. При тестировании с целью аттестации выполняется сравнение с некоторым заранее определенным стандартом.

Отладка (debug, debugging) – процесс поиска, локализации и исправления ошибок в программе (IEEE Std.610-12.1990).

Детективность: тест должен с большой вероятностью обнаруживать возможные ошибки

Покрывающая способность: один тест должен выявлять как можно больше ошибок.

Воспроизводимость: ошибка должна выявляться независимо от изменяющихся условий (например, от временных соотношений).

Типовые вопросы для устного опроса по теме «Понятие жизненного цикла ПО. Стандарты описания ЖЦ. ISO 12207. Процессы ЖЦ»

1. Что такое жизненный цикл программного обеспечения?
2. Каким стандартом (ми) описывается ЖЦ?
3. Перечислите основные этапы ЖЦ программного обеспечения
4. Перечислите процессы ЖЦ программного обеспечения.
5. Для чего предназначаются основные процессы ЖЦ ПО? Какие процессы включены?
6. Для чего предназначаются вспомогательные процессы ЖЦ ПО? Какие процессы включены?
7. Для чего предназначаются организационные процессы ЖЦ ПО? Какие процессы включены?
8. Все ли процессы, указанные в стандарте, должны быть выполнены при каждой разработке ПО?

Типовая практическая работа по теме «Обработка числовых данных Excel в среде VBA»

Тема: Объектная модель Microsoft Excel.

Цели: Практическое знакомство с интегрированной средой разработки программных продуктов Visual Basic for Applications (VBA). Получение основных практических навыков разработки, отладки и тестирования подпрограмм-макросов в среде VBA в Microsoft Excel. Изучение методов программирования объекта Range.

Задание для практической работы:

Запрограммировать подпрограмму-макрос для обработки числовых значений элементов матрицы.

Порядок выполнения:

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

Указания к работе

Проект сохранить в папке своей подгруппы в файле с именем <Практические (Фамилия И.О.).xlsm>, в модуле <МодульМатрица>.

Краткое изложение темы

Visual Basic for Applications (VBA, Visual Basic для приложений) – немного упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office, а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDRAW, WordPerfect и ESRI ArcGIS.

К достоинствам языка можно отнести сравнительную лёгкость освоения, благодаря которой приложения могут создавать даже пользователи, не программирующие профессионально. К особенностям VBA можно отнести выполнение скрипта именно в среде офисных приложений.

Range – представляет ячейку, строку, столбец, диапазон ячеек содержащих один или более смежных блоков ячеек на рабочем листе.

Для проверки наличия содержимого в ячейке применяется функция IsEmpty(аргумент), которая возвращает логический результат.

Функция IsNumeric(аргумент) проверяет, что переданный аргумент имеет числовое значение.

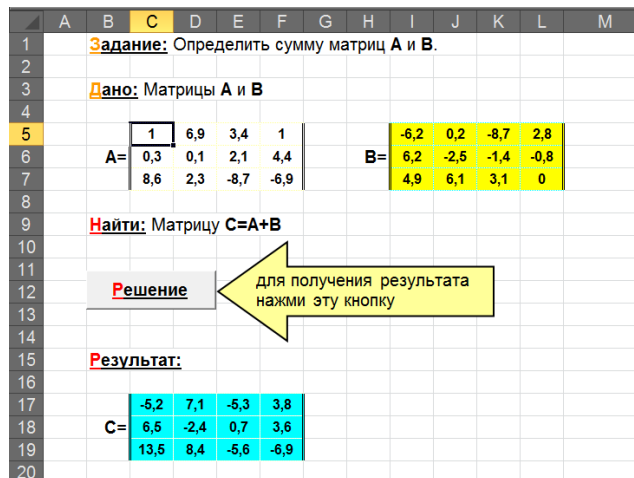
Пример выполнения задания

Найти сумму матриц A и B.

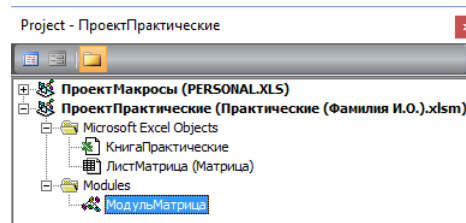
Решение (тестовый расчёт):

$$C = A + B = \begin{vmatrix} -1,5 & 6,9 & 7,7 & -1,8 \\ 0,3 & 0,1 & -5,3 & 4,4 \\ 8,6 & 2,3 & -8,7 & -6,9 \end{vmatrix} + \begin{vmatrix} -6,2 & 0,2 & -8,7 & 2,8 \\ 6,2 & -2,5 & -1,4 & -0,8 \\ 4,9 & 6,1 & 3,1 & 0 \end{vmatrix} = \begin{vmatrix} -7,7 & 7,1 & -1 & 1 \\ 6,5 & -2,4 & -6,7 & 3,6 \\ 13,5 & 8,4 & -5,6 & -6,9 \end{vmatrix}$$

Интерфейс программы:



Проект программы:



Программный код:

Option Explicit

```
Sub СуммироватьМатрицыАиВ()  
'Объявление переменных для работы с диапазонами  
Dim МатрицаА As Range, МатрицаВ As Range, МатрицаС As Range  
Dim Строка As Byte, Столбец As Byte  
  
'Установка ссылки на диапазон содержащий матрицу МатрицаС  
Set МатрицаС = [C17:F19]  
МатрицаС.ClearContents 'Очистка содержимого ячеек диапазона матрицы С  
  
Set МатрицаА = [C5:F7]  
'Проверка содержимого ячеек диапазона содержащего матрицу МатрицаА  
If Not СодержитЧислаНеПустые(МатрицаА) Then Exit Sub  
  
Set МатрицаВ = [I5:L7]  
If Not СодержитЧислаНеПустые(МатрицаВ) Then Exit Sub  
  
For Строка = 1 To МатрицаС.Rows.Count  
  For Столбец = 1 To МатрицаС.Columns.Count  
    МатрицаС.Cells(Строка, Столбец) = МатрицаА.Cells(Строка, Столбец) + _  
      МатрицаВ.Cells(Строка, Столбец)  
  Next Столбец  
Next Строка  
End Sub  
  
Private Function СодержитЧислаНеПустые(Диапазон As Range) As Boolean  
  Dim Ячейка As Range  
  
  For Each Ячейка In Диапазон.Cells  
    If Not IsNumeric(Ячейка) Then  
      MsgBox "Элемент не число.", vbCritical  
      Exit Function  
    ElseIf IsEmpty(Ячейка) Then  
      MsgBox "Элемент отсутствует.", vbCritical  
      Exit Function  
    End If  
  Next  
  СодержитЧислаНеПустые = True  
End Function
```

Типовые тестовые задания для зачета по теме «Разработка спецификаций для компонентов программного продукта»

Инструкция к тесту:

На выполнение теста дается 30 минут. Символ «*» в конце вопроса подразумевает наличие нескольких правильных ответов. Не задерживайтесь слишком долго над одним заданием. Ответ на задание состоит из номера задания и одной или нескольких букв – вариантов ответа.

Прежде чем приступить к решению, удостоверьтесь, что вы правильно поняли, что от вас требуется.

1. Легкость применения программного обеспечения это:
 - а) характеристики ПО, позволяющие минимизировать усилия пользователя по подготовке исходных данных, применению ПО;
 - б) отношение уровня услуг, предоставляемых ПО пользователю при заданных условиях, к объему используемых ресурсов;
 - в) характеристики ПО, позволяющие минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации.
2. Мобильность программного обеспечения это:
 - а) способность ПО выполнять набор функций, которые удовлетворяют потребности пользователей;
 - б) способность ПС безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени;
 - в) способность ПО быть перенесенным из одной среды (аппаратного / программного) в другое.
3. Укажите правильную последовательность этапов при каскадной модели жизненного цикла:
 - а) Определение требований → Тестирование → Реализация;

- б) Проектирование → Реализация → Тестирование;
- в) Проектирование → Определение требований → Реализация.
- 4. Устойчивость программного обеспечения – это:
 - а) свойство, характеризующее способность ПС завершать автоматически корректное функционирование ПК, несмотря на неправильные (ошибочные) входные данные;
 - б) свойство, характеризующее способность противостоять преднамеренным или непреднамеренным деструктивным действиям пользователя;
 - в) свойство, характеризующее способность ПС продолжать корректное функционирование, несмотря на неправильные (ошибочные) входные данные.
- 5. UML – это:
 - а) язык программирования, имеющий синтаксис схож с C#;
 - б) унифицированный язык визуального моделирования, использует нотацию диаграмм;
 - в) набор стандартов и спецификаций качества программного обеспечения.
- 6. При конструировании программного обеспечения на этапе разработки или выбора алгоритма решения реализуется следующее:
 - а) архитектурное для обработки программы;
 - б) выбор языка программирования;
 - в) совершенствование программы.
- 7. Проектирование ПО в основном рассматривается как
 - а) архитектурное проектирование;
 - б) коммуникационные методы;
 - в) детальные методы.
- 8. На этапе тестирования пользователь выполняет следующее:
 - а) синтаксическое отладки;
 - б) выбор тестов и метода тестирования;
 - в) определение формы выдачи результатов.
- 9. Что из приведенного не является одним из методов проектирования программного обеспечения?
 - А) структурное программирование;
 - б) объектно-ориентированное программирование;
 - в) алгебраическое программирования.
- 10. Как называется процесс разбиения одной сложной задачи на несколько простых подзадач?
 - а) абстракция;
 - б) декомпозиция;
 - в) реинжиниринг.
- 11. Что из приведенного является критериями оценки удобства интерфейсов?
 - А) скорость обучения;
 - б) адаптация к стилю работы пользователя;
 - в) все ответы правильные.
- 12. Интерфейс пользователя – это ...
 - а) набор методов взаимодействия компьютерной программы и пользователя этой программы;
 - б) набор методов для взаимодействия между программами;
 - в) способ взаимодействия между объектами.
- 13. Интерфейс – это ...
 - а) прежде всего, набор правил;
 - б) набор задач пользователя, которые он решает с помощью системы;
 - в) способ взаимодействия между объектами.
- 14. Техническое задание – это ...
 - а) документ объяснений для заказчика;
 - б) исходный документ для сдачи ПО в эксплуатацию;
 - в) выходной документ для проектирования, разработки автоматизированной системы.
- 15. Анализ требований – это ...

