

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Колледж

УТВЕРЖДАЮ
Проректор по УВР и ОВ
Т.Е. Наливайко
_____ 2021г.



ПРОГРАММА КВАЛИФИКАЦИОННОГО ЭКЗАМЕНА

по профессиональному модулю ПМ.1 «Разработка программных модулей
программного обеспечения для компьютерных систем»

по специальности среднего профессионального образования
09.02.03 – «Программирование в компьютерных системах»
(базовая подготовка)

на базе основного общего образования

Форма обучения

очная

Комсомольск-на-Амуре, 2021

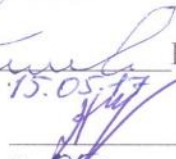
Программа квалификационного экзамена по ПМ.1 «Разработка программных модулей программного обеспечения для компьютерных систем» составлена на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования 09.02.03 «Программирование в компьютерных системах», утвержденного приказом Министерства образования и науки Российской Федерации от 28.07.2014 № 804.

Комплект контрольно-оценочных средств рассмотрен и утвержден на заседании кафедры «МОП ЭВМ»

Протокол № 13 от «15» 05 2017 г.

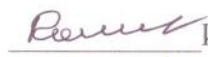
Заведующий кафедрой «МОП ЭВМ»  В.А. Тихомиров

Разработчик КОС


 О.Н. Попенко
«05» 05 2017 г.

СОГЛАСОВАНО

Декан факультета довузовской подготовки

 И.В. Конырева
«19» 05 2017 г.


Начальник учебно-методического управления

 Е.Е. Поздеева
«24» 05 2017 г.

Рецензент

руководитель проекта по реализации ФЦП, ПАО «Амурский судостроительный завод»



 В.А. Ханов
«14» 05 2017 г.

СОДЕРЖАНИЕ

1	Паспорт комплекта контрольно-оценочных средств	4
2	Результаты освоения профессионального модуля, подлежащие проверке	5
3	Оценка освоения профессионального модуля:	11
	3.1 Формы и методы оценивания	11
	3.2 Типовые задания для оценки освоения профессионального модуля	13
4	Контрольно-оценочные материалы для итоговой аттестации по профессионального модулю	24
5	Критерии и шкалы для интегрированной оценки уровня сформированности компетенций	36
6	Список рекомендуемой литературы и источников информации	37

1 Паспорт комплекта контрольно-оценочных средств

В результате освоения ПМ «Разработка программных модулей программного обеспечения для компьютерных систем» обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.03 - «Программирование в компьютерных системах» следующими знаниями, умениями, которые формируют общие и профессиональные компетенции.

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 команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7 - Брать на себя ответственность за работу членов команды (подчиненных), 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 3.1 - анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

ПК 3.2 - выполнять интеграцию модулей в программную систему.

ПК 3.3 - выполнять отладку программного продукта с использованием специализированных программных средств.

ПК 3.4 - осуществлять разработку тестовых наборов и тестовых сценариев.

ПК 3.5 - производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.

ПК 3.6 - разрабатывать технологическую документацию.

Формой промежуточной аттестации по модулю является экзамен.

2 Результаты освоения учебного модуля, подлежащие проверке

В результате текущей аттестации по модулю осуществляется комплексная проверка следующих умений (У) и знаний (З), а также динамика формирования общих (ОК) и профессиональных (ПК) компетенций.

Таблица 1

Результат обучения	Показатели оценки результата	Формы контроля
ОК 1 - Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.	Знать: 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.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ОК 2 - Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.	Знать: 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.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ОК 3 - Принимать решения в стан-	Знать: З1 – основные этапы разработки программного	Устный опрос, самостоятельная

дартных и нестандартных ситуациях и нести за них ответственность.	обеспечения; 32 – основные принципы технологии структурного и объектно-ориентированного программирования; 33 – основные принципы отладки и тестирования программных продуктов; 34 – методы и средства разработки технической документации.	работа, лабораторная работа, экзамен
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ОК 4 - Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.	Знать: 31 – основные этапы разработки программного обеспечения; 32 – основные принципы технологии структурного и объектно-ориентированного программирования; 33 – основные принципы отладки и тестирования программных продуктов; 34 – методы и средства разработки технической документации.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ОК 5 - Использовать информационно-коммуникационные технологии в профессиональной деятельности.	Знать: 31 – основные этапы разработки программного обеспечения; 32 – основные принципы технологии структурного и объектно-ориентированного программирования; 33 – основные принципы отладки и тестирования программных продуктов; 34 – методы и средства разработки технической документации.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: У1 – осуществлять разработку кода программного модуля на современных языках программирования; У2 – создавать программу по разработанному алгоритму как отдельный модуль;	Самостоятельная работа, лабораторная работа, экзамен

	<p>У3 – выполнять отладку и тестирование программы на уровне модуля;</p> <p>У4 – оформлять документацию на программные средства;</p> <p>У5 – использовать инструментальные средства для автоматизации оформления документации.</p>	
ОК 6 - Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.	<p>Знать:</p> <p>31 – основные этапы разработки программного обеспечения;</p> <p>32 – основные принципы технологии структурного и объектно-ориентированного программирования;</p> <p>33 – основные принципы отладки и тестирования программных продуктов;</p> <p>34 – методы и средства разработки технической документации.</p>	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	<p>Уметь:</p> <p>У1 – осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>У2 – создавать программу по разработанному алгоритму как отдельный модуль;</p> <p>У3 – выполнять отладку и тестирование программы на уровне модуля;</p> <p>У4 – оформлять документацию на программные средства;</p> <p>У5 – использовать инструментальные средства для автоматизации оформления документации.</p>	Самостоятельная работа, лабораторная работа, экзамен
ОК 7 - Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.	<p>Знать:</p> <p>31 – основные этапы разработки программного обеспечения;</p> <p>32 – основные принципы технологии структурного и объектно-ориентированного программирования;</p> <p>33 – основные принципы отладки и тестирования программных продуктов;</p> <p>34 – методы и средства разработки технической документации.</p>	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	<p>Уметь:</p> <p>У1 – осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>У2 – создавать программу по разработанному алгоритму как отдельный модуль;</p> <p>У3 – выполнять отладку и тестирование программы на уровне модуля;</p> <p>У4 – оформлять документацию на программные средства;</p> <p>У5 – использовать инструментальные средства для автоматизации оформления документации.</p>	Самостоятельная работа, лабораторная работа, экзамен
ОК 8 - Самостоятельно определять задачи профессионального и личностного развития, заниматься самооб-	<p>Знать:</p> <p>31 – основные этапы разработки программного обеспечения;</p> <p>32 – основные принципы технологии структурного и объектно-ориентированного программирования;</p> <p>33 – основные принципы отладки и тестирования</p>	Устный опрос, самостоятельная работа, лабораторная работа, экзамен

разованием, осознанно планировать повышение квалификации.	программных продуктов; З4 – методы и средства разработки технической документации.	
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ОК 9 - Ориентироваться в условиях частой смены технологий в профессиональной деятельности.	Знать: 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.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ПК 1.1 - Выполнять разработку спецификаций отдельных компонент.	Знать: 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.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: 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;	Самостоятельная работа, лабораторная работа, экзамен

	У5 – использовать инструментальные средства для автоматизации оформления документации.	
ПК 1.2 - Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.	Знать: 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.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ПК 1.3 - Выполнять отладку программных модулей с использованием специализированных программных средств.	Знать: 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.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен
	Уметь: 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.	Самостоятельная работа, лабораторная работа, экзамен
ПК 1.4 - Выполнять тестирование программных модулей.	Знать: 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.	Устный опрос, самостоятельная работа, лабораторная работа, экзамен

	<p>Уметь:</p> <p>У1 – осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>У2 – создавать программу по разработанному алгоритму как отдельный модуль;</p> <p>У3 – выполнять отладку и тестирование программы на уровне модуля;</p> <p>У4 – оформлять документацию на программные средства;</p> <p>У5 – использовать инструментальные средства для автоматизации оформления документации.</p>	<p>Самостоятельная работа, лабораторная работа, экзамен</p>
<p>ПК 1.5 - Осуществлять оптимизацию программного кода модуля.</p>	<p>Знать:</p> <p>З1 – основные этапы разработки программного обеспечения;</p> <p>З2 – основные принципы технологии структурного и объектно-ориентированного программирования;</p> <p>З3 – основные принципы отладки и тестирования программных продуктов;</p> <p>З4 – методы и средства разработки технической документации.</p>	<p>Устный опрос, самостоятельная работа, лабораторная работа, экзамен</p>
	<p>Уметь:</p> <p>У1 – осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>У2 – создавать программу по разработанному алгоритму как отдельный модуль;</p> <p>У3 – выполнять отладку и тестирование программы на уровне модуля;</p> <p>У4 – оформлять документацию на программные средства;</p> <p>У5 – использовать инструментальные средства для автоматизации оформления документации.</p>	<p>Самостоятельная работа, лабораторная работа, экзамен</p>
<p>ПК 1.6 - Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.</p>	<p>Знать:</p> <p>З1 – основные этапы разработки программного обеспечения;</p> <p>З2 – основные принципы технологии структурного и объектно-ориентированного программирования;</p> <p>З3 – основные принципы отладки и тестирования программных продуктов;</p> <p>З4 – методы и средства разработки технической документации.</p>	<p>Устный опрос, самостоятельная работа, лабораторная работа, экзамен</p>
	<p>Уметь:</p> <p>У1 – осуществлять разработку кода программного модуля на современных языках программирования;</p> <p>У2 – создавать программу по разработанному алгоритму как отдельный модуль;</p> <p>У3 – выполнять отладку и тестирование программы на уровне модуля;</p> <p>У4 – оформлять документацию на программные средства;</p> <p>У5 – использовать инструментальные средства для автоматизации оформления документации.</p>	<p>Самостоятельная работа, лабораторная работа, экзамен</p>

3 Оценка освоения дисциплины

3.1 Формы и методы оценивания

Предметом оценки служат умения и знания, предусмотренные ФГОС СПО по специальности, направленные на формирование общих и профессиональных компетенций.

