

Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 25.06.2026 13:02:32	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)	
Уникальный идентификатор документа: 04c19ed8b109b1506c07a48809a078888521525	Фонд оценочных средств для промежуточной аттестации по дисциплине "Архитектура вычислительных систем" по направлению подготовки (специальности) "09.03.04 Программная инженерия" направленности (профилю) <u>Разработка программно-информационных систем</u> ФГБОУ ВО «ЧелГУ»	стр. 1

**Фонд оценочных средств для промежуточной аттестации по дисциплине
Архитектура вычислительных систем**

Направление подготовки (специальность)

09.03.04 Программная инженерия

Направленность (профиль)

Разработка программно-информационных систем

Присваиваемая квалификация (степень)

бакалавр

Форма обучения

заочная форма обучения

Год(ы) набора 2026

Челябинск 2026 г.

09.03.04 Программная инженерия профиль Разработка программно-информационных систем, дисциплина Архитектура вычислительных систем, 2026 год набора, заочная форма обучения

Фонд оценочных средств дисциплины (модуля) одобрен и рекомендован:

Проректор по учебной работе утверждено 27.02.2026 А.А. Саламатов

Ученым советом института информационных технологий

Протокол заседания № 7 от 26.02.2026

Председатель Ученого совета
института информационных
технологий

согласовано

Ю.В. Петриченко

Заседанием кафедры информационных технологий и экономической информатики

Протокол заседания №7 от 26.02.2026

Заведующий кафедрой

согласовано

С.А. Скрипов

Автор (составитель)

С.А. Скрипов

Структура фондов оценочных средств соответствует приказу ректора ФГБОУ ВО «ЧелГУ» от 27 сентября 2022 № 573-1



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	6
3.1. Виды оценочных средств	6
3.2. Содержание оценочных средств	8
4. Порядок проведения и критерии оценивания промежуточной аттестации	30
4.1. Порядок проведения промежуточной аттестации	30
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	30
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций	30



1. Паспорт фонда оценочных средств

Направление подготовки: 09.03.04 Программная инженерия

Направленность: Разработка программно-информационных систем

Дисциплина: Архитектура вычислительных систем.

Курс (курсы) изучения: 1

Форма промежуточной аттестации: экзамен

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Архитектура вычислительных систем» направлено на формирование компетенций, приведённых в 1.

Таблица 1. Результаты обучения по дисциплине.

Коды компетенции и согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ОПК-1	Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-1.1. Демонстрирует знание основных положений и концепций в области математических и естественных наук, вычислительной техники и программирования ОПК-1.2. Демонстрирует умения решать стандартные задачи в профессиональной деятельности с применением естественнонаучных и общинженерных знаний, методов математического анализа и моделирования ОПК-1.3. Имеет практический опыт применения основных теорем и законов математики и естественных наук, методов моделирования, теоретического и экспериментального исследования для решения задач профессиональной деятельности	Знать: основные положения и концепции в области архитектуры вычислительных систем Уметь: решать стандартные задачи в профессиональной деятельности с учетом способов представления и обработки данных в ЭВМ Владеть: навыками программирования с использованием низкоуровневых языков программирования
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;	ОПК-6.1. Демонстрирует знание основ информатики, теории алгоритмов, методологии и технологии программирования ОПК-6.2. Демонстрирует умения разрабатывать алгоритмические и программные решения, проводить проектирование, конструирование и тестирование программных продуктов ОПК-6.3. Имеет	Знать: основы архитектуры вычислительных систем, теории алгоритмов Уметь: разрабатывать алгоритмические и программные решения с использованием низкоуровневых языков программирования Владеть: навыками использования технологий разработки программного обеспечения с использованием низкоуровневых языков программирования



		практический опыт использования технологий разработки программного обеспечения	
ОПК-7	Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой;	ОПК-7.1. Демонстрирует знание основных концепций, принципов, теорий и фактов, связанных с информатикой ОПК-7.2. Демонстрирует умения применять на практике основные концепции, принципы и теории из области информатики при решении стандартных задач ОПК-7.3. Имеет практический опыт решения задач профессиональной деятельности с использованием основ информатики	Знать: основные концепции, принципы, теории и факты, связанные с архитектурой вычислительных систем Уметь: применять на практике основные концепции, принципы и теории из области архитектуры вычислительных систем при решении стандартных задач Владеть: навыками решения задач профессиональной деятельности с использованием основ архитектуры вычислительных систем