- а) отображение функций системы и ее ограничений в модели проблемы;
- б) показатель, который определяет необходимые усилия для диагностики случаев отказов;
- в) отображение частей программ, которые будут модифицироваться.

16. Архитектура программной системы – это ...

- а) декомпозиция решения для выделенного спектра задач домена на подсистемы или иерархию подсистем;
- б) определение системы в терминах вычислительных составляющих (подсистем) и интерфейсов между ними, которое отражает правила декомпозиции проблемы на составляющие;
- в) соответствующие вариации состава выделенных компонент.

17. Ассоциация – это ...

- а) возможность для некоторого класса находиться одновременно в связи с одним элементом из определенного множества классов;
- б) объединение нескольких понятий в более новое понятие, существенные признаки нового понятия об этом могут быть либо суммой компонент, либо существенно новыми (отношение «доля – целое»);
- в) самое общее отношение, утверждает наличие связи между понятиями, не уточняя зависимости их содержания и объемов.

18. Внешние метрики продукта:

- а) метрики надежности;
- б) метрики размера;
- в) метрики сложности.

19. Внутренние метрики продукта:

- а) метрики сопровождения;
- б) метрики годности;
- в) метрики стиля.

20. Процесс разработки ПО включает следующие процессы:

- а) сопровождения;
- б) проектирование;
- в) эксплуатация.

21. Последовательность работ по каскадной модели:

- а) требования, проектирование, реализация;
- б) проектирование, сопровождение, тестирование;
- в) требования, сопровождение, тестирование.

22. Проектирование – это ...

- а) преобразование требований в последовательность проектных решений по системе;
- б) определение главных структурных особенностей системы;
- в) определение подробностей функционирования и связей для всех компонент системы.

23. Модель жизненного цикла – это ...

- а) определение определенных действий, которые сопровождают изменения состояний объектов;
- б) типичная схема последовательности работ на этапах разработки программного продукта;
- в) отражение динамики изменений состояния каждого класса объектов.

24. Понятность – это ...

- а) атрибут функциональности, указывающий на возможность предотвращать несанкционированный доступ;
- б) атрибут надежности, который указывает на способность программы к перезапуску для повторного выполнения;
- в) атрибут удобства, определяющий усилия, необходимые для распознавания логических концепций и условий их применения.

Ключи:

Вопрос	Ответ	Вопрос	Ответ	Вопрос	Ответ
--------	-------	--------	-------	--------	-------

1	а	9	в	17	в
2	в	10	б	18	а
3	б	11	в	19	в
4	б	12	а	20	б
5	б	13	в	21	а
6	б	14	в	22	а
7	а	15	а	23	б
8	б	16	б	24	в

Задача 1. Напишите программу, которая вычисляет площадь треугольника по вводимым пользователем длинам его сторон.

Решение:

Описание алгоритма:

В ходе выполнения программы необходимо выполнить два действия, которые могут вызвать ошибку и требуют обработки исключительных ситуаций:

- обеспечить ввод сторон треугольника
- вычисление площади треугольника

Оба действия вызывают ошибку `ValueError`, ввиду чего объединить их в один блок `try` – `except` невозможно, поэтому второе действие следует выделить в отдельный блок и разместить его в разделе `else` первого блока.

Алгоритм:

1. Проверить наличие ошибок при вводе стороны треугольника;
2. Если произошло исключение `ValueError` – вывести сообщение о некорректности введенных данных, завершить выполнение программы;
3. Проверить возможность вычисления площади;
4. Если произошло исключение `ValueError` – вывести сообщение о том, что треугольник не существует, завершить выполнение программы;
5. Вывести площадь треугольника.

Код программы:

```
#!/usr/bin/env python3
```

```
# Выполнил: Иванов Иван Иванович
```

```
# Задача: Напишите программу, которая вычисляет площадь треугольника по вводимым пользователем длинам его сторон
```

```
import math
```

```
def main():
```

```
    try:
```

```
        a=float(input('Введите a: '))
```

```
        b=float(input('Введите b: '))
```

```
        c=float(input('Введите c: '))
```

```
    except ValueError:
```

```
        print('Введены нечисловые данные')
```

```
    else:
```

```
        try:
```

```
            p=(a+b+c)/2
```

```
            s=math.sqrt(p*(p-a)*(p-b)*(p-c))
```

```
        except ValueError:
```

```
            print('Треугольник не существует')
```

```
        else:
```

```
            print('Площадь треугольника по формуле Герона равна: S=',round(s,2))
```

```
    finally:
```

```
        print('Работа программы завершена')
```

```
if __name__ == "__main__":  
    main()
```

Задача 2. Дан текст в MS Word. Надо создать макросы, выполняющие задачи:

1. Удалить в тексте все слова, в которые напечатаны красным шрифтом.
2. Сосчитать количество слов, в которых имеется фуква “ф”.
3. Вставить перед каждым абзацем пустую строку.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ МОДУЛЯ.

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

Основными видами аудиторной работы обучающихся являются урок и практические занятия.

В ходе урока преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы. Во время занятий необходимо вести конспект. Преподаватель дает на уроке задания для закрепления пройденного материала, организует и оказывает обучающемуся помощь в самостоятельной работе во время урока, дает рекомендации на подготовку к практической работе и указания на выполнение домашней работы. Во время урока преподаватель также проводит проверку теоретических знаний по теме прошлого урока. Активное участие обучающегося во всех этапах занятия, позволит ему качественно усвоить необходимый теоретический и практический материал, разобраться в основных вопросах и получить дополнительные необходимые для понимания и дальнейшей практической деятельности рекомендации преподавателя.

Целями выполнения практических работ является:

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

Практические занятия вырабатывают у обучающихся навыки применения полученных знаний для решения профессиональных практических задач. На практических занятиях обучающиеся выполняют тренировочные упражнения, решают задачи, занимаются построением схем, трассировочных таблиц, и т. д.

По своему содержанию практические работы представляют собой практическое применение теоретической информации и тесно связаны с темой занятия. Обучающийся обязан выполнить весь перечень практических работ.

Для выполнения практических работ обучающимся выдается сборник практических работ или инструкция (методические рекомендации). Каждая инструкция содержит цель работы, ход выполнения работы, перечень заданий и задания для закрепления, которые выполняются обучающимся самостоятельно дома.

В процессе выполнения работы каждый обучающийся составляет отчет с учетом правил оформления. Небрежное оформление отчета, исправление уже написанного недопустимо.

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

Требования к оформлению отчетов к практическим работам

Отчеты к выполненным практическим работам должны соответствовать требованиям Единой системы конструкторской документации (ЕСКД) и Единой системы технической документации (ЕСТД).

Отчеты начинаются с титульного листа. Текстовые документы выполняются рукописным способом на писчей бумаге. Буквы и цифры необходимо писать четко, пастой или чернилами одного цвета (черной, синей, фиолетовой).

Все листы нумеруются сквозной нумерацией. Титульный лист входит в количество листов. На всех последующих листах нумерация проставляется в правом нижнем углу.

Отчет к практической работе разбивается на пункты, которые обозначаются арабскими цифрами. Пункты при необходимости могут быть разбиты на подпункты, которые нумеруются в пределах каждого пункта, например: 1.2, 1.3, 1.4.

Цифровые материалы, помещаемые в отчете, оформляются в виде таблиц. Над правым верхним углом таблицы должна быть надпись «Таблица» с указанием ее порядкового номера. Каждая практическая работа начинается с нового листа (страницы).

Типовая инструкция по охране труда для обучающихся

1. Будьте внимательны и дисциплинированы.
2. Не приступайте к выполнению работы без разрешения преподавателя.
3. Размещайте материалы на своем рабочем месте таким образом, чтобы исключить их падение.
4. Перед выполнением работы необходимо внимательно изучить ее содержание и ход выполнения.
5. Не оставляйте рабочего места без разрешения преподавателя.

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

В ходе изучения профессионального предусмотрена внеаудиторная (домашняя) самостоятельная работа в объеме 210 часов.

Внеаудиторная самостоятельная работа выполняется обучающимися в целях:

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

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

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

Оценка за выполнение домашнего задания выставляется в журнал учебных занятий.

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

Технологическая карта практических работ

УМК. ФК-ОС. Разработка алгоритмов различных структур.