В результате промежуточной аттестации по профессиональному модулю осуществляется комплексная проверка умений и знаний, а также динамика формирования общих и профессиональных компетенций (таблица 2).

Самостоятельная подготовка студентов по дисциплине предполагает следующие формы:

- работа с информационными источниками (поиск дополнительной информации по заданной теме в различных информационных источниках, таких как учебники, справочники, журналы и интернет);
- творческие задания (выполнение различных заданий, связанных с развитием творческого воображения и умением применять свои знания в самостоятельной работе);
- подготовка презентационных материалов (поиск и систематизация заданной информации, ее обработка, оформление в виде презентации и выступление в виде защиты перед группой);
- оформление лабораторных работ, отчетов и подготовка к их защите.

При оценивании лабораторной и самостоятельной работы студента учитывается:

- качество выполнения практической части работы;
- качество оформления отчета по работе;

Качество устных ответов на контрольные вопросы при защите работы.

Элемент учебной дисциплины	Формы и методы контроля					
	Текущий контроль		Рубежный контроль (другие формы контроля)		Промежуточная аттестация(экзамен)	
	Форма контроля	Проверяемые ПК, ОК, У, З	Форма контроля	Проверяемые ПК, ОК, У, З	Форма контроля	Проверяемые ПК, ОК, У, З
Раздел 1 Системное программирование.	Самостоятельная работа Домашнее задание Проектно-исследовательская деятельность Лабораторные работы	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4	Тестирование	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4	Дифференцированный зачет	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4

Элемент учебной дисциплины	Формы и методы контроля					
	Текущий контроль		Рубежный контроль (другие формы контроля)		Промежуточная Аттестация(экзамен)	
	Форма контроля	Проверяемые ПК, ОК, У, З	Форма контроля	Проверяемые ПК, ОК, У, З	Форма контроля	Проверяемые ПК, ОК, У, З
Раздел 2 Прикладное программирование.	Самостоятельная работа Домашнее задание Проектно-исследовательская деятельность Лабораторные работы	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4	Тестирование	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4	Дифференцированный зачет	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4
Раздел 3 Программирование для мобильных устройств.	Самостоятельная работа Домашнее задание Проектно-исследовательская деятельность Лабораторная работа	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4	Тестирование	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4	Дифференцированный зачет	ПК 1.1-1.6 ОК 1-9 У 1-5 З 1-4

3.2 Типовые задания для оценки освоения учебной дисциплины

1) Входное тестирование

Вариант 1

1) Системная программа – это:

1) программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.

2) программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ.

3) системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СООИ, восстановление работы системы после проявления неисправностей в технических средствах.

2) Исходный модуль – это:

1) программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

2) программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

3) программный модуль, получаемый в результате трансляции исходного модуля.

3) Автокод – это:

1) символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.

2) язык программирования, понятия и структура которого удобны для восприятия человеком.

3) язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

4)... - реализация смысла некоторого синтаксически законченного текста, представленного на конкретном языке.

1) Интерпретация.

2) Трансляция.

3) Компиляция.

5) Программное обеспечение – это:

1) совокупность программ СООИ и программных документов, необходимых для их эксплуатации.

2) комплекс программ, которые обеспечивают управление компонентами компьютерной системы.

3) компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы.

6) Регистр – это:

1) Ячейка в оперативной памяти.

2) Ячейка памяти процессора.

3) Адресуемая ячейка памяти.

7) Какие регистры относятся к сегментным:

1) Es.

2) Dх.

3) Ip.

4) Cs.

5) Al.

8) Команда mov –

- 1) Команда помещения в стек.
2) Команда перемещения данных.
 3) Команда пересылки байта.
- 9) Какая команда относится к логическим командам:
 1) Sub.
2) Or.
 3) Inc.
- 10) Команда ... начинает выполнение с новой ветки в любом случае.
1) Безусловного перехода.
 2) Условного перехода.
 3) Цикла.
- 11) Объявляется непосредственно внутри ассемблера - ...
1) Локальная метка.
 2) Глобальная метка.
 3) Системная метка.
- 12) Прерывание, которое возникает при изменении какого-либо устройства:
 1) Внутренние.
2) Аппаратное.
 3) Программное.
- 13) Какой командой вызывается процедура:
 1) Proc.
2) Call.
 3) Ret.
- 14) Директива – это
1) Символическое имя для определения места в памяти, где размещены данные.
 2) Блок команд, который может быть вызван из любого места основной программы.
 3) Процесс приостановке программы.
- 15) Какой командой вызывается прерывания:
 1) In.
 2) Inc.
3) Int.
- 16) Определить ошибки в программе:
 1 Var X: integer;
 2 Begin
 3 X:=100;
 4 asm
 5 Mov bl, 25
 6 Mov ax, X
 7 Div ax,bl
 8 Mov X,bl
 9 end;
 10 writeln ('X=',X);
 11 end.
 1) Строка 5,7.
 2) Строка 6,7.
3) Строка 7,8.
- 17) Определить результат выполнения программы:
 Begin
 asm
 Mov Ah,02h

Mov DI,'*'
Int 21h
end;
end.

- 1) **Вывести символ *.**
- 2) Вывести символ с кодом *.
- 3) Вывести символ * 2 раза.

Вариант 2

1) Управляющая программа - это
1) программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.

2) программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ.

3) системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СООИ, восстановление работы системы после проявления неисправностей в технических средствах.

2) Объектный модуль – это:

1) программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

2) программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

3) программный модуль, получаемый в результате трансляции исходного модуля.

3) Язык высокого уровня – это:

1) символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.

2) язык программирования, понятия и структура которого удобны для восприятия человеком.

3) язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

4) ... - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

1) Интерпретация.

2) Трансляция.

3) Компиляция.

5) Прикладное программное обеспечение – это

1) совокупность программ СООИ и программных документов, необходимых для их эксплуатации.

2) комплекс программ, которые обеспечивают управление компонентами компьютерной системы.

3) компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы.

6) Язык Ассемблер – это:

1) система записи программы с детализацией до отдельной машинной команды.

2) язык программирования, понятия и структура которого удобны для восприятия человеком.

3) система, образуемая языком программирования, компилятором или интерпретатором программ.

7) Какие регистры относятся к регистрам данных:

- 1) Es.
- 2) **Dx.**
- 3) Ip.
- 4) Cs.
- 5) **Al.**

8) Команда add –

- 1) **Команда сложения**
- 2) Команда умножения
- 3) Команда вычитания

9) Какая команда относится к арифметическим командам:

- 1) In.
- 2) Or.
- 3) **Inc.**

10) Команда ... повторяет действия несколько раз.

- 1) Безусловного перехода.
- 2) Условного перехода.
- 3) **Цикла.**

11) Ссылка на определенное место в программе - ...

- 1) Адресация.
- 2) **Метка.**
- 3) Переход.

12) Прерывание, которое возникает при существовании ошибок в программе:

- 1) Внутренние.
- 2) Аппаратное.
- 3) **Программное.**

13) Какой командой осуществляется начало процедуры:

- 1) **Proc.**
- 2) Call.
- 3) Ret.

14) Прерывания – это:

- 1) Символическое имя для определения места в памяти, где размещены данные.
- 2) Блок команд, который может быть вызван из любого места основной программы.

3) **Процесс приостановки программы.**

15) Какие команды относятся к директивам:

- 1) **DD.**
- 2) DE.
- 3) **DB.**
- 4) DC.
- 5) **DW.**

16) Определить ошибки в программе:

- 1 Var a;boolean;
- 2 begin
- 3 asm
- 4 Mov AX,00111011b
- 5 Or Al,00001111b
- 6 Mov a,AL
- 7 end;
- 8 Writeln('A=',A);

9 end.

1) Строка 1,4.

2) Строка 1,5.

3) Строка 4,6.

17) Определить результат выполнения программы:

```
Var x:integer;
```

```
Begin
```

```
X:=50;
```

```
asm
```

```
mov ax,x
```

```
sub ax,13
```

```
sub ax,4
```

```
sub ax,20
```

```
mov x,ax
```

```
end;
```

```
writeln('x=',x);
```

```
end.
```

1) 13.

2) 14.

3) 15.

Вариант 3

1) Управляющая программа – это:

1) программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.

2) программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ.

3) системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СООИ, восстановление работы системы после проявления неисправностей в технических средствах.

2) Программный модуль – это:

1) программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

2) программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

3) программный модуль, получаемый в результате трансляции исходного модуля.

3) Машинный язык – это:

1) символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.

2) язык программирования, понятия и структура которого удобны для восприятия человеком.

3) язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

4) ... - преобразование программы на машинный язык.

1) Интерпретация.

2) Трансляция.

3) Компиляция.

5) Системное программное обеспечение – это:

1) совокупность программ СООИ и программных документов, необходимых для

их эксплуатации.

2) комплекс программ, которые обеспечивают управление компонентами компьютерной системы.

3) компьютерные программы, написанные для пользователей или самими пользователями, для задания компьютеру конкретной работы.

6) Адресация предназначена ...

1) Для указания адреса ячеек.

2) Для обмена данными.

3) Для помещения данных в регистры.

7) Назовите регистр указания команд:

1) Es.

2) Dx.

3) Ip.

4) Cs.

5) Al.

8) Команда pop –

1) Команда помещения в стек.

2) Команда перемещения данных.

3) Команда извлечения из стека.

9) Какая команда относится к битовым командам:

1) Bt.

2) Or.

3) Inc.

10) Команда ... начинает выполнение с новой ветки в случае, если условие ложно.

1) Безусловного перехода.

2) Условного перехода.

3) Цикла.

11) Объявляется в Паскале - ...

1) Локальная метка.

2) Глобальная метка.