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

№ п/п	Код компетенции/ планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1	ОПК-1.1. Демонстрирует знание основных положений и концепций в области математических и естественных наук, вычислительной техники и программирования Знать: основные положения и концепции в области архитектуры вычислительных систем	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 87-128
2	ОПК-1.2. Демонстрирует умения решать стандартные задачи в профессиональной деятельности с применением естественнонаучных и общинженерных знаний, методов математического анализа и моделирования Уметь: решать стандартные задачи в профессиональной деятельности с учетом способов представления и обработки данных в ЭВМ	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 22-86
3	ОПК-1.3. Имеет практический опыт применения основных теорем и законов математики и естественных наук, методов моделирования, теоретического и экспериментального исследования для решения задач профессиональной деятельности Владеть: навыками программирования с использованием низкоуровневых языков программирования	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 1-21, 129-167
4	ОПК-6.1. Демонстрирует знание основ информатики, теории алгоритмов, методологии и технологии программирования	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие	Тест	Задания теста № 87-128



	Знать: основы архитектуры вычислительных систем, теории алгоритмов	структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.		
5	ОПК-6.2. Демонстрирует умения разрабатывать алгоритмические и программные решения, проводить проектирование, конструирование и тестирование программных продуктов Уметь: разрабатывать алгоритмические и программные решения с использованием низкоуровневых языков программирования	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 22-86
6	ОПК-6.3. Имеет практический опыт использования технологий разработки программного обеспечения Владеть: навыками использования технологий разработки программного обеспечения с использованием низкоуровневых языков программирования	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 1-21, 129-167
7	ОПК-7.1. Демонстрирует знание основных концепций, принципов, теорий и фактов, связанных с информатикой Знать: основные концепции, принципы, теории и факты, связанные с архитектурой вычислительных систем	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 87-128
8	ОПК-7.2. Демонстрирует умения применять на практике основные концепции, принципы и теории из области информатики при решении стандартных задач Уметь: применять на практике основные концепции, принципы и теории из области архитектуры вычислительных систем при решении стандартных задач	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 22-86



9	ОПК-7.3. Имеет практический опыт решения задач профессиональной деятельности с использованием основ информатики Владеть: навыками решения задач профессиональной деятельности с использованием основ архитектуры вычислительных систем	Понятие архитектуры ЭВМ. Представление данных в ЭВМ. Организация вычислений. Взаимодействие структурных элементов ЭВМ и передача данных. Обзор современных архитектур и систем команд.	Тест	Задания теста № 1-21, 129-167
---	---	--	------	-------------------------------

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2. Содержание оценочных средств

База тестовых вопросов

№ п/п	Формулировка вопроса	Варианты ответов (полужирным шрифтом – верные варианты)
1.	Следующая часть кода выполняется на процессоре Intel 8086: ... mov ax,1 mov bx,2 push ax push bx pop bx pop ax ... Что будет храниться в регистре bx?	a. 2 b. 1 c. 0 d. 3
2.	Следующая часть кода выполняется на процессоре Intel 8086: ... mov ax,10 mov bx,3 push ax push bx pop ax pop bx ... Что будет храниться в регистре ax?	a. 3 b. 10 c. 0
3.	Следующая часть кода выполняется на процессоре Intel 8086: ... mov BP,SP inc [BP]	a. Последний элемент в стеке увеличится на 1 b. Последний элемент в стеке увеличится на 2 c. Предпоследний элемент в стеке увеличится на 1 d. Предпоследний элемент в стеке увеличится на 2



	add BP,2 ... Что произойдет?	
4.	Следующая часть кода выполняется на процессоре Intel 8086: ... push 3 push 5 pop ax ... Что будет храниться в регистре ax?	a. 5 b. 3 c. 8 d. 0
5.	Следующая часть кода выполняется на процессоре Intel 8086: ... push 3 push 5 push 6 pop bx pop bx pop bx ... Что будет храниться в регистре bx?	a. 3 b. 6 c. 5 d. 0 e. 14
6.	В процессоре Intel x86 стек:	a. Поддерживается аппаратно, вершина стека хранится в регистре SP b. Полностью хранится в кэше процессора c. Не поддерживается, стек, при необходимости, реализуется с помощью прикладной программы
7.	Особенностью стека является:	a. Извлекается последний помещенный элемент b. Извлекается первый помещенный элемент c. В стек всегда помещается бесконечное количество элементов (ограничено только объемом физической памяти)
8.	Следующая часть кода выполняется на процессоре Intel 8086: ... mov ax,-1 mov bx,-2 cmp ax,bx ja label3 ... ja (jump if above) предполагает, что числа не имеют знака. Переход происходит, если первое число больше. Произойдет ли переход по метке label3?	a. да b. нет c. Код не будет выполняться, так как содержит ошибку
9.	Следующая часть кода выполняется на процессоре Intel 8086:	a. да b. нет c. Из данного примера неясно