№ занятия	Тема практической работы	Кол. часов	Задание	Литература со стр.
	МДК 01.01. Системное программирование	86		
	Тема 1.2. Высокоуровневые языки программирования.	58		
11	Составление программ линейной структуры.	2	<ul style="list-style-type: none"> – провести тестовый расчет и вычислить результат заданного выражения; – создать проект в среде программирования; – написать, отладить и запустить программу линейной структуры с исходными тестовыми данными. 	[14, стр. 1-3] [20, стр. 5-6]
13-14	Составление программ разветвляющейся структуры. Конструкция if – else.	4	<ul style="list-style-type: none"> – составить блок-схему решения задачи; – создать проект в среде программирования; – написать программу с использованием оператора if ... else, выполнить отладку; – подобрать наборы исходных данных для тестирования всех ветвей программы. 	[14, стр. 4-6] [20, стр. 20-27]
15	Составление программ разветвляющейся структуры. Конструкция switch – case.	2	<ul style="list-style-type: none"> – переработать программный код предыдущей практической работы с помощью конструкций switch ... case; – доработать программу алгоритма разветвляющейся структуры с помощью кода обработки ошибок времени выполнения; – подобрать наборы исходных данных для тестирования всех ветвей программы и механизма обработки ошибок; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 7-9] [20, стр. 28-30]
17	Составление программ циклической структуры: for.	2	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя цикл с параметром; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 10-12] [20, стр. 31-35]
19-20	Составление программ циклической структуры: while, do-while.	4	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя циклы с условиями; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 13-15] [20, стр. 36-40]
21	Составление программ усложненной структуры.	2	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя базовые конструкции: линейные, ветвление, цикл. 	УМК. ФКОС. Составление

				программ усложненной структуры.
25-26	Обработка одномерных массивов. Обработка ошибок.	4	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя массивы; – использовать механизм отлавливания и обработки ошибок в написанной программе; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 16-20] [20, стр. 54-62]
27	Обработка двумерных массивов. Обработка ошибок.	2	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя массивы; – использовать механизм отлавливания и обработки ошибок в написанной программе; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 21-23] [20, стр. 63-73]
29-30	Работа со строковыми переменными. Использование стандартных функций для работы со строками.	4	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя символьные данные; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 24-27] [20, стр. 74-79]
31	Разработка программ со структурированными типами данных.	2	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя массивы; – разработать программу алгоритма решения задачи, используя символьные данные. 	УМК. ФКОС. Разработка программ со структурированными типами данных.
35-36	Организация и использование функций.	4	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя функции; – включить в программу блок обработки ошибок; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 28-29] [20, стр. 48-52]
37	Разработка программ с применением рекурсивных функций.	2	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя рекурсивную функцию; – выполнить отладку и запуски программы с тестовыми наборами исходных данных. 	[14, стр. 30-32] [20, стр. 52-53]
38	Разработка программ с использованием функций.	2	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, написав пользовательскую функцию. 	УМК. ФКОС. Разработка программ с использованием функций.
40	Разработка программ с использованием указателей.	2	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя указатели; 	[14, стр. 33-36] [20, стр. 63-73]

			– выполнить отладку и запуск и программы с тестовыми наборами исходных данных.	
42-43	Разработка программ для обработки динамических массивов.	4	– разработать программу алгоритма решения задачи, используя динамические массивы; – *использовать механизм отлавливания и обработки ошибок в написанной программе; – выполнить отладку и запуски программы с тестовыми наборами исходных данных.	[14, стр. 37-38]
46-47	Разработка программ с использованием структур.	4	– разработать программу алгоритма решения задачи, используя структуры; – *использовать механизм отлавливания и обработки ошибок в написанной программе; – выполнить отладку и запуски программы с тестовыми наборами исходных данных.	[14, стр. 39-41] [20, стр. 87-90]
51	Разработка иерархической структуры классов: объявление класса, создание экземпляров класса.	2	– разработать программу алгоритма решения задачи, используя классы; – *использовать механизм отлавливания и обработки ошибок в написанной программе; – выполнить отладку и запуски программы с тестовыми наборами исходных данных.	[14, стр. 42-44]
52	Разработка иерархической структуры классов: создание наследованного класса.	2	– разработать программу алгоритма решения задачи, используя классы; – *использовать механизм отлавливания и обработки ошибок в написанной программе; – выполнить отладку и запуски программы с тестовыми наборами исходных данных.	[14, стр. 45-48]
53	Разработка иерархической структуры классов: перегрузка методов класса.	2	– доработать программу с использованием классов из практической работы № 15, внедрив перегружаемые методы; – *использовать механизм отлавливания и обработки ошибок в написанной программе; – выполнить отладку и запуски программы с тестовыми наборами исходных данных.	[14, стр. 49-52]
56	Выполнение операций с объектами файловой системы. Обработка ошибок.	2	– разработать программу алгоритма решения задачи, используя библиотеки работы с файлами; – использовать механизм отлавливания и обработки ошибок в написанной программе; – *оформить запись и чтение из файла в виде подпрограмм; – выполнить отладку и запуск программы с тестовыми наборами исходных данных.	[14, стр. 53-55] [20, стр. 91-92]

57-58	Выполнение операций с объектами файловой системы. Запись структур.	4	<ul style="list-style-type: none"> – разработать программу алгоритма решения задачи, используя структуры и библиотеки работы с файлами; – использовать механизм отлавливания и обработки ошибок в написанной программе; – * оформить заполнение структуры и вывод полей структуры на экран в виде подпрограмм; – * оформить запись и чтение из файла в виде подпрограмм; – выполнить отладку и запуск программы с тестовыми наборами исходных данных. 	[14, стр. 56-59] [20, стр. 92-94]
	Тема 1.3. Технологии объектно-ориентированного программирования	22		
64	Каркас Windows – приложения. Программирование элементов управления.	2	<ul style="list-style-type: none"> – разработать оконный графический интерфейс приложения; – использовать механизм отлавливания и обработки ошибок; – выполнить отладку и запуск программы с тестовыми наборами исходных данных. 	[14, стр. 60-63]
65-66	Windows – приложение. Программирование элементов управления. Работа с массивами.	4	<ul style="list-style-type: none"> – разработать пользовательский графический интерфейс для обработки одномерного массива; – использовать механизм отлавливания и обработки ошибок; – выполнить отладку и запуск программы с тестовыми наборами исходных данных. 	[14, стр. 64-68]
67-68	Windows – приложение. Программирование элементов управления. Работа с функциями.	4	<ul style="list-style-type: none"> – разработать пользовательский графический интерфейс с интеграцией новых объектов, содержащих в себе объявление и описание пользовательских функций; – использовать механизм отлавливания и обработки ошибок (ErrorProvider); – выполнить отладку и запуск программы с тестовыми наборами исходных данных. 	[14, стр. 69-73]
73-75	Windows – приложение. Сбор сведений о системе.	4	<ul style="list-style-type: none"> – разработать пользовательский графический интерфейс для приложения «Сведения о системе»; – для получения информации воспользоваться функциями Windows API; – выполнить отладку и запуск программы с тестовыми наборами исходных данных. 	[14, стр. 74-77]
77-79	Windows – приложение. Мониторинг процессов.	4	<ul style="list-style-type: none"> – разработать пользовательский графический интерфейс для приложения «Мониторинг процессов»; – для получения информации воспользоваться функциями Windows API; – выполнить отладку и запуск программы с тестовыми наборами 	[14, стр. 78-81]

			исходных данных.	
	Тема 1.4. Оформление технической документации	6		
85-86	Разработка технического задания.	4	– подготовить техническое задание на разработку программы; – для получения информации воспользоваться ГОСТ 19.201-78 и 34.602-89.	[14, стр. 86-88]
87	Зачетное занятие по практическим работам.	2	тестирование по всем разделам МДК.	УМК
	МДК 01.02. Прикладное программирование	154		
	Тема 2.1. Разработка спецификаций для компонентов программного продукта.	16		
12	Анализ проблемы. Постановка задачи.	2	– составить описание системы; – на основании описания провести анализ осуществимости (составление списка заинтересованных лиц, анкетирование и проведение интервью, заключение аналитика); – распределить роли в группе (руководитель проекта-разработчик, системный аналитик-разработчик, тестер-разработчик); – указать возможные риски; – спланировать длительность проекта (указать этапы и время их выполнения); – построить блок-схему решения задачи.	[15, стр. 1-3]
13-14	Разработка технического задания к программному продукту.	4	– разработать техническое задание на программный продукт в соответствии с ГОСТ 19.201–78 и ГОСТ 34.602–89; – оформить работу в соответствии с ГОСТ 19.106–78.	[15, стр. 4-11]
17-18	Разработка спецификаций при структурном подходе.	4	на основе технического задания из практической работы № 2 выполнить анализ функциональных и эксплуатационных требований к программному продукту; определить основные технические решения (выбор языка программирования, структура программного продукта, состав функций ПП, режимы функционирования и т.д.); разработать архитектуру проектируемого ПП. Для проектирования архитектуры системы использовать диаграммы потоков данных (DFD), блок-схемы, flow-диаграммы, диаграммы Насси-Шнейдермана и т.д.; определить объекты и их атрибуты для моделирования структур данных на основе построенной модели архитектуры системы;	[15, стр. 12-16]