3) Системная метка.

12) Прерывание, которое возникает при обнаружении ошибок в работе процессора:

1) Внутренние.

2) Аппаратное.

3) Программное.

13) Какой командой осуществляется возврат из процедуры:

1) Proc.

2) Call.

3) Ret.

14) Подпрограмма – это:

1) Символическое имя для определения места в памяти, где размещены данные.

2) Блок команд, который может быть вызван из любого места основной программы.

3) Процесс приостановке программы.

15) Какие функции относятся к прерыванию 21H:

1) 02h.

2) 09h.

3) 08h.

4) 2ah.

5) 5fh.

16) Определить ошибки в программе:

1 Var

```

2 str:='gruppa$';
3 asm
4 Mov Cx,3
5 @Lab:
6 Mov Ah,9
7 Mab Dx,offset str
8 int 21h
9 Loop @Lab
10 end;
11 readln;
12 end.

```

1) Строка 4,6.

2) Строка 4,7.

3) Строка 6,7.

17) Определить результат выполнения программы:

```

Begin
asm
Mov ax,6;
Mov bx,8;
Mov dx, ax;
cmp ax, bx;
jae Li;
Mov dx, bx;
Li:
mov dx,ax
end;
end.

```

1) 6.

2) 8.

3) 14.

2) Самостоятельная работа

- работа с информационными источниками;
- творческие задания;
- подготовка презентационных материалов.

3) Домашнее задание

- Дать определение «Системное программирование»;
- Изучить историю создания языка Ассемблер;
- Дать определение понятию «Регистр»;
- Провести сравнение написания арифметических операций;
- Рассмотреть табличный поиск;
- Закомментировать код программы;
- Закончить ввод программы;
- Рассмотреть дополнительные директивы;
- Рассмотреть модульное программирование для разных языков программирования;
- Подготовиться к сдаче дифференцированного зачета;
- Создать глоссарий;
- Повторение материала по языку программирования C++;
- Повторение конспектов по библиотекам;
- Составить классификацию мобильных устройств;
- Преимущества и недостатки других платформ;
- Подбор ресурсов;
- Создание макета пользовательского интерфейса;

- Создание глоссария;
- Разобрать различные БД;
- Повторение материала по сетям.

4) Проектно-исследовательская деятельность

- Место языков Ассемблера среди языков программирования.
- Структура МП Intel 80x86: используемые регистры.
- Структура МП Intel 80x86: операционное устройство и шинный интерфейс.
- Размещение данных в памяти. Сегментация памяти.
- Структура регистра флагов. Команды установки флагов.
- Структура и форматы команд МП Intel 80x86. Команды пересылки данных.
- Способы адресации в командах МП Intel 80x86.
- Представление данных в IBM PC: целые числа.
- Представление данных в IBM PC: двоично-десятичные числа.
- Представление данных в IBM PC: алфавитно-цифровые данные.
- Задание массивов через указатели.
- Задание функции через указатели.
- Использование процессорных средств. Модифицировать лабораторные работы, используя процессорные средства.
- Использование множественное наследование класса. Потомки и родительские классы.
- Использование спецификатора virtual при множественном наследовании.
- Изучение функций с динамическим числом аргументов.
- Изучение шаблонов классов и функций.
- Использование спецификатора friend при определении классов и методов.
- Типичные ошибки при работе со множественным наследованием.
- Вставка и удаление элементов в массиве с помощью вектора.
- Операции вставки, удаления и просмотра элементов в односвязном списке.
- Решение задач с помощью итераторов.
- Операции с элементами в очередях и стеке.
- Подключение компонентов библиотеки Win API 32 к проекту.
- Графические библиотеки Direct X. Устройство и возможность подключения в C++.
- Графическая библиотека Open GL. Устройство и возможность подключения в C++.
- История возникновения мобильных устройств.
- Операционные системы мобильных устройств.
- Среды разработки под iPhone, Windows Store.
- История Android.
- Особенности платформы Android.
- Безопасность.
- Полномочия.
- стек Активностей.
- Отслеживание состояний Активностей.
- Локализация интерфейса.
- Юзабилити.
- Привязка данных.
- Общие настройки.
- Статические файлы.
- Курсоры.
- Работа с СУБД без адаптера.
- Simple Cursor Adapter.
- Диалоги.

- Ordered Broadcast.
- Pending Intent.
- Подготовка к публикации разработанного мобильного приложения.
- Развертывание приложения в Google-маркете

5) Лабораторная работа

Раздел 1:

Лабораторная работа 01 «Программы и программное обеспечение».

Лабораторная работа 02 «Этапы подготовки программы».

Лабораторная работа 03 «Предложения языка».

Лабораторная работа 04 «Регистры».

Лабораторная работа 05 «Двоичное и шестнадцатеричное представление».

Лабораторная работа 06 «Семестры».

Лабораторная работа 07 «Способы адресации».

Лабораторная работа 08 «Директивы определения данных».

Лабораторная работа 09 «Директива определения байта».

Лабораторная работа 10 «Директива определения слова».

Лабораторная работа 11 «Сегментные регистры: CS, DS, SS и ES».

Лабораторная работа 12 «Регистры общего назначения: AX, BX, CX и DX».

Лабораторная работа 13 «Регистровые указатели: SP и BP».

Лабораторная работа 14 «Индексные регистры: SI и DI».

Лабораторная работа 15 «Регистр командного указателя: IP».

Лабораторная работа 16 «Флаговый регистр».

Лабораторная работа 17 «Умножение».

Лабораторная работа 18 «Сдвиг регистровой пары DX:AX».

Лабораторная работа 19 «Деление».

Лабораторная работа 20 «Обработка данных в форматах ASCII и BCD».

Лабораторная работа 21 «Сдвиг и округление».

Лабораторная работа 22 «REP: Префикс повторения цепочечной команды».

Лабораторная работа 23 «MOVS: Пересылка строк».

Лабораторная работа 24 «LODS: Загрузка строки».

Лабораторная работа 25 «STOS: Запись строки».

Лабораторная работа 26 «CMPS: Сравнение строк».

Лабораторная работа 27 «SCAS: Сканирование строк».

Лабораторная работа 28 «Прямой табличный доступ».

Лабораторная работа 29 «Транслирующая команда XLAT».

Лабораторная работа 30 «Комментарии в программах на Ассемблере».

Лабораторная работа 31 «Инициализация программы».

Лабораторная работа 32 «Ввод программы».

Лабораторная работа 33 «Ассемблирование программы».

Лабораторная работа 34 «Компоновка программы».

Лабораторная работа 35 «Файл перекрестных ссылок».

Лабораторная работа 36 «Двухпроходной Ассемблер – первый проход».

Лабораторная работа 37 «Структура таблиц Ассемблера».

Лабораторная работа 38 «Двухпроходной Ассемблер – второй проход».

Лабораторная работа 39 «Директивы связывания».

Лабораторная работа 40 «Одно- и многопроходной Ассемблер».

Лабораторная работа 41 «Этапы разработки программы на Ассемблере».

Лабораторная работа 42 «Трансляция, компоновка и отладка программы».

Лабораторная работа 43 «Режимы адресации».

Лабораторная работа 44 «Десятичные числа. Сложение и вычитание целых чисел.

Умножение и деление целых чисел».

Лабораторная работа 45 «Программирование арифметических задач».

Лабораторная работа 46 «Логические команды и команды сдвига».

Лабораторная работа 47 «Программирование ветвлений и циклов. Команды условного перехода. Команда безусловного перехода. Организация циклов».

Лабораторная работа 48 «Применение команд цикла для программирования действий с массивами данных. Организация циклов».

Лабораторная работа 49 «Массивы».

Лабораторная работа 50 «Команды пересылки данных. стек».

Лабораторная работа 51 «Программирование операций ввода-вывода».

Лабораторная работа 52 «Подпрограммы».

Лабораторная работа 53 «Работа с файлами».

Лабораторная работа 54 «Работа с графикой».

Лабораторная работа 55 «Комплексная работа».

Лабораторная работа 56 «Составление эскизного проекта».

Раздел 2:

Лабораторная работа 01 «Операторы ввода и вывода на языке C++».

Лабораторная работа 02 «Условные и безусловные операторы в языке C++».

Лабораторная работа 03 «Операторы цикла в языке C++».

Лабораторная работа 04 «Массивы и указатели».

Лабораторная работа 05 «Работа с функциями в C++».

Лабораторная работа 06 «Препроцессорные средства в C++».

Лабораторная работа 07 «Работа с файлами и строками в языке C++».

Лабораторная работа 08 «Простые классы: Доступ к элементам класса».

Лабораторная работа 09 «Простые классы: Шаблоны классов и функций C++».

Лабораторная работа 10 «Перегрузка операторов в C++».

Лабораторная работа 11 «Полиморфизм».

Лабораторная работа 12 «Множественное наследование в C++».

Лабораторная работа 13 «Виртуальный базовый класс и наследование».

Лабораторная работа 14 «Дружественный класс и дружественные функции».

Лабораторная работа 15 «Работа с динамической памятью».

Лабораторная работа 16 «Конструкторы и деструкторы».

Лабораторная работа 17 «Контейнеры».

Лабораторная работа 18 «Vector – динамический массив».

Лабораторная работа 19 «Списки».

Лабораторная работа 20 «Односвязный и двухсвязный списки».

Лабораторная работа 21 «Деревья».

Лабораторная работа 22 «Бинарное дерево».

Лабораторная работа 23 «Использование итераторов в C++».

Лабораторная работа 24 «Использование очереди и стека в C++».

Лабораторная работа 25 «Создание пользовательского интерфейса для системы тестирования».

Лабораторная работа 26 «Интерфейс администратора системы тестирования».

Лабораторная работа 27 «Использование библиотек Win API 32».

Лабораторная работа 28 «Создание окна».

Лабораторная работа 29 «Использование библиотек Win API 32: Функции CreateFile, ReadFile, WriteFile».