	<pre>... mov ax,3 mov bx,1 cmp ax,100 sub ax,bx jg label3 ... jg (jump if greater) предполагает, что числа имеют знак. Произойдет ли переход по метке label3?</pre>	
10.	<p>Следующая часть кода выполняется на процессоре Intel 8086:</p> <pre>... mov ax,3 mov bx,1 sub ax,bx cmp ax,100 jg label3 ... jg (jump if greater) предполагает, что числа имеют знак. Произойдет ли переход по метке label3?</pre>	<p>a. да b. нет c. Из данного примера неясно</p>
11.	<p>Следующая часть кода выполняется на процессоре Intel 8086:</p> <pre>... mov ax,3 mov bx,1 sub ax,bx jg label3 ... jg (jump if greater) предполагает, что числа имеют знак. Произойдет ли переход по метке label3?</pre>	<p>a. да b. нет c. Из данного примера неясно</p>
12.	<p>Следующая часть кода выполняется на процессоре Intel 8086:</p> <pre>... mov ax,3 mov bx,ax jg label3 ... jg (jump if greater) предполагает, что числа имеют знак. Команда mov значение регистра флагов не изменяет. Произойдет ли переход по метке label3?</pre>	<p>a. да b. нет c. Из данного примера неясно</p>
13.	Следующий код	a. 2



	выполняется на процессоре Intel 8086: <code>mov ax,2 jmp label3 mov ax,4 label3: mov bx,ax</code> Что будет записано в регистр <code>bx</code> ?	b. 4 c. 6 d. 0
14.	Следующая часть кода выполняется на процессоре Intel 8086: ... <code>mov ax,-1 mov bx,-2 cmp ax,bx jg label3</code> ... <code>jg (jump if greater)</code> предполагает, что числа имеют знак. Произойдет ли переход по метке <code>label3</code> ?	a. да b. нет c. Из данного примера неясно
15.	Следующая часть кода выполняется на процессоре Intel 8086: ... <code>mov ax,-1 mov bx,2 cmp ax,bx ja label3</code> ... <code>ja (jump if above)</code> предполагает, что числа не имеют знака. Переход происходит, если первое число больше. Произойдет ли переход по метке <code>label3</code> ?	a. да b. нет c. Код не будет выполняться, так как содержит ошибку
16.	Следующая часть кода выполняется на процессоре Intel 8086: ... <code>mov ax,-128 mov bx,128 cmp ax,bx ja label3</code> ... <code>ja (jump if above)</code> предполагает, что числа не имеют знака. Переход происходит, если первое число больше. Произойдет ли переход по метке <code>label3</code> ?	a. да b. нет c. Из данного примера неясно
17.	Следующая часть кода выполняется на	a. да b. нет



	<p>процессоре Intel 8086: ... mov ax,-2 mov bx,-1 cmp ax,bx jg label3 ... jg (jump if greater) предполагает, что числа имеют знак. Произойдет ли переход по метке label3?</p>	<p>с. Из данного примера неясно</p>
18.	<p>Следующая часть кода выполняется на процессоре Intel 8086: ... mov ax,-5 mov bx,ax mov ax,8 jg label3 ... jg (jump if greater) предполагает, что числа имеют знак. Команда mov значение регистра флагов не изменяет. Произойдет ли переход по метке label3?</p>	<p>а. да b. нет с. Из данного примера неясно</p>
19.	<p>Следующая часть кода выполняется на процессоре Intel 8086: ... mov ax,0 mov si,3 label1: inc ax sub si,3 jmp label1 ... Что будет записано в регистре ax?</p>	<p>а. Программа никогда не завершится b. 3 c. 2 d. 4</p>
20.	<p>Следующая часть кода выполняется на процессоре Intel 8086: ... mov ax,2 mov bx,-1 cmp ax,bx jg label3 ... jg (jump if greater) предполагает, что числа имеют знак. Произойдет ли переход по метке label3?</p>	<p>а. да b. нет с. Из данного примера неясно</p>
21.	<p>Следующая часть кода выполняется на</p>	<p>а. да b. нет</p>