			добавить словарь терминов (данных).	
22-23	Разработка спецификаций ПО при объектном подходе (диаграммы прецедентов, классов).	4	<ul style="list-style-type: none"> – определить актеров, выбрать прецеденты, разработать диаграмму прецедентов, используя нотации языка UML. определить типы связей, указать для них категории (где это возможно); – составить спецификацию прецедента на основании диаграммы; – разработать диаграмму классов, используя нотации языка UML. 	[15, стр. 17-23]
25	Зачет	2	Тестирование по теме 2.1. Разработка спецификаций для компонентов программного продукта.	УМК
	Тема 2.2. Разработка кода программного продукта на уровне модуля.	100		
5-8	Объектная модель Microsoft Excel	8	Запрограммировать подпрограмму-макрос для обработки числовых значений элементов матрицы.	[5, стр. 3]
9-13	Объектная модель Microsoft Excel	10	Запрограммировать класс со свойствами и методами для обработки ячеек диапазона, содержащихся произвольные типы значений. Записать макрос, использующий экземпляр объекта класса.	
16-18	Объектная модель взаимодействия Microsoft.Office.Interop.Excel.	6	Модифицировать в рамках управляемого кода проект программы для обработки числовых значений элементов матрицы (диапазона).	[5, стр. 7]
19-21	Объектная модель взаимодействия Microsoft.Office.Interop.Excel.	6	Модифицировать в рамках управляемого кода проект программы для обработки нечисловых значений элементов диапазона.	
27-31	Объектная модель Microsoft Word.	10	Запрограммировать класс с методами для обработки текстовых данных документа. Записать макрос, использующий экземпляр объекта класса.	[5, стр. 10]
32-34	Объектная модель взаимодействия Microsoft.Office.Interop.Word.	6	Модифицировать в рамках управляемого кода проект программы для обработки текстовых данных документа. Расширить функциональные возможности программы путем добавления не менее трех параметров обработки.	[5, стр. 15]
42-47	Программирование объектов модели DAO.	12	Запрограммировать класс с методами для формирования структуры базы данных. Записать макрос, использующий экземпляр объекта класса.	[5, стр. 18]
48-51	Разработка форм. Программирование событий формы и элементов управления.	8	Разработать формы пользовательского интерфейса для многофункционального приложения. Запрограммировать события форм и элементов управления.	[5, стр. 20]
52-56	Взаимодействие объектных моделей приложений Access, Excel и объектной модели доступа к данным.	10	Разработать модуль класса с методами формирования отчета в формате Excel в среде выполнения VBA Access.	[5, стр. 23]
57-61	Взаимодействие объектных моделей приложений Access, Word и объектной модели доступа к данным.	10	Разработать модуль класса с методами формирования отчета в формате Word в среде выполнения VBA Access.	[5, стр. 25]
64-70	Надстройки на уровне приложений	14	Разработать модуль класса с методами для реализации функций надстройки	[5, стр. 27]

	пакета Microsoft Office.		приложений пакета Microsoft Office.	
	Тема 2.3. Отладка и тестирование программного продукта на уровне модулей	26		
4-5	Разработка, отладка и тестирование программы с помощью средств среды разработки.	4	Выполнить отладку разрабатываемых функций ПП встроенными средствами среды Visual Studio, используя точки останова, пошаговый запуск и добавление в окно контроля переменных.	[17, стр. 1-3]
11-12	Разработка тестовых сценариев (Test case), bug report.	4	<ul style="list-style-type: none"> – построить тестовые случаи (положительные, отрицательные тесты) для проверки ПП методом «Черного ящика»; – создать отчет об ошибке (bug report). 	[17, стр. 4-7]
14	Применение техник создания тестов для черного ящика.	2	<ul style="list-style-type: none"> – разработать тест, используя технику эквивалентного разбиения и граничных условий. 	[17, стр. 8-10]
17-18	Построение управляющего графа программы.	4	<ul style="list-style-type: none"> – построить управляющий граф программы (УГП); – выделить компоненты УГП; – определить трассы для дальнейшего составления тестовых наборов. 	[17, стр. 11-14]
21-23	Модульное тестирование.	6	<ul style="list-style-type: none"> – разработать unit-тест для алгоритма в соответствии с вариантом задания; – реализовать алгоритм в соответствии с номером варианта на языке C#. 	[17, стр. 15-18]
25-26	Разработка системы тестов.	4	<ul style="list-style-type: none"> – разработать набор тестов для алгоритма в соответствии с вариантом задания; – реализовать алгоритм в соответствии с номером варианта на языке C#; – обеспечить максимально возможное покрытие кода тестами. 	[17, стр. 19-21]
31	Зачет	2	Тестирование по теме 2.3. Отладка и тестирование программного продукта на уровне модулей.	УМК
	Тема 2.4. Документирование.	12		
3-4	Разработка сопроводительной документации. Описание программы.	4	Составить описание программы в соответствии с ГОСТ 19.402-78 ЕСПД. Описание программы. Требования к содержанию и оформлению.	[18, стр. 1-3]
6-7	Разработка сопроводительной документации. Руководство пользователя.	4	В соответствии с ГОСТ 19.505-79 ЕСПД. «Руководство оператора. Требования к содержанию и оформлению» и РД 50-34.698-90 в п.п. 3.4. «Руководство пользователя» разработать руководство пользователя для разработанного ПП.	[18, стр. 4-6]
9	Разработка сопроводительной документации. Руководство программиста.	2	В соответствии с ГОСТ 19.504-79 ЕСПД. «Руководство программиста. Требования к содержанию и оформлению» разработать руководство программиста для разработанного ПП.	[18, стр. 7-9]
10	Зачет	2	Тестирование по теме 2.4. Документирование.	УМК

Задания для внеаудиторной (домашней) самостоятельной работы

Номер, наименование разделов, тем	Вид внеаудиторной самостоятельной работы	Задания для внеаудиторной самостоятельной работы	Примерный объем времени на выполнение, в час.
Раздел 1. ПМ.01. Разработка программных модулей для управления компонентами компьютерной системы МДК.01.01. Системное программирование			
Тема 1.1. Общие сведения о системном программном обеспечении.			
1.1.1 Системное программное обеспечение.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
1.1.2 Классификация системных программ. 1.1.3 Интерфейсы операционной системы.	Проработка конспекта лекций, работа с учебником.	Используя конспект и дополнительную литературу (см. список литературы) дать развернутый ответ на вопросы: <ol style="list-style-type: none"> 1. Что такое программа? 2. Что такое программное обеспечение? 3. Перечислите классы программных продуктов. 4. Дайте определение системного программного обеспечения. 5. Классифицируйте системное программное обеспечение. 6. Задачи системного программного обеспечения? 7. Дайте определение прикладного программного обеспечения. Перечислите его составляющие. 8. Дайте определение инструментального программного обеспечения. Перечислите его составляющие. 9. Что такое управляющая программа? 10. Что такое обрабатывающая системная программа? 11. Что такое базовое программное обеспечение? 12. Что такое сервисное программное обеспечение? 13. Что такое интерфейс операционной системы? 14. Поясните термин Интерфейс системных вызовов. 15. Поясните термин API (Интерфейс прикладного программирования). 16. Поясните назначения интерфейса POSIX. 	1,5
	Подготовка сообщений по	Используя Интернет-ресурсы, подготовить сообщение по	1

	теме	теме «Интерфейсы»: 1. Понятие интерфейса. Согласованный интерфейс. Виды интерфейсов. 2. Аппаратный интерфейс. Понятие. Последовательные и параллельные аппаратные интерфейсы. 3. Пользовательский интерфейс. Командный интерфейс. 4. Пользовательский интерфейс. Графический интерфейс. 5. Пользовательский интерфейс. Речевой интерфейс. 6. Интерфейс операционной системы: системные вызовы. 7. Интерфейс операционной системы: API. 8. Интерфейс операционной системы: POSIX.	
Тема 1.2. Высокоуровневые языки программирования.			
1.2.1 Методы программирования: структурный, модульный, объектно-ориентированный. Достоинства и недостатки методов программирования. 1.2.2 Понятие о программном модуле. Основные понятия ООП: объект, класс, экземпляр класса. 1.2.3 Высокоуровневые языки программирования. Структура простейшей программы. 1.2.4 Средства разработки приложений: Visual Studio.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций, работа с учебником.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Что такое метод программирования? 2. Что такое Методология программирования? 3. В чем особенность структурного программирования? 4. В чем особенность модульного программирования? 5. В чем особенность объектно-ориентированного программирования? 6. Перечислите основные принципы ООП. 7. Что такое инкапсуляция? 8. Что такое наследование? 9. Что такое полиморфизм? 10. Что такое программный модуль? 11. Что характерно для модуля? 12. Что такое объект? Чем он характеризуется? 13. Что такое класс? Приведите примеры. 14. Что такое язык программирования? 15. В чем разница между низкоуровневыми и	2

		<p>высокоуровневыми ЯП?</p> <p>16. Опишите структуру простейшей программы на C#.</p> <p>17. Что такое генерация программы?</p> <p>18. Перечислите этапы создания программ на C#.</p>	
	Поиск материала в интернете.	<p>Воспользовавшись Интернет-ресурсами (см. список литературы) найдите подробное описание команд, входящих в состав простейшей программы на C#.</p> <p>Ознакомьтесь лично с интерфейсом среды разработки приложений Visual Studio.</p>	1
1.2.5 Основные структуры языка: переменные, константы и типы данных. Объявление. Оператор присвоения. Пользовательские типы данных.	Составление глоссария.	<p>Найти в конспекте все основные определения по теме и выделить их.</p>	0,5
1.2.6 Основные структуры языка: потоки ввода-вывода.	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <ol style="list-style-type: none"> 1. Что такое переменная? 2. Что такое константа? 3. По каким правилам задаются идентификаторы? Приведите примеры верных и ошибочных идентификаторов. 4. Какие значения могут принимать логические переменные? 5. Что такое тип данных? 6. Какие типы данных используются в языке C#? 7. Зачем необходимо именовать данные и декларировать (объявлять) их? 8. Для чего нужен оператор присвоения? Как он используется? 9. Какие операнды могут присутствовать в символьных выражениях? Какие операции можно использовать? 10. Что такое пользовательский тип данных? 11. Что такое базовый тип данных? 12. С помощью каких операторов реализуются в программе ввод данных с клавиатуры и вывод на экран? Опишите их синтаксис. 13. Каким способом можно управлять форматом ввода данных? Приведите примеры для разных типов данных. 	2