Раздел 3:

Лабораторная работа 01 «Сравнение языков программирования».

Лабораторная работа 02 «Преимущества и недостатки платформы».

Лабораторная работа 03 «Архитектура Android Studio».

Лабораторная работа 04 «Эмулятор мобильного устройства».

Лабораторная работа 05 «Создание активности».

Лабораторная работа 06 «Ресурсы мобильного приложения».

Лабораторная работа 07 «Создание и использование ресурсов: картинки, стили, темы и др.».

Лабораторная работа 08 «Класс Application».

Лабораторная работа 09 «Меню».

Лабораторная работа 10 «Разметка».

Лабораторная работа 11 «Представление».

Лабораторная работа 12 «События».

Лабораторная работа 13 «Анимация».

Лабораторная работа 14 «Адаптеры».

Лабораторная работа 15 «Запуск Активностей с помощью Намерений».

Лабораторная работа 16 «Работа с настройками и состоянием приложения».

Лабораторная работа 17 «Работа с файлами».

Лабораторная работа 18 «Базы данных в Android».

Лабораторная работа 19 «СУБД SQ Lite».

Лабораторная работа 20 «Работа с БД в Android».

Лабораторная работа 21 «Контент-провайдеры».

Лабораторная работа 22 «Интернет-сервисы».

Лабораторная работа 23 «Широковещательные Приемники».

Лабораторная работа 24 «Broadcast».

4 Контрольно-оценочные материалы для итоговой аттестации по профессиональному модулю

Оценка освоения МДК предусматривает проведение дифференцированного зачета.
Оценка освоения ПМ предусматривает проведение экзамена.

4.1 Методические материалы, определяющие процедуры оценивания знаний, умений, практического опыта, характеризующие этапы формирования компетенций

Вопросы к экзамену по дисциплине:

- 1) Программы и программное обеспечение.
- 2) Системное программирование.
- 3) Этапы подготовки программы.
- 4) Программирование на языке Ассемблера.
- 5) Предложения языка Ассемблера.
- 6) Регистры.
- 7) Биты и байты ASCII.
- 8) Двоичные числа.
- 9) Шестнадцатеричное представление.
- 10) Сегменты.
- 11) Расширение набора команд.
- 12) Способы адресации.
- 13) Директивы.
- 14) Директивы определения данных.
- 15) Директива определения байтов (DB).
- 16) Директива определения слова (DW).
- 17) Сегментные регистры: CS, DS, SS и ES.
- 18) Регистры общего назначения: AX, BX, CX и DX.
- 19) Регистровые указатели: SP и BP.
- 20) Индексные регистры: SI и DI.
- 21) Регистр командного указателя: IP.
- 22) Флаговый регистр.
- 23) Обработка двоичных данных.
- 24) Беззнаковые и знаковые данные.
- 25) Умножение.
- 26) Сдвиг регистровой пары DX:AX.
- 27) Деление.
- 28) Преобразование знака.
- 29) Обработка данных в форматах ASCII и BCD.
- 30) Двоично-десятичный формат (BCD).
- 31) Сдвиг и округление.
- 32) Свойства операций над строками.
- 33) REP: Префикс повторения цепочечной команды.
- 34) MOVS: Пересылка строк.
- 35) LODS: Загрузка строки.
- 36) STOS: Запись строки.
- 37) CMPS: Сравнение строк.
- 38) SCAS: Сканирование строк.
- 39) Сканирование и замена.
- 40) Альтернативное кодирование.

- 41) Дублирование образца.
- 42) Определение таблиц.
- 43) Прямой табличный доступ.
- 44) Табличный поиск.
- 45) Транслирующая команда XLAT.
- 46) Операторы типа, длина и размеры.
- 47) Комментарии в программах на Ассемблере.
- 48) Формат кодирования.
- 49) Директивы.
- 50) Память и регистры.
- 51) Инициализация программы
- 52) Ввод программы.
- 53) Подготовка программы для выполнения.
- 54) Ассемблирование программы.
- 55) Компоновка программы.
- 56) Выполнение программы.
- 57) Файл перекрестных ссылок.
- 58) Двухпроходной Ассемблер – первый проход.
- 59) Структура таблиц Ассемблера.
- 60) Двухпроходной Ассемблер – второй проход.
- 61) Некоторые дополнительные директивы.
- 62) Директивы связывания.
- 63) Одно- и многопроходной Ассемблер.
- 64) Модульное программирование.
- 65) Одномодульное программирование: использование функций, классификация параметров, составление функций с переменным списком параметров.
- 66) Составление рекурсивных функций на Ассемблере.
- 67) Отладка, тестирование и оптимизация программ на языке Ассемблера.
- 68) Просмотр и исполнение кода программы.
- 69) Внесение в тело программы точек выхода, которые останавливают исполнение программы.
- 70) Проверка и изменение в памяти значений переменных. Ассемблирование и реассемблирование кода.
- 71) Требования к программному проекту.
- 72) Требования к оформлению программной документации.
- 73) Составление эскизного проекта.
- 74) Технологии прикладного программирования: цели, задачи и основные принципы и инструменты.
- 75) Алгоритмическая и объектно-ориентированная декомпозиция.
- 76) Принципы объектно-ориентированного анализа.
- 77) Объектно-ориентированное проектирование.
- 78) Простой класс.
- 79) Классы и объекты.
- 80) Определение класса.
- 81) Инкопсуляция.
- 82) Простой класс.
- 83) Использование класса.
- 84) Вызов методов класса.
- 85) Объекты программы и объекты реального мира.
- 86) Класс как тип данных.
- 87) Конструктор.
- 88) Полный конструктор.

- 89) Конструктор по умолчанию.
- 90) Объекты в качестве аргументов функции.
- 91) Деструктор.
- 92) Конструктор копирования по умолчанию.
- 93) Статические данные класса.
- 94) Раздельное объявление и определение полей класса.
- 95) Константные методы.
- 96) Полиморфизм.
- 97) Перегрузка функций.
- 98) Перегрузка унарных операций.
- 99) Перегрузка бинарных операций.
- 100) Особые случаи перегрузки операций.
- 101) Преобразование типов.
- 102) Параметрический полиморфизм.
- 103) Шаблоны функций.
- 104) Шаблоны классов.
- 105) Наследование.
- 106) Базовый и производственные классы.
- 107) Множественное наследование.
- 108) Состав стандартной библиотеки C++.
- 109) Контейнеры.
- 110) Вектор.
- 111) Список.
- 112) Итераторы и алгоритмы.
- 113) Очереди.
- 114) Стек.
- 115) Ассоциативные контейнеры.
- 116) Объекты функции и предикаты.
- 117) Интерфейс пользователя.
- 118) Основные понятия.
- 119) Стандартизация пользовательского интерфейса.
- 120) Интерфейс типа «Вопрос-ответ».
- 121) Интерфейс командной строки.
- 122) Текстовый интерфейс.
- 123) Оконный интерфейс.
- 124) Графический оконный интерфейс.
- 125) Web – интерфейс.
- 126) Социальный интерфейс.
- 127) Современный графический пользовательский интерфейс.
- 128) Взаимодействие пользователя с программами.
- 129) Графический пользовательский интерфейс и его реализация в определенной системе Windows.
- 130) Основной объект интерфейса: окно и его основные части.

4.2 Контрольно-оценочные материалы для итоговой аттестации по учебному модулю «Разработка программных модулей программного обеспечения для компьютерных систем»

I. ПАСПОРТ

Назначение:

КОС (контрольно-оценочные средства) предназначен для контроля и оценки результатов освоения учебного ПМ 01 Разработка программных модулей программного

обеспечения для компьютерных систем по специальности СПО 09.02.03 «Программирование в компьютерных системах».

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

ОК1 - Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК2 - Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК3 - Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК4 - Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК5 - Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК6 - Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.

ОК7 - Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК8 - Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК9 - Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Профессиональные компетенции

ПК1.1 – выполнять разработку спецификаций отдельных компонент;

ПК1.2 – осуществлять разработку программного продукта на основе готовых спецификаций на уровне модуля;

ПК1.3 – выполнять отладку программных модулей с использованием специализированных программных средств;

ПК1.4 – выполнять тестирование программных модулей;

ПК1.5 – осуществлять оптимизацию программного кода модуля;

ПК1.6 – разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

II. ЗАДАНИЕ ДЛЯ ЭКЗАМЕНУЮЩЕГОСЯ. Вариант № 1

Вариант 1

Инструкция для обучающихся

Внимательно прочитайте задание.

Время выполнения задания – 0,75 часа

Задание

Теоретическая часть:

- 1) Что такое программы и программное обеспечение?
- 2) Опишите понятие Системное программирование.
- 3) Какие этапы подготовки программы вы знаете?
- 4) Как происходит программирование на языке Ассемблера?
- 5) Как записываются предложения языка Ассемблера?
- 6) Что такое регистры?

III. ПАКЕТ ЭКЗАМЕНАТОРА

III а. УСЛОВИЯ

Количество вариантов задания для экзаменуемого – 26.

Время выполнения задания – 1, 5 часа.

Оборудование:

Компьютер.

Программное обеспечение:

- операционная система;
- файловый менеджер (в составе операционной системы или др.);
- электронные средства образовательного назначения:
 - Microsoft® Office Professional Plus 2007 Russian;
 - Microsoft® Office Standard 2013 Russian.

Эталоны ответов на теоретические вопросы:

1) Программа - это данные, предназначенные для управления конкретными компонентами системы обработки информации (СОИ) в целях реализации определенного алгоритма.

Обратить внимание: программа — это данные.

Один из основных принципов машины фон Неймана - то, что и программы, и данные хранятся в одной и той же памяти. Сохраняемая в памяти программа представляет собой некоторые коды, которые могут рассматриваться как данные. Возможно, с точки зрения программиста программа - активный компонент, она выполняет некоторые действия. Но с точки зрения процессора команды программы - это данные, которые процессор читает и интерпретирует. С другой стороны программа - это данные с точки зрения обслуживающих программ, например, с точки зрения компилятора, который на входе получает одни данные — программу на языке высокого уровня (ЯВУ), а на выходе выдает другие данные — программу в машинных кодах.