		<p>14. Как заканчивается любая команда в программе?</p> <p>15. Зачем нужны служебные слова (директивы)?</p> <p>16. Где может записываться комментарий и в чем состоит правило его записи?</p>	
	Подготовка к выполнению практических работ.	Используя основную и дополнительную литературу (см. список литературы), воспроизвести все базовые формы записей алгоритмов. Уточнить особенности применения линейной.	1
	Решение задачи на построение алгоритма линейной структуры.	<p>Построить блок-схему для линейного алгоритма:</p> <p>А) $Z = \left(\frac{x+1}{x-1}\right)^x + 18xy^2$</p> <p>Б) $Y = \frac{1 + \sin \sqrt{x+1}}{\cos(12 \cdot z - 4)}$</p>	0,5
1.2.7 Основные структуры языка: алгоритмы ветвления, безусловный и условный переходы.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <p>1. Приведите примеры общего правила декларации переменных числового и символьного типов.</p> <p>2. Нарисуйте синтаксическую диаграмму оператора ветвления и сформулируйте правила его выполнения.</p> <p>3. Приведите общую форму записи усеченного оператора условного перехода.</p> <p>4. Нарисуйте синтаксическую диаграмму оператора выбора и сформулируйте правила его выполнения.</p> <p>5. Запишите оператор безусловного перехода и объясните, как он действует.</p>	1,5
	Решение задач на построение алгоритма простой и усложненной разветвляющейся структуры.	<p>Построить блок-схему для алгоритма с ветвлением и написать программу:</p> <p>А) Сравнить два числа между собой и вывести результат.</p> <p>Б) Определить правильность введенной даты (число от 1 до 31, месяц – от 1 до 12). Если введены не корректные данные, то сообщить об этом.</p> <p>В) Вычислить значение функции:</p>	2

		$F(x) = \begin{cases} -x^2 - 3x + 9 & \text{при } x \geq 3 \\ \frac{x}{x^3 - 6} & \text{при } x < 3 \end{cases} .$	
1.2.8 Основные структуры языка: циклы. Понятие цикла. Цикл с параметром.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
1.2.9 Основные структуры языка: циклы. Цикл с условием.	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Что такое цикл? 2. Какие средства предоставляются языком C# для организации циклов? 3. Что такое цикл с параметром? 4. Приведите общие формы записи арифметических операторов цикла и объясните, как они действуют. 5. Что такое итерационный цикл? 6. Представьте общую форму записи оператора цикла итерационного типа с предусловием и объясните, как он действует. 7. Представьте общую форму записи оператора цикла итерационного типа с постусловием и объясните, как он действует. 8. Когда применяются операторы Break и Continue? Можно ли выйти из цикла, не завершая его? 9. Как нужно организовать цикл, чтобы можно было войти в тело цикла?	1,5
	Решение задачи на построение алгоритма простой и усложненной циклической структуры.	Решить задачу (построить блок-схему, написать программу), используя три варианта операторов цикла: цикл «Пока», цикл «До» и цикл с параметром: А) Вычислить значение функции на отрезке [1;4] с шагом 1: $Y = 2x - \sin x$ Б) Создать простые числа на основе формулы $2x^2 + 29$ при $0 \leq x \leq 28$.	2
1.2.10 Основные структуры языка: массивы. Объявление, обращение к элементам. Заполнение массива.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:	1,5

<p>1.2.11 Основные структуры языка: массивы. Ввод и вывод массивов. Обработка массивов.</p> <p>1.2.12 Основные структуры языка: массивы. Обработка ошибок.</p>		<ol style="list-style-type: none"> 1. Что такое массив? 2. Что такое размер массива и размерность массива? 3. Как объявляются одномерные и двумерные массивы? 4. Можно ли в качестве верхних и нижних границ массивов употреблять не числа, а символы? 5. Как идентифицируются элементы массива? Приведите примеры записи элементов одномерного массива и матрицы. 6. В какой последовательности в памяти компьютера располагаются элементы многомерного массива? 7. Какие действия можно выполнять над массивом в целом (не поэлементно)? 8. Приведите фрагмент кода построчного вывода матрицы (двумерного массива) на экран компьютера. 9. Какие ошибки при работе с массивами могут возникнуть? 10. Что такое исключение? 11. Как возбуждаются исключения? 12. Каким образом обрабатываются исключения? 	
	<p>Решение задач на построение алгоритма обработки одномерных и двумерных массивов.</p>	<p>Построить блок-схему решения задачи и написать программу:</p> <p>А) Дан вектор $\{z_i\}, i = 1, \dots, 50$. Вычислить его длину $L = \sqrt{z_1^2 + z_2^2 + \dots + z_{50}^2}$</p> <p>Б) В массиве расположены случайные числа. Подсчитать количество чисел, значения которых лежат в интервале $[0; 1]$.</p> <p>В) Объявлен одномерный массив из 100 элементов, заполненный случайными двузначными числами (от 10 до 99). Вывести на экран в обратном порядке.</p>	<p>2</p>
<p>1.2.13 Основные структуры языка: символы и строки.</p>	<p>Составление глоссария.</p>	<p>Найти в конспекте все основные определения по теме и выделить их.</p>	<p>0,5</p>
	<p>Проработка конспекта лекций.</p>	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <ol style="list-style-type: none"> 1. Что такое символьная строка? Чем она отличается от 	<p>1,5</p>

		<p>массива символов?</p> <p>2. Что означает операция конкатенация строк?</p> <p>3. Перечислите функции обработки символьных и строковых данных. Приведите примеры.</p>	
	Изучение стандартных функций для работы со строками.	<ul style="list-style-type: none"> ▪ В консольном проекте написать программу: ▪ 1. для проверки стандартных функций работы со строками (библиотека <code>cstring</code>): <code>strlen()</code>; <code>strcat()</code>; <code>strcpy()</code>; <code>strcmp()</code>. <p>2. Объявить строки, массив символов. Записать в строку и массив слово и строку, введенные с клавиатуры (<code>gets()</code>, <code>cin</code>). Вывести результат на экран.</p>	2
	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
<p>1.2.14 Основные структуры языка: организация подпрограмм. Функции.</p> <p>1.2.15 Основные структуры языка: пользовательские функции. Рекурсии.</p>	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <ol style="list-style-type: none"> 1. Какие подпрограммы можно составлять на C#? 2. Что такое пользовательская функция? 3. Из каких разделов состоит подпрограмма-функция? 4. Что такое формальные и фактические параметры подпрограммы? 5. В чем заключается разница между параметрами – переменными и параметрами-значениями в их описании и использовании при обращении к подпрограмме? Приведите пример. 6. Какие правила необходимо соблюдать при вызове функции и указании ее фактических параметров? 7. Могут ли в функции формальные параметры одновременно служить параметрами входа и выхода? 8. Существуют ли функции без формальных параметров? 9. Может ли одна функция вызвать другую функцию? 10. Существуют ли какие-либо ограничения на количество пользовательских функций? 11. Что такое рекурсия? 12. Как определяются рекурсивные подпрограммы? 13. Что такое спуск и подъем рекурсии? 	1,5

	Решение задач по теме.	<p>Построить блок-схему решения задачи и написать программу:</p> <p>А) Вычислить площадь кольца по значениям внутреннего и внешнего радиусов (функция).</p> <p>Б) По координатам вершин треугольника вычислить его периметр, используя подпрограмму вычисления длины отрезка между двумя точками (функция).</p> <p>В) Найти факториал числа n (рекурсия).</p> <p>Г) Вычислить сумму элементов одномерного массива размерностью 100 (рекурсия).</p>	2
1.2.16 Основные структуры языка: динамические структуры данных. Указатели. Объявление, использование.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <ol style="list-style-type: none"> 1. Что такое указатель? Как он идентифицируется в программе? Приведите примеры. 2. Зачем используются переменные – указатели? Какие данные могут хранить эти переменные? 3. Поясните назначение унарного оператора *. 4. Поясните назначение унарного оператора &. Как он связан с указателем? 5. Какие арифметические операции над указателями допускаются в языке C#? 6. Для каких типов данных может быть использован указатель? 7. Как осуществляется инициализация указателей на вещественные типы данных? 8. Как осуществляется инициализация указателей на символьные типы данных? 9. Какая операция используется для разыменования указателя? 10. Какой смысл имеет значение указателя NULL? 11. Как следует определять и инициализировать указатель на константу? 12. Как следует определять и инициализировать 	1,5

		<p>константный указатель?</p> <p>13. Какое отличие константного указателя от указателя на константу?</p>	
	Изучение особенностей использования указателей.	<p>В консольном проекте написать программу для изучения особенностей использования указателей:</p> <ol style="list-style-type: none"> 1. Создать и инициализировать целочисленную переменную. 2. Объявить указатель. 3. Вывести значение переменной и то, на которое ссылается указатель. 4. Выполнить операцию разыменования указателя. 5. Вывести адрес переменной, на которую ссылается указатель. 6. Выполнить все арифметические операции с указателем и вывести результат на экран. 	2
1.2.17 Основные структуры языка: массивы с динамическим выделением памяти.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <ol style="list-style-type: none"> 1. Что такое массив с динамическим выделением памяти? 2. В чем заключается разница между статическим и динамическим выделением памяти? 3. Какие конструкции в языке C# можно использовать для выделения памяти под массив? 4. Какое правило необходимо соблюдать при работе с динамическими массивами? 5. Как можно освободить место в памяти, ранее выделенное под динамический массив? 	1,5
	Изучение способов обработки динамических массивов.	<p>Построить блок-схему и написать программу:</p> <ol style="list-style-type: none"> 1. Объявить одномерный и двумерный массивы, размерность которых вводится пользователем с клавиатуры (т.е. заранее неизвестна). 2. Заполнить массивы случайными числами в интервале [0, 9]. 	2

		3. Освободить память после вывода значений на экран.	
1.2.18 Основные структуры языка: структуры. Объявление, использование.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Что такое комбинированный тип данных и как он описывается в программе? 2. Как идентифицируются элементы структуры? Приведите примеры. 3. Что такое переменная типа структура? 4. Как выполняется инициализация структуры? 5. Поясните, как структура представлена в памяти? 6. Можно ли организовать вложенные структуры? 7. Поясните термин «массив структур». Каким образом он организуется? 8. Каким образом структура используется в программе?	1,5
	Решение задач на использование структур.	Создать и заполнить структуру: А) Химический элемент из таблицы Д.И.Менделеева. Б) Кондитерское изделие. В) Учетная запись студента.	2
1.2.19 Основные структуры языка: классы. Объявление, использование. Абстрактные классы. 1.2.20 Основные структуры языка: классы. Пользовательские классы.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Поясните основные концепции ООП. 2. Что такое класс? 3. Что такое экземпляр класса? 4. Что такое прототип? 5. В чем отличие класса от структуры? 6. Каким образом класс объявляется? 7. Что такое спецификатор доступа? Перечислите их,	2

		<p>указав особенности применения.</p> <ol style="list-style-type: none"> 8. Что такое поле класса? 9. Что такое метод класса? 10. Зачем нужны закрытые поля и методы? 11. Каким образом можно получить доступ к членам класса? 12. Что такое конструктор, деструктор класса? 13. Когда вызывается деструктор? 14. Каким образом в классах реализуется наследование? 15. Что такое абстрактный класс? 16. Что такое пользовательский класс? 17. Каким образом реализуется полиморфизм в классах? 18. Что такое перегрузка методов? 19. Что означает переопределение метода базового класса? 20. Может ли класс содержать в качестве полей данных экземпляры других классов? 	
	Решение задач на создание классов.	<p>Объявить класс, описать его с указанием методов и полей:</p> <p>А) «Телефонный звонок» - номер телефона, дата разговора, продолжительность, код города.</p> <p>Б) «Кинотеатр» - название кинофильма, сеанс, стоимость билета, количество зрителей.</p>	1,5
	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
<p>1.2.21 Основные структуры языка: файлы. Понятие и организация, операции с файлами.</p> <p>1.2.22 Основные структуры языка: файлы. Стандартные функции для работы с файлами.</p>	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <ol style="list-style-type: none"> 1. Что такое файл? 2. Как назначить переменной файл? В чем состоит сущность назначения в физическом смысле? 3. С помощью каких команд производится открытие файла для чтения и записи? 4. Что значит открыть файл для записи, для чтения? 5. Что происходит при выполнении закрытия файла? 6. Зачем закрывать файл после записи-чтения? 7. В чем состоит различие в использовании файлов 	1,5

		последовательного и прямого доступа? 8. Что такое манипулятор? Приведите примеры.	
	Решение задач на построение алгоритма работы с объектами файловой системы.	Построить блок-схему решения задачи и написать программу: А) Записать в файл несколько значений среднесуточной температуры. Значения генерируются случайным образом. Б) Дан файл вещественных чисел. Подсчитать количество нулевых значений в нем и вывести результат.	2
Тема 1.3. Технологии объектно-ориентированного программирования.			
	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
1.3.1 Общие принципы и этапы разработки программного обеспечения. Жизненный цикл программного обеспечения. Типы приложений. 1.3.2 Визуальное событийно - управляемое программирование. Интерфейс Windows -приложения. 1.3.3 Каркас Windows - приложения.	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Что такое программное обеспечение? 2. Перечислите общие принципы разработки ПО. 3. Что такое ЖЦ ПО? 4. Какими документами регламентируется ЖЦ ПО? 5. Какие типы приложений вам известны? 6. Поясните термин визуальное программирование. В чем его особенности? 7. Что такое событийно-управляемое программирование? 8. Что такое оконное приложение? 9. Перечислите наиболее часто используемые типы данных Windows. 10. Опишите структуру простейшей программы на языке C#. 11. Что такое директива предпроцессора условной компиляции? Директивы #include, #define, #pragma. 12. Опишите структуру оконного приложения (каркас), написанного на языке C#. 13. Поясните назначение функции WinMain(). 14. Для чего создается цикл обработки сообщений? 15. Для чего нужна оконная функция WndProc?	1,5
	Сообщения по теме	Используя Интернет-ресурсы, подготовить сообщение по	2

	«Визуальное событийно - управляемое программирование».	теме «Визуальное событийно - управляемое программирование»: 1. Визуальное программирование: понятие, языки, средства разработки. 2. Событийно-управляемое программирование. 3. Визуальное программирование: Google Blockly, Scratch, AppInventor и т.д.	
1.3.4 API. Программирование с использованием Windows API. Программирование с использованием Windows API. Сбор сведений о системе.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Что такое API? 2. Что такое Windows API? 3. Для чего подключается файл «Windows.h»? 4. Что такое snapshot? Для чего он нужен? Как он создается? 5. Какая структура содержит информацию о текущей компьютерной системе? 6. Какая функция Windows API позволяет получить сведения о центральном процессоре? С какой структурой взаимодействует эта функция? Что требуется для ее работы? 7. Какая функция Windows API позволяет получить сведения о версии операционной системы, работающей в данный момент? С какой структурой взаимодействует эта функция? Что требуется для ее работы? 8. Какая функция Windows API позволяет получить сведения о памяти? С какой структурой взаимодействует эта функция? Что требуется для ее работы?	1,5
	Изучение функций API для сбора сведений о системе.	1. Создать проект Windows Forms. 2. Создать слепок системы (snapshot). 3. Найти структуру, содержащую информацию о текущей компьютерной системе. 4. Определить назначение всех параметров структуры.	2
1.3.5 Программирование с	Составление глоссария.	Найти в конспекте все основные определения по теме и	0,5

использованием Windows API. Мониторинг процессов.		выделить их.	
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Какая структура содержит информацию о процессах в системе? 2. Что такое PID? 3. Какие функции Windows API позволяют перебрать список всех процессов, доступных в системе? Подключение какой библиотеки требуется для их работы? 4. Какая функция позволяет уничтожить процесс?	1,5
	Изучение функций API для мониторинга процессов.	1. Создать проект Windows Forms. 2. Создать слепок системы (snapshot). 3. Найти структуру, содержащую информацию о процессах в системе. 4. Определить назначение всех параметров структуры.	2
1.3.6 Программирование с использованием Windows API. Обработка сообщений от периферийных устройств.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Посредством чего обеспечивается взаимодействие приложений между собой? 2. Опишите механизм обмена сообщениями между приложениями. 3. Каким образом можно идентифицировать сообщение? 4. Дайте краткую характеристику сообщений, поступающих о мыши. Приведите примеры. 5. Какие функции Windows API используются для работы с мышью? 6. Что такое фокус ввода? 7. Дайте краткую характеристику сообщений, генерируемых драйвером клавиатуры. Приведите примеры. 8. Какие функции Windows API используются для работы с клавиатурой?	1,5
1.3.7 Программирование с	Составление глоссария.	Найти в конспекте все основные определения по теме и	0,5

использованием Windows API. Мониторинг окон.	Проработка конспекта лекций.	выделить их. Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Что такое окно в прикладной программе? 2. Что является окнами в ОС Windows? 3. Перечислите функции API, позволяющие выполнить мониторинг окон. 4. Что такое дескриптор (handle) окна? 5. Какие сообщения ОС Windows посылает любому окну? 6. Какая библиотека содержит функции работы с окнами?	1
	Изучение способов получения и обработки сообщений об объектах ОС.	1. Создать проект Windows Forms. 2. Создать слепок системы (snapshot). 3. Найти структуру, содержащую информацию об объектах ОС. 4. Определить назначение всех параметров структуры. 5. Вывести дескриптор рабочего стола.	2
Тема 1.4. Оформление технической документации			
1.4.1 ЕСПД. Требования к оформлению технической документации. 1.4. Использование инструментальных средств для автоматизации оформления документации.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Как называется комплекс государственных стандартов Российской Федерации, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации? 2. Что включает в себя стандарт ЕСПД? 3. Перечислите виды программных документов. 4. Назовите состав информационно-справочных документов.	1
	Изучение ГОСТ 19.101-78, 19.105-78, 19.201-78 и 34.602-89.	Ознакомиться с нормативными документами «Стандарты, определяющие формы и содержание программных документов» (ЕСПД) и ответить на вопросы: 1. Какие бывают виды программ?	

		<p>2. Какие бывают виды программных документов?</p> <p>3. Из каких разделов состоит техническое задание?</p> <p>4. Расскажите о содержании разделов технического задания.</p>	
	Подготовка к зачетному занятию по практическим работам и итоговому занятию.	По типовым вопросам, используя конспект, основную и дополнительную литературу, подготовиться к зачетному занятию по практическим работам и итоговому занятию.	4
Раздел 2. ПМ.01. Разработка программных модулей для прикладного программного обеспечения			
МДК.01.02. Прикладное программирование			
Тема 2.1. Разработка спецификаций для компонентов программного продукта.			
	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
<p>2.1.1 Прикладное программное обеспечение.</p> <p>2.1.2 Основные тенденции образования и развития технологии программирования.</p>	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <ol style="list-style-type: none"> 1. Назовите назначение прикладного ПО. 2. Классифицируйте прикладное ПО. 3. Расшифруйте аббревиатуру ППП. 4. Назовите причину разработки значительного числа ППП одинакового функционального назначения. 5. Перечислите наиболее важные требования к разработке проблемно-ориентированных ППП. 6. Для чего предназначены ППП автоматизированного проектирования? 7. Что такое технология программирования? 8. Перечислите методологии программирования. 9. Поясните термин «стихийное программирование». 10. Поясните термин «структурное программирование». 11. Поясните термин «модульное программирование». 12. Поясните термин «объектно-ориентированное 	1