Программное обеспечение (ПО) - совокупность программ СОИ и программных документов, необходимых для их эксплуатации

Существенно, что ПО - это программы, предназначенные для многократного использования и применения разными пользователями. В связи с этим следует обратить внимание на ряд необходимых свойств ПО.

а) Необходимость документирования

По определению программы становятся ПО только при наличии документации. Конечный пользователь не может работать, не имея документации. Документация делает возможным тиражирование ПО и продажу его без его разработчика. По Бруксу ошибкой в ПО является ситуация, когда программное изделие функционирует не в соответствии со своим описанием, следовательно, ошибка в документации также является ошибкой в программном изделии.

б) Эффективность

ПО, рассчитанное на многократное использование (например, ОС, текстовый редактор) пишется и отлаживается один раз, а выполняется многократно. Таким образом, выгодно переносить затраты на этап производства ПО и освободить от затрат этап выполнения, чтобы избежать тиражирования затрат.

в) Надежность

В том числе:

- тестирование программы при всех допустимых спецификациях входных данных
- защита от неправильных действий пользователя
- защита от взлома — пользователи должны иметь возможность взаимодействия с ПО только через легальные интерфейсы.

Появление ошибок любого уровня не должно приводить к краху системы. Ошибки должны вылавливаться диагностироваться и (если их невозможно исправить) превращаться в корректные отказы.

Системные структуры данных должны сохраняться безусловно.

Сохранение целостности пользовательских данных желательно.

Возможность сопровождения

Возможные цели сопровождения — адаптация ПО к конкретным условиям применения, устранение ошибок, модификация.

Во всех случаях требуется тщательное структурирование ПО и носителем информации о структуре ПО должна быть программная документация.

Адаптация во многих случаях может быть передоверена пользователю — при тщательной отработке и описании сценариев инсталляции и настройки.

Исправление ошибок требует развитой сервисной службы, собирающей информацию об ошибках и формирующей исправляющие пакеты.

Модификация предполагает изменение спецификаций на ПО. При этом, как правило, должны поддерживаться и старые спецификации. Эволюционное развитие ПО экономит вложения пользователей.

2) Системная программа - программа, предназначенная для поддержания работоспособности СООИ или повышения эффективности ее использования.

Прикладная программа - программа, предназначенная для решения задачи или класса задач в определенной области применения СООИ.

В соответствии с терминологией, системное программирование — это процесс разработки системных программ (в том числе, управляющих и обслуживающих).

С другой стороны, система - единое целое, состоящее из множества компонентов и множества связей между ними. Тогда системное программирование - это разработка программ сложной структуры.

Эти два определения не противоречат друг другу, так как разработка программ сложной структуры ведется именно для обеспечения работоспособности или повышения эффективности СООИ.

Подразделение ПО на системное и прикладное является до некоторой степени устаревшим. Сегодняшнее деление предусматривает по меньшей мере три градации ПО:

- системное
- промежуточное
- прикладное

Промежуточное ПО (middleware) мы определяем как совокупность программ, осуществляющих управление вторичными (конструируемыми самим ПО) ресурсами, ориентированными на решение определенного (широкого) класса задач. К такому ПО относятся менеджеры транзакций, серверы БД, серверы коммуникаций и другие программные серверы. С точки зрения инструментальных средств разработки промежуточное ПО ближе к прикладному, так как не работает на прямую с первичными ресурсами, а использует для этого сервисы, предоставляемые системным ПО.

С точки зрения алгоритмов и технологий разработки промежуточное ПО ближе к системному, так как всегда является сложным программным изделием многократного и многоцелевого использования и в нем применяются те же или сходные алгоритмы, что и в системном ПО.

Современные тенденции развития ПО состоит в снижении объема как системного, так и прикладного программирования. Основная часть работы программистов выполняется в промежуточном ПО. Снижение объема системного программирования определено

современными концепциями ОС, объектно-ориентированной архитектурой и архитектурой микроядра, в соответствии с которыми большая часть функций системы выносятся в утилиты, которые можно отнести и к промежуточному ПО. Снижение объема прикладного программирования обусловлено тем, что современные продукты промежуточного ПО предлагают все больший набор инструментальных средств и шаблонов для решения задач своего класса.

Значительная часть системного и практически все прикладное ПО пишется на языках высокого уровня, что обеспечивает сокращение расходов на их разработку/модификацию и переносимость.

Системное ПО подразделяется на системные управляющие программы и системные обслуживающие программы.

Управляющая программа - системная программа, реализующая набор функций управления, который включает в себя управление ресурсами и взаимодействие с внешней средой СОО, восстановление работы системы после проявления неисправностей в технических средствах.

Программа обслуживания (утилита) - программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу СОО.

Управляющая программа совместно с набором необходимых для эксплуатации системы утилит составляют операционную систему(ОС).

Кроме входящих в состав ОС утилит могут существовать и другие утилиты (того же или стороннего производителя), выполняющие дополнительное (опционное) обслуживание. Как правило, это утилиты, обеспечивающие разработку программного обеспечения для операционной системы.

Система программирования - система, образуемая языком программирования, компилятором или интерпретатором программ, представленных на этом языке, соответствующей документацией, а также вспомогательными средствами для подготовки программ к форме, пригодной для выполнения.

3) При разработке программ, а тем более - сложных, используется принцип модульности, разбиения сложной программы на составные части, каждая из которых может подготавливаться отдельно. Модульность является основным инструментом структурирования программного изделия, облегчающим его разработку, отладку и сопровождение.

Программный модуль - программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память.

При выборе модульной структуры должны учитываться следующие основные соображения:

- а) Функциональность - модуль должен выполнять законченную функцию
- б) Несвязность - модуль должен иметь минимум связей с другими модулями, связь через глобальные переменные и области памяти нежелательна
- в) Специфицируемость - входные и выходные параметры модуля должны четко формулироваться

Программа пишется в виде исходного модуля.

Исходный модуль - программный модуль на исходном языке, обрабатываемый транслятором и представляемый для него как целое, достаточное для проведения трансляции.

Первым (не для всех языков программирования обязательным) этапом подготовки программы является обработка ее Макропроцессором (или Препроцессором). Макропроцессор обрабатывает текст программы и на выходе его получается новая редакция текста. В большинстве систем программирования Макропроцессор совмещен с транслятором, и для программиста его работа и промежуточный ИМ «не видны».

Следует иметь в виду, что Макропроцессор выполняет обработку текста, это означает, с одной стороны, что он «не понимает» операторов языка программирования и «не

знает» переменных программы, с другой, что все операторы и переменные Макроязыка (тех выражений в программе, которые адресованы Макропроцессору) в промежуточном ИМ уже отсутствуют и для дальнейших этапов обработки «не видны».

Так, если Макропроцессор заменил в программе некоторый текст А на текст В, то транслятор уже видит только текст В, и не знает, был этот текст написан программистом «своей рукой» или подставлен Макропроцессором.

Следующим этапом является трансляция.

Трансляция - преобразование программы, представленной на одном языке программирования, в программу на другом языке программирования, в определенном смысле равносильную первой.

Как правило, выходным языком транслятора является машинный язык целевой вычислительной системы. (Целевая ВС - та ВС, на которой программа будет выполняться.)

Машинный язык - язык программирования, предназначенный для представления программы в форме, позволяющей выполнять ее непосредственно техническими средствами обработки информации.

Трансляторы - общее название для программ, осуществляющих трансляцию. Они подразделяются на Ассемблеры и Компиляторы - в зависимости от исходного языка программы, которую они обрабатывают. Ассемблеры работают с Автокодами или языками Ассемблера, Компиляторы - с языками высокого уровня.

Автокод - символьный язык программирования, предложения которого по своей структуре в основном подобны командам и обрабатываемым данным конкретного машинного языка.

Язык Ассемблера - язык программирования, который представляет собой символьную форму машинного языка с рядом возможностей, характерных для языка высокого уровня (обычно включает в себя макросредства).

Язык высокого уровня - язык программирования, понятия и структура которого удобны для восприятия человеком.

Объектный модуль - программный модуль, получаемый в результате трансляции исходного модуля.

Поскольку результатом трансляции является модуль на языке, близком к машинному, в нем уже не остается признаков того, на каком исходном языке был написан программный модуль. Это создает принципиальную возможность создавать программы из модулей, написанных на разных языках. Специфика исходного языка, однако, может сказываться на физическом представлении базовых типов данных, способах обращения к процедурам/функциям и т.п. Для совместимости разноязыковых модулей должны выдерживаться общие соглашения. Большая часть объектного модуля - команды и данные машинного языка именно в той форме, в какой они будут существовать во время выполнения программы. Однако, программа в общем случае состоит из многих модулей. Поскольку транслятор обрабатывает только один конкретный модуль, он не может должным образом обработать те части этого модуля, в которых запрограммированы обращения к данным или процедурам, определенным в другом модуле. Такие обращения называются внешними ссылками. Те места в объектном модуле, где содержатся внешние ссылки, транслируются в некоторую промежуточную форму, подлежащую дальнейшей обработке. Говорят, что объектный модуль представляет собой программу на машинном языке с неразрешенными внешними ссылками. Разрешение внешних ссылок выполняется на следующем этапе подготовки, который обеспечивается Редактором Связей (Компоновщиком). Редактор Связей соединяет вместе все объектные модули, входящие в программу. Поскольку Редактор Связей «видит» уже все компоненты программы, он имеет возможность обработать те места в объектных модулях, которые содержат внешние ссылки. Результатом работы Редактора Связей является загрузочный модуль.

Загрузочный модуль - программный модуль, представленный в форме, пригодной для загрузки в оперативную память для выполнения.

Загрузочный модуль сохраняется в виде файла на внешней памяти. Для выполнения

программа должна быть перенесена (загружена) в оперативную память. Иногда при этом требуется некоторая дополнительная обработка (например, настройка адресов в программе на ту область оперативной памяти, в которую программа загрузилась). Эта функция выполняется Загрузчиком, который обычно входит в состав операционной системы. Возможен также вариант, в котором редактирование связей выполняется при каждом запуске программы на выполнение и совмещается с загрузкой. Это делает Связывающий Загрузчик. Вариант связывания при запуске более расходуемый, т.к. затраты на связывание тиражируются при каждом запуске. Но он обеспечивает:

- большую гибкость в сопровождении, так как позволяет менять отдельные объектные модули программы, не меняя остальных модулей;
- экономию внешней памяти, т.к. объектные модули, используемые во многих программах не копируются в каждый загрузочный модуль, а хранятся в одном экземпляре.

Вариант интерпретации подразумевает прямое исполнение исходного модуля.

Интерпретация - реализация смысла некоторого синтаксически законченного текста, представленного на конкретном языке.

Интерпретатор читает из исходного модуля очередное предложение программы, переводит его в машинный язык и выполняет. Все затраты на подготовку тиражируются при каждом выполнении, следовательно, интерпретируемая программа принципиально менее эффективна, чем транслируемая. Однако, интерпретация обеспечивает удобство разработки, гибкость в сопровождении и переносимость.

Не обязательно подготовка программы должна вестись на той же вычислительной системе и в той же операционной среде, в которых программа будет выполняться. Системы, обеспечивающие подготовку программ в среде, отличной от целевой называются кросс-системами. В кросс-системе может выполняться вся подготовка или ее отдельные этапы:

- а) Макрообработка и трансляция
- б) Редактирование связей
- в) Отладка

Типовое применение кросс-систем — для тех случаев, когда целевая вычислительная среда просто не имеет ресурсов, необходимых для подготовки программ, например, встроенные системы. Программные средства, обеспечивающие отладку программы на целевой системе можно также рассматривать как частный случай кросс-системы.

4) Язык Ассемблера - система записи программы с детализацией до отдельной машинной команды, позволяющая использовать мнемоническое обозначение команд и символическое задание адресов.

Поскольку в разных аппаратных архитектурах разные программно-доступные компоненты (система команд, регистры, способы адресации), язык Ассемблера аппаратно-зависимый. Программы, написанные на языке Ассемблера могут быть перенесены только на вычислительную систему той же архитектуры.

Программирование на языке Ассемблера позволяет в максимальной степени использовать особенности архитектуры вычислительной системы. До недавнего времени воспринималась как аксиома, что ассемблерная программа всегда является более эффективной и в смысле быстродействия, и в смысле требований к памяти. Для Intel-архитектуры это и сейчас так.

Но это уже не так для RISK-архитектур. Для того, чтобы программа могла эффективно выполняться в вычислительной среде с распараллеливанием на уровне команд, она должна быть определенным образом оптимизирована, то есть, команды должны быть расположены в определенном порядке, допускающим их параллельное выполнение. Программист просто не сможет покомандно оптимизировать всю свою программу. С задачей такой оптимизации более эффективно справляются компиляторы.

Доля программ, которые пишутся на языках Ассемблеров в мире, неуклонно уменьшается, прикладное программирование на языках Ассемблеров применяется только по недомыслию. Язык Ассемблера «в чистом виде» применяется только для написания отдельных неболь-

ших частей системного ПО: микроядра ОС, самых нижних уровней драйверов - тех частей, которые непосредственно взаимодействуют с реальными аппаратными компонентами.

Этим занимается узкий круг программистов, работающих в фирмах, производящих аппаратуру и ОС. Зачем же нам тогда изучать построение Ассемблера?

Хотя разработка программ, взаимодействующих с реальными аппаратными компонентами, - редкая задача, в современном программировании при разработке прикладного, а еще более - промежуточного ПО довольно часто применяется технологии виртуальных машин. Для выполнения того или иного класса задач программно моделируется некоторое виртуальное вычислительное устройство, функции которого соответствуют нуждам этого класса задач.

Для управления таким устройством для него может быть создан соответствующий язык команд. (Широко известные примеры: MI AS/400, JVM.) Говоря шире, любую программу можно представить себе как виртуальное «железо», решающее конкретную задачу. (Конечный пользователь обычно не видит разницы между программой и аппаратурой и часто говорит не «мне программа выдала то-то», а «мне компьютер выдал то-то»). В некоторых случаях интерфейс программы удобно представить в виде системы команд, а следовательно, нужен соответствующий Ассемблер. (Это, конечно, относится не к программам «для чайников», а к инструментальным средствам программистов, системам моделирования).

5) Предложения языка Ассемблера описывают команды или псевдокоманды (директивы). Предложения-команды задают машинные команды вычислительной системы; обработка Ассемблером команды приводит к генерации машинного кода. Обработка псевдокоманды не приводит к непосредственной генерации кода, псевдокоманда управляет работой самого Ассемблера. Для одной и той же аппаратной архитектуры могут быть построены разные Ассемблеры, в которых команды будут обязательно одинаковые, но псевдокоманды могут быть разные.

Во всех языках Ассемблеров каждое новое предложение языка начинается с новой строки. Каждое предложение, как правило, занимает одну строку, хотя обычно допускается продолжение на следующей строке/строках. Формат записи предложений языка м.б. жесткий или свободный. При записи в жестком формате составляющие предложения должны располагаться в фиксированных позициях строки. (Например: метка должна располагаться в позициях 1-8, позиция 9 - пустая, позиции 10-12 - мнемоника команды, позиция 13 - пустая, начиная с позиции 14 - операнды, позиция 72 - признак продолжения). Обычно для записи программ при жестком формате создаются бланки. Жесткий формат удобен для обработки Ассемблером (удобен и для чтения).

Свободный формат допускает любое количество пробелов между составляющими предложения.

В общих случаях предложения языка Ассемблера состоят из следующих компонент:

- а) метка или имя;
- б) мнемоника;
- в) операнды;
- г) комментарии.

Метка или имя является необязательным компонентом. Не во всех языках Ассемблеров эти понятия различаются. Если они различаются (например, MASM), то метка - точка программы, на которую передается управление, следовательно, метка стоит в предложении, содержащем команду; имя - имя переменной программы, ячейки памяти, следовательно, имя стоит в предложении, содержащем псевдокоманду резервирования памяти или определения константы. В некоторых случаях метка и имя могут отличаться даже синтаксически, так, в MASM/ TASM после метки ставится двоеточие, а после имени - нет.

Однако, физический смысл и метки, и имени - одинаков, это - адрес памяти. Во всех случаях, когда Ассемблер встречает в программе имя или метку, он заменяет ее на адрес той ячейки памяти, к которую имя/метка именуется.

Правила формирования имен/меток совпадают с таковыми для языков программирования. В некоторых Ассемблерах (HLAM S/390) не делается различия между меткой и именем.

В языке должны предусматриваться некоторые специальные правила, позволяющие Ассемблеру распознать и выделить метку/имя, например:

а) метка/имя должна начинаться в 1-й позиции строки

б) если метки/имени нет, то в 1-й позиции должен быть пробел, или за меткой/именем должно следовать двоеточие, и т.п.

Мнемоника - символическое обозначение команды/псевдокоманды.

Операнды - один или несколько операндов, обычно разделяемые запятыми. Операндами команд являются имена регистров, непосредственные операнды, адреса памяти (задаваемые в виде констант, литералов, символических имен или сложных выражений, включающих специальный синтаксис). Операнды псевдокоманд могут быть сложнее и разнообразнее.

Комментарии - любой текст, который игнорируется Ассемблером. Комментарии располагаются в конце предложения и отделяются от текста предложения, обрабатываемого Ассемблером, каким-либо специальным символом (в некоторых языках - пробелом). Всегда предусматривается возможность строк, содержащих только комментариев, обычно такие строки содержат специальный символ в 1-й позиции.

Константы - могут представлять непосредственные операнды или абсолютные адреса памяти. Применяются 10-е, 8-е, 16-е, 2-е, символьные константы.

Непосредственные операнды - записываются в сам код команды.

Имена - адреса ячеек памяти.

При трансляции Ассемблер преобразует имена в адреса. Способ преобразования имени в значение зависит от принятых способов адресации. Как правило, в основном способом адресации в машинных языках является адресация относительная: адрес в команде задается в виде смещения относительно какого-то базового адреса, значение которого содержится в некотором базовом регистре. В качестве базового могут применяться либо специальные регистры (DS, CS в Intel) или регистры общего назначения (S/390).

Литералы - записанные в особой форме константы. Концептуально литералы - те же имена. При появлении в программе литерала Ассемблер выделяет ячейку памяти и записывает в нее заданную в литерале константу. Далее все появления этого литерала Ассемблер заменяет на обращения по адресу этой ячейки. Таким образом, литеральные константы, хранятся в памяти в одном экземпляре, независимо от числа обращений к ним.

Специальный синтаксис - явное описание способа адресации (например, указание базового регистра и смещения).

б) Программа в машинном коде состоит из различных сегментов для определения данных, для машинных команд и для сегмента, названного стеком, для хранения адресов. Для выполнения арифметических действий, пересылки данных и адресации компьютер имеет ряд регистров.

Для выполнения программ компьютер временно записывает программу и данные в основную память. Компьютер имеет также ряд регистров, которые он использует для временных вычислений.

Экзаменационная ведомость.

ШБ. КРИТЕРИИ ОЦЕНКИ

Критерии оценки:

Отметка «5» - выставляется в том случае, когда в ответе обучающегося полно и верно раскрыто основное содержание вопроса, соблюдена логическая последовательность элементов ответа; общие положения конкретизируются фактами, обосновываются аргументами.