		программирование».	
	Подготовка сообщений по теме.	Используя Интернет-ресурсы, подготовить сообщение по теме: – Динамические библиотеки DLL. Назначение, особенности применения. – История и современные тенденции развития в ООП в программировании. Наличие презентации приветствуется.	1,5
2.1.3 Понятие жизненного цикла ПО. Стандарты описания ЖЦ. ISO 12207. Процессы ЖЦ.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Дайте определение ЖЦ разработки ПО. 2. Что понимают под этапом в модели ЖЦ разработки ПП? 3. Какие этапы классического ЖЦ вам известны? 4. Охарактеризуйте содержание этапов классического ЖЦ. 5. Объясните достоинства и недостатки классического ЖЦ. 6. Назовите основные группы процессов ЖЦ и перечислите процессы каждой из групп. 7. Назовите организационные процессы и перечислите их. 8. Дайте характеристику процесса управления качеством ЖЦ. 9. Какой международный стандарт определяет перечень и содержание процессов ЖЦ ПО? 10. Все ли процессы, указанные в стандарте, должны быть выполнены при каждой разработке ПО?	1,5
	Изучение ГОСТ Р ИСО/МЭК 12207-2010.	На основе ГОСТ дайте развернутый ответ: 1. Какие ключевые понятия рассматриваются в стандарте? 2. Перечислите процессы в контексте системы, поясните их сущность.	1

<p>Модели ЖЦ ПО: каскадная, итерационная, V-образная, прототипирования, быстрой разработки, спиральная и т.д.</p>	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: <ol style="list-style-type: none"> 1. Что такое модель ЖЦ ПП? 2. Какие основные модели ЖЦ разработки ПП вы знаете? 3. В чем заключаются принципиальные отличия этих моделей? 4. Объясните и охарактеризуйте модель: а) каскадную, б) итерационную, в) V – образную, г) прототипирования, д) RAD, е) спиральную. 5. В чем отличие между моделями: <ol style="list-style-type: none"> а) с промежуточным контролем и каскадной; б) спиральной и каскадной. 6. Что такое риски? 7. Что такое прототип? 	2
<p>2.1.4 Оценка качества процессов создания ПО. Модели ISO и CMM.</p>	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
<p>2.1.5 Понятие технологичности ПО.</p> <p>2.1.6 Определение требований к ПО и исходных данных для его проектирования.</p>	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: <ol style="list-style-type: none"> 1. Что такое качество в отношении процессов создания ПО? 2. Что понимают под моделями качества процессов разработки программного обеспечения? Для чего они разработаны? Что гарантирует сертификация качества процессов? Почему? 3. Какие модели качества вам известны? 4. Чем характеризуется модель ISO? 5. Чем характеризуется модель CMM? 6. Назовите критерии для оценки зрелости компании? 7. Что такое метрика? 8. Перечислите группы метрик. Охарактеризуйте их, приведите примеры. 9. Что подразумевается под технологичностью ПО? 	1,5

		<p>10. Почему мы говорим, что современный этап развития технологии программирования характеризуется переходом от ремесленного к промышленному производству программного обеспечения?</p> <p>11. Поясните сущность нисходящего и восходящего проектирования программ.</p> <p>12. Перечислите основные эксплуатационные требования к ПО.</p>	
2.1.7 Этапы разработки ПО.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
2.1.8 Понятие спецификации ПО.	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы:</p> <p>1. Перечислите этапы разработки ПО и укажите примерное распределение затрат по этапам разработки.</p> <p>2. На каком этапе работ выполняется оценка необходимых ресурсов, объемов и сложности разрабатываемого ПП?</p> <p>3. На каком этапе производится преобразование результатов проектирования в ПП?</p> <p>4. Что такое декомпозиция?</p> <p>5. Что такое спецификация?</p> <p>6. Поясните термин функциональная спецификация.</p> <p>7. Какие сведения содержит в себе спецификация требований программного обеспечения?</p> <p>8. Каким стандартом описывается спецификация требований?</p> <p>9. Почему требования в спецификации требований должны быть однозначными?</p> <p>10. Чем может быть вызвана необходимость внесения изменений в ПП, находящийся в эксплуатации, и на каком этапе эта работа выполняется?</p>	1,5
	Изучение ГОСТ 19.202-78.	<p>На основе ГОСТ ответьте на следующие вопросы:</p> <p>1. Что такое спецификация?</p> <p>2. Какие разделы включает спецификация?</p> <p>3. Поясните назначение и особенности оформления</p>	1

		разделов спецификации.	
2.1.9 Определение спецификаций ПО при структурном подходе.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
2.1.10 Проектирование спецификаций ПО при структурном подходе.	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. В чем сущность структурного подхода к программированию? 2. Что понимают под термином «спецификации»? В чем сложность их уточнения? Назовите модели, используемые в качестве функциональных спецификаций при структурном подходе. Какие характеристики проектируемого программного обеспечения описывает каждая из них? 3. Что является основополагающим принципом структурного подхода? 4. Перечислите виды моделей, используемые при структурном программировании. 5. Поясните термин спецификация процессов. В виде чего они могут быть представлены? 6. Что такое словарь терминов? 7. Что такое диаграмма? 8. Что такое диаграмма переходов состояний (SDT)? В каких случаях целесообразно использовать диаграммы переходов состояний? 9. Поясните назначение функциональных диаграмм. 10. Поясните назначение диаграмм потоков данных (DFD).	1,5
	Подготовка к выполнению практических работ.	Разработать спецификации процессов (структурный подход) для следующих задач: 1. Дан одномерный массив $A[n]$. Выполнить сортировку массива по возрастанию (блок-схема). 2. Вычислить выражение $\max(xyz, x+y+z) - 3$ (flow-диаграмма). 3. Программа сортировки одномерного массива разными методами: перебором, вставками, пузырька	2

		(функциональная диаграмма).	
2.1.11 Определение спецификаций ПО при объектном подходе.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
2.1.12 Проектирование спецификаций ПО при объектном подходе.	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Что такое модель? 2. Дайте характеристику UML. 3. Какие блоки входят в словарь языка UML? 4. Что такое сущность? Какие типы сущностей поддерживаются в UML? 5. Охарактеризуйте структурные сущности. 6. Охарактеризуйте поведенческие сущности. 7. Охарактеризуйте группирующие сущности. 8. Охарактеризуйте аннотационные сущности. 9. Что такое отношение? Какие типы отношений поддерживаются в UML? 10. Что такое диаграмма? 11. Чем характеризуются диаграммы вариантов использования? 12. Чем характеризуется диаграмма классов?	1,5
	Подготовка к выполнению практических работ.	1. Определить актеров, выделить прецеденты и разработать диаграмму прецедентов для следующей задачи: - спортивный клуб. - строительная компания. 2. Построить диаграмму классов: - выставочный комплекс - почтовое отделение.	2
2.1.13 Разработка пользовательских интерфейсов.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Поясните термин конечный пользователь. 2. Что такое интерфейс? 3. В чем особенность пользовательского интерфейса?	1,5

		<p>4. Перечислите свойства пользовательского интерфейса.</p> <p>5. Какие составляющие определяет интерфейс?</p> <p>6. Что такое язык пользователя? Язык сообщений?</p> <p>7. Перечислите стандарты пользовательского интерфейса.</p> <p>8. Что такое usability (практичность пользовательского интерфейса)?</p> <p>9. Перечислите составляющие классификации пользовательских интерфейсов.</p> <p>10. Существуют ли стандарты, определяющие качество пользовательского интерфейса?</p> <p>11. Перечислите этапы разработки пользовательского интерфейса.</p> <p>12. Что необходимо учитывать при разработке пользовательского интерфейса?</p> <p>13. Перечислите разновидности структуры диалога при организации пользовательского интерфейса.</p>	
	Подготовка к зачету.	По типовым вопросам, используя конспект, основную и дополнительную литературу, подготовиться к зачету.	3
Тема 2.2 Разработка кода программного продукта на уровне модуля.			
Объектная модель приложения Microsoft Excel.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: назначение основных элементов объектной модели приложения Microsoft Excel.	4
	Поиск материала в Интернете.	Воспользовавшись Интернет-ресурсами (см. список литературы) найдите подробное описание элементов объектной модели приложения Microsoft Excel.	2
	Решение задач.	Запрограммировать алгоритм обработки ячеек диапазона листа книги Microsoft Excel.	6
Программирование в Visual Studio .NET с использованием основной сборки взаимодействия Microsoft.Office.Interop.Excel.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: назначение	4

		сборки взаимодействия Microsoft.Office.Interop.Excel.	
	Поиск материала в Интернете.	Воспользовавшись Интернет-ресурсами (см. список литературы) найдите подробное описание элементов сборки взаимодействия Microsoft.Office.Interop.Excel.	2
	Решение задач.	Запрограммировать алгоритм обработки ячеек диапазона листа книги Microsoft Excel в среде программирования Microsoft Visual Studio.	6
	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
Объектная модель приложения Microsoft Word.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: назначение основных элементов объектной модели приложения Microsoft Word.	4
	Поиск материала в Интернете.	Воспользовавшись Интернет-ресурсами (см. список литературы) найдите подробное описание элементов объектной модели приложения Microsoft Word.	2
	Решение задач.	Запрограммировать алгоритм обработки текстовых данных документа Microsoft Word.	6
Программирование в Visual Studio .NET с использованием основной сборки взаимодействия Microsoft.Office.Interop.Word.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: назначение сборки взаимодействия Microsoft.Office.Interop.Word.	4
	Поиск материала в интернете.	Воспользовавшись Интернет-ресурсами (см. список литературы) найдите подробное описание элементов сборки взаимодействия Microsoft.Office.Interop.Word.	2
	Решение задач.	Запрограммировать алгоритм обработки текстовых данных Microsoft Word в среде программирования Microsoft Visual Studio.	6
Объектная модель приложения Microsoft Access.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: назначение	3