Отметка «4» - выставляется в том случае, когда в ответе обучающегося содержится верное освещение темы вопроса, но отсутствует полнота раскрытия; соблюдена логика изложения.

Отметка «3» - выставляется в том случае, когда в ответе обучающегося приведены отдельные несистематизированные положения, отсутствует конкретизация их фактами или частично приведены отдельные верные факты.

Образец экзаменационного билета

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

20__ / 20__ учебный год

Экзамен
по ПМ 01 Разработка программных модулей программного
обеспечения для компьютерных систем
Билет № 1

ЗАДАНИЕ ДЛЯ ЭКЗАМЕНУЮЩЕГОСЯ

Оцениваемые компетенции: ОК1-9, ПК1.1-1.6

Вариант № 1

Теоретическая часть:

- 1) Что такое программы и программное обеспечение?
- 2) Опишите понятие Системное программирование.
- 3) Какие этапы подготовки программы вы знаете?
- 4) Как происходит программирование на языке Ассемблера?
- 5) Как записываются предложения языка Ассемблера?
- 6) Что такое регистры?

Вы можете воспользоваться конспектом лекций.

Максимальное время выполнения задания – 90 мин.

Утвержден на заседании кафедры «_____»,
протокол № ___ от «__» _____ 20__ г.

Зав. кафедрой «_____» _____ В.А. Тихомиров
(подпись)

5 Критерии и шкалы для интегрированной оценки уровня сформированности компетенций:

Индикаторы компетенции	неудовлетворительно	удовлетворительно	хорошо	отлично
Полнота знаний	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущено много негрубых ошибок.	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок.
Наличие умений	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки.	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания но не в полном объеме.	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения, решены все основные задачи с отдельными незначительными недочетами, выполнены все задания в полном объеме.
Характеристика сформированности компетенции	Компетенция в полной мере не сформирована. Имеющихся знаний, умений, навыков недостаточно для решения практических (профессиональных) задач. Требуется повторное обучение	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач, но требуется дополнительная практика по большинству практических задач.	Сформированность компетенции в целом соответствует требованиям, но есть недочеты. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения практических (профессиональных) задач, но требуется дополнительная практика по некоторым профессиональным задачам.	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для решения сложных практических (профессиональных) задач.
Уровень сформированности компетенций	Низкий	Ниже среднего	Средний	Высокий

6 Список рекомендуемой литературы и источников информации

Основная литература:

1 Партыка, Т. Л. Операционные системы, среды и оболочки : учебное пособие / Т. Л. Партыка, И. И. Попов. – 5-е изд., перераб. и доп. – Москва : ФОРУМ : ИНФРА-М, 2021. – 560 с. – (Среднее профессиональное образование). – ISBN 978-5-00091-501-1. – Текст : электронный. – URL: <https://znanium.com/catalog/product/1189335>. – Режим доступа: по подписке.

2 Рудаков, А. В. Операционные системы и среды : учебник / А.В. Рудаков. — Москва : КУРС : ИНФРА-М, 2022. — 304 с. — (Среднее профессиональное образование). - ISBN 978-5-906923-85-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1843025>. – Режим доступа: по подписке.

3 Операционные системы. Основы UNIX : учебное пособие / А.Б. Вавренюк, О.К. Курышева, С.В. Кутепов, В.В. Макаров. — Москва : ИНФРА-М, 2021. — 160 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-013981-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189336>. – Режим доступа: по подписке.

4 Тихомиров, В.А. Операционные системы и сети: лабораторный практикум / В.А. Тихомиров. - Комсомольск-на-Амуре: Изд-во Комсомольского-на-Амуре гос. техн. ун-та, 2016. - 117с.

5 Системное программное обеспечение: лабораторный практикум / С.П. Черный, С.А. Васильченко, А.С. Гудим, В.И. Суздорф. - Комсомольск-на-Амуре: Изд-во Комсомольского-на-Амуре гос. техн. ун-та, 2016. - 90с

6 Максимов, Н. В. Архитектура ЭВМ и вычислительных систем : учебник / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. — 5-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2022. — 511 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-511-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1856720>. – Режим доступа: по подписке.

7 Колдаев, В. Д. Архитектура ЭВМ : учебное пособие / В.Д. Колдаев, С.А. Лупин. — Москва : ФОРУМ : ИНФРА-М, 2021. — 383 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0868-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1136788>. – Режим доступа: по подписке.

8 Степина, В. В. Архитектура ЭВМ и вычислительные системы : учебник / В.В. Степина. — Москва : КУРС : ИНФРА-М, 2021. — 384 с. — (Среднее профессиональное образование). - ISBN 978-5-906923-07-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1423169>. – Режим доступа: по подписке.

9 Заславская, О.Ю. Архитектура компьютера : лекции, лабораторные работы, комментарии к выполнению. Учебно-методическое пособие для сред. проф. образования / О.Ю. Заславская. – М.: Московский городской

педагогический университет, 2013. – 148 с. – Режим доступа: <http://www.iprbookshop.ru/26450.html>, ограниченный. – Загл. с экрана.

10 Костюкова Н.И. Программирование на языке Си : методические рекомендации и задачи по программированию / Н.И. Костюкова. — Электрон. текстовые данные. — Новосибирск: Сибирское университетское издательство, 2017. — 160 с. //IPRbooks: электронно-библиотечная система. – Режим доступа: <http://www.iprbookshop.ru/65289.html>, ограниченный. – Загл. с экрана.

11 Голицына, О. Л. Языки программирования : учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 3-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 399 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-613-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1209231>. – Режим доступа: по подписке.

12 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1172261>. – Режим доступа: по подписке.

13 Кузин, А. В. Программирование на языке Си : учебное пособие / А.В. Кузин, Е.В. Чумакова. — Москва : ФОРУМ : ИНФРА-М, 2021. — 144 с. — (Высшее образование). - ISBN 978-5-00091-066-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1222078>. – Режим доступа: по подписке.

14 Белева, Л. Ф. Программирование на языке C++ : учебное пособие для сред. проф. образования / Л.Ф. Белева. – Саратов: Ай Пи Эр Медиа, 2018. – 81 с. //IPRbooks: электронно-библиотечная система. – Режим доступа: <http://www.iprbookshop.ru/72466.html>, ограниченный. – Загл. с экрана.

15 Тихомиров, В.А. Разработка простейших приложений для мобильных устройств: учебное пособие / В.А. Тихомиров. - Комсомольск-на-Амуре: Изд-во Комсомольского-на-Амуре гос. техн. ун-та, 2013. – 133 с.

Дополнительная литература:

1 Assembler: Практикум. 2-е изд. / В.И. Юров. – СПб.: Питер, 2006 г.

2 Assembler для DOS, Windows и UNIX. - 3-е изд., стер. / С.В. Зубков. - СПб: Питер, 2004 г.

3 Assembler: Учебник для вузов / В.И. Юров. – СПб.: Питер, 2010 г.

4 Аляев Ю., Козлов О. Алгоритмизация и языки программирования Pascal, C+, Visual Basic. – М.: Финансы и статистика, 2003.

5 Гуриков, С. Р. Введение в программирование на языке Visual C# : учебное пособие / С. Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2019. — 447 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-540-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1012397>. – Режим доступа: по подписке.

6 Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ :

ИНФРА-М, 2022. — 200 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-713-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1873259>. - Режим доступа: по подписке.

7 Канцедал, С. А. Алгоритмизация и программирование : учебное пособие / С. А. Канцедал. — Москва : ФОРУМ : ИНФРА-М, 2021. — 352 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0727-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189320>. - Режим доступа: по подписке.

8 Шакин, В. Н. Базовые средства программирования на Visual Basic в среде Visual Studio.NET. Практикум : учебное пособие / В.Н. Шакин. — Москва : ФОРУМ : ИНФРА-М, 2021. — 287 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-565-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1241807>. - Режим доступа: по подписке.

9 Шакин, В. Н. Базовые средства программирования на Visual Basic в среде Visual Studio .NET : учебное пособие / В. Н. Шакин. — Москва : ФОРУМ : ИНФРА-М, 2019. — 303 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-564-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/961507>. - Режим доступа: по подписке.

10 Шакин, В. Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio.NET : учебное пособие / В.Н. Шакин, А.В. Загвоздкина, Г.К. Сосновииков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 398 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-551-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1082462>. - Режим доступа: по подписке.

11 Зак Д. Самоучитель Visual Basic.Net. – Киев.: ВНУ; СПб.: Питер, 2003.

12 Кетков Ю., Кетков А. Практика программирования: Visual Basic, C++Builder. – СПб.: ВНУ, 2002.

13 Семакин И.Г., Шестаков А.П. Основы программирования: Учебник. – М.: Мастерство, 2002. – 432 с.

14 Использование Turbo Assembler при разработке программ. Киев, Диалектика, 2003.

15 П.И. Рудаков, К.Г. Финогенов. Програмуємо на языкe асемблера IBM PC. Обнинск, 2005.

16 Visual Basic 6.0: Пер. с англ. – СПб.: БХВ - Петербург, 2003. – 992 с.: илл.

17 Ткаченко О.Н. Взаимодействие пользователей с интерфейсами информационных систем для мобильных устройств: исследование опыта: учебное пособие / Ткаченко О.Н. - М.: Магистр: ИНФРА-М, 2018. — 152 с.

18 Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2022. — 158 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015447-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1860121>. - Режим доступа: по подписке.

19 Лисьев, Г. А. Программное обеспечение компьютерных сетей и web-

серверов : учебное пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. — Москва : ИНФРА-М, 2021. — 145 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014514-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189343>. – Режим доступа: по подписке.

20 Введение в разработку приложений для ОС Android / Ю.В. Березовская [и др.]. — 2-е изд. – М.: ИНТУИТ, 2021. — 433 с. //IPRbooks: электронно-библиотечная система. – Режим доступа: <http://www.iprbookshop.ru/73669.html>, ограниченный. – Загл. с экрана.