		основных элементов объектной модели приложения Microsoft Access.	
	Поиск материала в интернете.	Воспользовавшись Интернет-ресурсами (см. список литературы) найдите подробное описание элементов объектной модели приложения Microsoft Access.	2
	Решение задач.	Разработать проект пользовательского интерфейса многофункционального приложения.	6
Настройки Visual Studio .NET на уровне приложений Microsoft Office.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: назначение надстроек на уровне приложений.	1
	Поиск материала в Интернете.	Воспользовавшись Интернет-ресурсами (см. список литературы) найдите подробное описание создания надстроек на уровне приложения.	1
	Решение задач.	Продумать функции меню надстройки конкретного приложения	2
Тема 2.3. Отладка и тестирование программного продукта на уровне модулей.			
Понятие отладки. Принципы и виды отладки.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
Автономная отладка программного средства. Комплексная отладка программного средства.	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: <ol style="list-style-type: none"> 1. Что такое отладка? 2. Какие виды отладки вам известны? Поясните их сущность. 3. Что такое локализация? 4. Перечислите способы обнаружения ошибки. 5. Перечислите принципы локализации ошибок. 6. Перечислите принципы исправления ошибок. 7. Какие методы отладки вам известны? Поясните их сущность. 8. Перечислите средства отладки. 9. Что такое автономная отладка? Когда она применяется? 10. Какие шаги выполняются при автономной отладке? 	3

		<p>11. Что такое комплексная отладка? Когда она применяется?</p> <p>12. Поясните сущность тестирования архитектуры ПС.</p> <p>13. Поясните сущность тестирования внешних функций.</p> <p>14. Поясните сущность тестирования качества ПС.</p> <p>15. Поясните сущность тестирования документации по применению ПС.</p> <p>16. Поясните сущность тестирования определения требований к ПС.</p> <p>17. Что такое ведущий отладочный модуль?</p> <p>18. Что такое отладочный имитатор программного модуля?</p>	
<p>Тестирование: определение и принципы тестирования. Виды ошибок и способы их определения.</p> <p>Стратегия проектирования тестов. Методология тестирования.</p> <p>Разновидности тестирования.</p>	<p>Составление глоссария.</p>	<p>Найти в конспекте все основные определения по теме и выделить их.</p>	<p>1</p>
	<p>Проработка конспекта лекций.</p>	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос:</p> <ol style="list-style-type: none"> 1. Что такое тестирование ПП? 2. Каково назначение этапа тестирования в ЖЦ разработки ПП? 3. Перечислите принципы тестирования ПП. 4. Что подвергается тестированию в течение ЖЦ разработки ПП? 5. Какие бывают стратегии тестирования? 6. Какие существуют уровни тестирования и кто ответственен за проведение тестирования на каждом из них? 7. Что является программной ошибкой? 8. Какими характеристика должен обладать хороший тест? 9. Поясните термин методология тестирования. 10. В чем заключается метод восходящего тестирования? 11. В чем заключается метод нисходящего тестирования? 12. Сравните методы восходящего и нисходящего 	<p>2</p>

		<p>тестирования.</p> <p>13. Какого назначение и основные элементы тестирования: а) модульного, б) интеграционного, в) системного, г) выходного?</p> <p>14. Объясните значения терминов «черный ящик», «белый ящик».</p>	
	Построение схемы.	На основании конспекта лекций и основной литературы, построить схему классификации тестирования ПП.	1
Процесс тестирования: этапы и задачи. Тест план.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
Дефекты, причины, описание, отслеживание.	Проработка конспекта лекций.	<p>Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос:</p> <ol style="list-style-type: none"> 1. Перечислите основные этапы ЖЦ тестирования. 2. Какие задачи ставятся перед тестированием на каждом этапе? 3. Какими стандартами регламентируется ЖЦ тестирования ПП? 4. Что такое тест план? 5. Что такое тестовая ситуация? Какую структуру имеет test case? 6. Что такое ошибка? Дефект? 7. Перечислите основные источники ошибок. 8. Поясните назначение системы отслеживания ошибок. 9. Что такое bug report и какую структуру он имеет? 10. Что такое серьезность ошибки? Какую градацию она имеет? 11. Что такое приоритет ошибки? Какую градацию он имеет? 12. Перечислите основные ошибки при написании bug report. 13. Перечислите состояния жизненного цикла дефекта. 	2
	Построение тестового случая, отчета об ошибке.	<p>Приложение рассчитывает значение выражения, коэффициенты вводятся пользователем вручную.</p> <p>– Построить test case (позитивный и негативный тесты).</p>	1

		– Создать отчет об ошибке.	
Техники создания тестов для черного ящика.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: 1. Поясните сущность методики тестирования «Черный ящик». 2. Когда применяется эта методика? 3. Какие техники создания тестов для черного ящика вам известны? 4. Что такое класс эквивалентности? 5. Какие условия должны соблюдаться для того, чтобы группа тестов представляла собой класс эквивалентности? 6. Какие критерии используют для определения класса эквивалентности? 7. Дайте определение понятия «границы класса эквивалентности». 8. Поясните сущность техники «Предположение об ошибке».	2
Структурное тестирование: на основе потока управления, на основе потока данных.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: 1. Что такое структурное тестирование? 2. Поясните сущность управляющего графа программы. Как он строится? 3. Поясните сущность информационного графа программы. Как он строится?	0,5
	Подготовка к практической работе.	– Построить управляющий граф программы. – Составить трассы покрытия всех путей.	2
Модульное тестирование (unit-тестирование). Особенности.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: 1. Поясните термин модульное тестирование.	2

		<ul style="list-style-type: none"> 2. Какие достоинства и преимущества имеет unit-тестирование? 3. Поясните назначение stub- и mock- объектов. 4. Когда целесообразно применение unit-тестирования? 	
Автоматизация. Нагрузочное тестирование.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
Анализ результатов тестирования программы.	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: <ul style="list-style-type: none"> 1. Что такое нагрузочное тестирование? Какие особенности оно имеет? 2. Перечислите разновидности нагрузочного тестирования ПП. 3. Укажите преимущества и недостатки автоматизации нагрузочного тестирования. 4. Какие результаты тестирования документируются и с какой целью? 	0,5
	Подготовка сообщений по теме.	Используя Интернет-ресурсы, подготовить сообщение по теме: <ul style="list-style-type: none"> 1. Обзор средств автоматизации тестирования ПО. 2. Преимущества и недостатки автоматизированного тестирования ПО. 3. Интеграционное тестирование. 4. Регрессионное тестирование. 	2
Испытание программных средств. Планирование и оценка испытаний.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопрос: <ul style="list-style-type: none"> 1. Поясните термин аттестация ПС. 2. Перечислите виды испытаний ПС. 3. Перечислите методы оценки качества ПС. 4. Что такое верификация? Уточните разницу между верификацией и аттестацией ПП. 5. Для чего нужно планирование верификации и аттестации? 	1

		6. Какие этапы включает в себя планирование испытаний?	
	Изучение ГОСТ 19.301-79.	Ознакомиться с нормативным документом и ответить на вопросы: 1. Из каких разделов состоит документ «Программа и методика испытаний»? 2. Поясните содержание разделов документа «Программа и методика испытаний». 3. В соответствии с какими нормативами должен оформляться документ «Программа и методика испытаний»?	1
	Подготовка к зачету.	По типовым вопросам, используя конспект, основную и дополнительную литературу, подготовиться к зачету.	2,5
Тема 2.4. Документирование.			
	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
2.3.1 Сопровождение программ. Определение программного документа. Документирование ПО в соответствии с ЕСПД. ГОСТ 19.XXX. Виды программных продуктов. 2.3.2 Средства разработки технической документации. Автоматизированные средства оформления документации.	Проработка конспекта лекций.	Используя конспект и основную литературу (см. список литературы) дать развернутый ответ на вопросы: 1. Каково назначение этапа сопровождения в ЖЦ ПП? 2. Дайте пояснение термина программный документ. 3. Перечислите виды программных документов. 4. Какими стандартами регламентируются виды сопроводительной документации? 5. Какие задачи выполняют средства разработки технической документации? 6. Поясните особенности использования автоматизированных средств оформления документации. Приведите примеры средств.	1,5
Описание программы. Определение, структура, стандарты.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Изучение ГОСТ 19.402-78.	Ознакомиться с нормативными документами «Стандарты, определяющие формы и содержание программных документов» (ЕСПД) и ответить на вопросы: 1. Каковы требования к содержанию и оформлению программного документа «Текст программы»?	0,5

		<p>2. Каковы требования к содержанию и оформлению программного документа «Описание программы»?</p> <p>3. Каковы требования к содержанию и оформлению программного документа «Описание применения»?</p>	
Руководство пользователя. Определение, структура, стандарты.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	1
	Изучение ГОСТ 19.505-79 «Руководство оператора», РД 50-34.698-90 (п.п. 3.4 «Руководство пользователя)»	<p>Ознакомиться с нормативными документами и ответить на вопросы:</p> <p>1. Из каких разделов состоит программный документ «Руководство оператора»?</p> <p>2. Поясните содержание разделов программного документа «Руководство оператора»</p>	2
Руководство программиста. Определение, структура, стандарты.	Составление глоссария.	Найти в конспекте все основные определения по теме и выделить их.	0,5
	Изучение ГОСТ 19.504-79.	<p>Ознакомиться с нормативными документами и ответить на вопросы:</p> <p>1. Из каких разделов состоит программный документ «Руководство системного программиста»?</p> <p>2. Поясните содержание разделов программного документа «Руководство системного программиста».</p> <p>3. Из каких разделов состоит программный документ «Руководство программиста»?</p> <p>4. Поясните содержание разделов программного документа «Руководство программиста».</p>	1
	Подготовка к зачету.	По типовым вопросам, используя конспект, основную и дополнительную литературу, подготовиться к зачету.	1,5