21 Кузьмичёв, А.Э. Программирование для Windows Phone для начинающих / А.Э. Кузьмичёв. – М.: ИНТУИТ, Ай Пи Эр Медиа, 2019. – 165 с. //IPRbooks: электронно-библиотечная система. – Режим доступа: <http://www.iprbookshop.ru/79729.html>, ограниченный. – Загл. с экрана.

22 Семакова, А. Введение в разработку приложений для смартфонов на ОС Android / А. Семакова. — 2-е изд. – М.: ИНТУИТ, 2021. – 102 с. //IPRbooks: электронно-библиотечная система. – Режим доступа: <http://www.iprbookshop.ru/73670.html>, ограниченный. – Загл. с экрана.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»:

1 intuit.ru : Национальный открытый университет : сайт. – Москва, 2003. – . – URL: <https://www.osp.ru>.

2 edu.ru : Федеральный образовательный портал : сайт. – Москва, 2002. – . – URL: <https://www.edu.ru>.

Периодическая литература:

1 Интеллектуальные системы в производстве [Электронный ресурс]: науч. журн.

2 Информационные технологии и вычислительные системы [Электронный ресурс]: науч. журн.

3 Программирование [Электронный ресурс]: науч. журн.

Современные профессиональные базы данных и информационные справочные системы:

1 Электронно-библиотечная система eLIBRARY.RU (периодические издания) Договор № ЕП 44/3 на оказание услуг доступа к электронным изданиям ИКЗ 211 272 7000769 270 301 001 0010 002 6311 244 от 04 февраля 2021 г.

2 Электронно-библиотечная система ZNANIUM.COM. Договор № 4997 эбс ИКЗ 21 1 2727000769 270301001 0010 004 6311 244 от 13 апреля 2021 г.

3 Электронно-библиотечная система IPRbooks. Лицензионный договор № ЕП 44/4 на предоставление доступа к электронно-библиотечной системе IPRbooks ИКЗ 21 1 2727000769 270301001 0010 003 6311 244 от 05 февраля 2021 г.

4 Электронно-библиотечная система издательства «Академия». Лицензионный договор № 001386/ЭБ-20 ИКЗ 201272700076927030100100240015811244 от 17 июля 2020 г.

Лист изменений и дополнений

в комплект контрольно-оценочных средств учебного модуля «Разработка программных модулей программного обеспечения для компьютерных систем» по специальности 09.02.03 «Программирование в компьютерных системах»

<i>№ изменения, дата изменения; номер страницы с изменением</i>	
БЫЛО	СТАЛО
<ol style="list-style-type: none"> 1. Министерство образования и науки Российской Федерации – стр.1. 2. «Комсомольский-на-Амуре государственный технический университет» - стр.1 	<ol style="list-style-type: none"> 1. Министерство науки и высшего образования Российской Федерации – стр.1. 2. «Комсомольский-на-Амуре государственный университет» - стр.1. 3. Рудаков, А.В. Операционные системы и среды [Электронный ресурс]: учебник / А.В. Рудаков. – М.: КУРС: ИНФРА-М, 2018. – 304 с. //ZNANIUM.COM: электронно-библиотечная система. – Режим доступа: http://www.znanium.com/catalog.php, ограниченный. – Загл. с экрана. – стр.37. 4. Максимов, Н.В. Архитектура ЭВМ и вычислительных систем [Электронный ресурс]: учебник для сред. проф. образования / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. – М.: ФОРУМ: ИНФРА-М, 2018. – 511 с. //ZNANIUM.COM: электронно-библиотечная система. – Режим доступа: http://www.znanium.com/catalog.php, ограниченный. – Загл. с экрана. – стр.37. 5. Колдаев, В.Д. Архитектура ЭВМ [Электронный ресурс]: учеб. пособие для сред. проф. образования / В.Д. Колдаев, С.А. Лупин. – М.: ИД «ФОРУМ»: ИНФРА-М, 2018. – 383 с. //ZNANIUM.COM: электронно-библиотечная система. – Режим доступа: http://www.znanium.com/catalog.php, ограниченный. – Загл. с экрана. – стр.37. 6. Голицына, О.Л. Языки программирования [Электронный ресурс]: учеб. пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. – М.: ФОРУМ: ИНФРА-М, 2018. – 399 с. //ZNANIUM.COM: электронно-библиотечная система. – Режим доступа: http://www.znanium.com/catalog.php, ограниченный. – Загл. с экрана. – стр.37. 7. Немцова, Т.И. Программирование на языке высокого уровня. Программирование на

языке С++ [Электронный ресурс]: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 512 с.

//ZNANIUM.COM: электронно-библиотечная система. — Режим доступа: <http://www.znanium.com/catalog.php>, ограниченный. — Загл. с экрана. — стр.37.

8. Белева, Л. Ф. Программирование на языке С++ [Электронный ресурс]: учебное пособие для сред. проф. образования / Л.Ф. Белева. — Саратов: Ай Пи Эр Медиа, 2018. — 81 с. //IPRbooks: электронно-библиотечная система. — Режим доступа: <http://www.iprbookshop.ru/72466.html>, ограниченный. — Загл. с экрана. — стр.38.

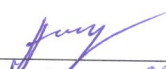
9. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ» : ИНФРА-М, 2018. — 512 с. — стр.38.

10. Ткаченко О.Н. Взаимодействие пользователей с интерфейсами информационных систем для мобильных устройств: исследование опыта: учебное пособие / Ткаченко О.Н. - М.: Магистр: ИНФРА-М, 2018. — 152 с. — стр.38.

11. Кузьмичёв, А.Э. Программирование для Windows Phone для начинающих [Электронный ресурс] / А.Э. Кузьмичёв. — М.: ИНТУИТ, Ай Пи Эр Медиа, 2019. — 165 с. //IPRbooks: электронно-библиотечная система. — Режим доступа: <http://www.iprbookshop.ru/79729.html>, ограниченный. — Загл. с экрана. — стр.38.

Основание:

1. Постановление Правительства РФ от 15.06.2018 №682 «Об утверждении Положения о Министерстве науки и высшего образования Российской Федерации и признании утратившими силу некоторых актов Правительства Российской Федерации.
2. Приказ Минобрнауки России от 3 октября 2017г. № 997 «О переименовании федерального государственного бюджетного образовательного учреждения высшего образования «Комсомольский-на-Амуре государственный технический университет» и его филиала и о внесении изменений в устав федерального государственного бюджетного образовательного учреждения высшего образования «Комсомольский-на-Амуре государственный технический университет»
3. Приобретение новой литературы


подпись 03.09.18

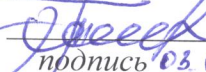
/ О.Н. Попенко

Инициалы, фамилия внесшего изменения

Рассмотрено и одобрено на заседании кафедры «МОП ЭВМ»

Протокол № 10 от 03.09.2018

Зав.каф. «МОП ЭВМ»


подпись 03.09.18

/ В.А. Тихомиров/

Инициалы, фамилия

РЕЦЕНЗИЯ

на комплект оценочных средств по профессиональному модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем»,

составленные преподавателем КнАГТУ Попенко Ольгой Николаевной

Представленный на рецензию комплект оценочных средств по профессиональному модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем» разработан в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования для специальности 09.02.03 «Программирование в компьютерных системах») для очной формы обучения.

В структуре комплекта оценочных средств представлены следующие элементы: паспорт комплекта оценочных средств; комплект оценочных средств для проведения текущего контроля и промежуточной аттестации по междисциплинарным курсам МДК.01.01 «Системное программирование», МДК.01.02 «Прикладное программирование», МДК.01.03 «Программирование для мобильных устройств»; комплект оценочных средств для проведения промежуточной аттестации по профессиональному модулю.

Представленные оценочные средства по ПМ.01 позволяют стимулировать познавательную активность обучающихся за счет разнообразных форм заданий, разного уровня сложности, наличия материалов само- и взаимооценки.

Задания, представленные в комплекте оценочных средств, максимально приближены к условиям будущей профессиональной деятельности обучающихся. В программу квалификационного экзамена для контроля и оценки результатов освоения модуля ПМ.01 включено 130 экзаменационных вопросов.

Представленный комплект оценочных средств, по нашему мнению, позволяет объективно оценить уровень знаний, умений, сформированность практического опыта, общих и профессиональных компетенций обучающихся и их соответствие требованиям ФГОС СПО по данной специальности.


Рецензируемый комплект оценочных средств рекомендуется для использования в качестве диагностического инструментария при реализации профессионального модуля ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем».

Рецензент:

Рецензент: руководитель проекта по реализации ФЦП

ПАО «Амурский судостроительный завод»




В.А. Ханов
« 11 » 05 2017г.

Лист изменений и дополнений

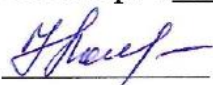
в комплект контрольно-оценочных средств учебного модуля
«Разработка программных модулей программного обеспечения
для компьютерных систем»
по специальности среднего профессионального образования
09.02.03 «Программирование в компьютерных системах»
на 2021-2022 учебный год

<i>№ изменения, дата изменения</i>
1 Титульный лист, изменено «Факультет довузовской подготовки» на «Колледж» Основание: Приказ ректора университета № 421-«О» от 30.11.2020 «О создании Колледжа»

Рассмотрено и одобрено на заседании кафедры «Общепрофессиональных и специальных дисциплин»

Протокол № 9 « 05 » мая 2021 г.

Зав.каф. «Общепрофессиональных и специальных дисциплин»



/ Н.С. Ломакина

Лист изменений и дополнений

в комплект контрольно-оценочных средств учебного модуля
«Разработка программных модулей программного обеспечения
для компьютерных систем»
по специальности среднего профессионального образования
09.02.03 «Программирование в компьютерных системах»
на 2023-2024 учебный год

<i>№ изменения, дата изменения</i>
1 Актуализирован Список рекомендуемой литературы и источников информации