	процессоре Intel 8086: ... mov ax,2 mov bx,1 jne label3 ... Команда mov регистр флагов не изменяет. Произойдет ли переход по метке label3?	с. Из данного примера неясно
22.	Переменные имеют размер 1 байт. В них хранятся числа без знака. Производится операция сложения: 10000000+ 00000001 Произойдет ли переполнение?	а. нет б. да
23.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 00101011+ 01010101 Произойдет ли переполнение?	а. да б. нет
24.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 01101010+ 01010101 Произойдет ли переполнение?	а. да б. нет
25.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 01111111+ 00000001 Произойдет ли переполнение?	а. да б. нет
26.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде.	а. да б. нет



	Производится операция сложения: 01111111+ 00000010 Произойдет ли перенос в знаковый разряд?	
27.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 01111111+ 01111111 Произойдет ли переполнение?	a. да b. нет
28.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 01111111+ 10000000 Произойдет ли переполнение?	a. нет b. да
29.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 01111111+ 10000010 Произойдет ли перенос в знаковый разряд?	a. да b. нет
30.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 10000000+ 01111111 Произойдет ли перенос за пределы разрядной сетки?	a. нет b. да
31.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения:	a. да b. нет



	10000000+ 10000000 Произойдет ли переполнение?	
32.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 10000000+ 11111111 Произойдет ли перенос за пределы разрядной сетки?	a. да b. нет
33.	Переменные имеют размер 1 байт. В них хранятся числа без знака. Производится операция сложения: 11111110+ 00000001 Произойдет ли переполнение?	a. нет b. да
34.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 10000000+ 11111111 Произойдет ли переполнение?	a. да b. нет
35.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 10000001+ 00000010 Произойдет ли перенос в знаковый разряд?	a. нет b. да
36.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 10000001+ 00000010 Произойдет ли перенос за пределы разрядной сетки?	a. нет b. да



37.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 1000001+ 1000010 Произойдет ли перенос в знаковый разряд?	a. нет b. да
38.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 1000001+ 1000010 Произойдет ли перенос за пределы разрядной сетки?	a. да b. нет
39.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 1010101+ 0101010 Произойдет ли переполнение?	a. нет b. да
40.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 1010101+ 1101010 Произойдет ли переполнение?	a. да b. нет
41.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 1101010+ 1010101 Произойдет ли переполнение?	a. да b. нет
42.	Переменные имеют размер 1 байт. В них	a. нет b. да



	хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 11010101+ 10101011 Произойдет ли переполнение?	
43.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 11111111+ 01111111 Произойдет ли переполнение?	a. нет b. да
44.	Переменные имеют размер 1 байт. В них хранятся числа без знака. Производится операция сложения: 11111110+ 00000010 Произойдет ли переполнение?	a. да b. нет
45.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 11111111+ 11111111 Произойдет ли переполнение?	a. нет b. да
46.	Переменные имеют размер 1 байт. В них хранятся числа без знака. Производится операция сложения: 11111111+ 11111111 Произойдет ли переполнение?	a. да b. нет
47.	Переменные имеют размер 1 байт. В них хранятся числа со знаком, записанные в дополнительном коде. Производится операция сложения: 00000001+	a. нет b. да



54.	Нормализованная запись числа 0.01 это:	a. $1 \cdot 10^{-2}$ b. $1 \cdot 10^2$ c. 10^{-3}
55.	Нормализованная запись числа 54654 это:	a. $5.4654 \cdot 10^4$ b. $0.54654 \cdot 10^5$ c. $1 \cdot 10^{54654}$
56.	Нормализованная запись числа 8.3 это:	a. $8.3 \cdot 10^0$ b. $8.3 \cdot 10^1$ c. $0.83 \cdot 10^1$
57.	Особенность сложения вещественных чисел состоит в том, что:	a. Могут быть потеряны младшие разряды b. Могут быть потеряны старшие разряды c. Может быть потерян знак
58.	При сложении вещественных чисел:	a. Порядок приводится к одному значению для обоих чисел b. Мантисса приводится к одному значению для обоих чисел c. Порядок первого числа приводится к мантиссе второго
59.	В переменную типа real (8 байт) можно записать числа в диапазоне:	a. $-1.8 \cdot 10^{308}..1.8 \cdot 10^{308}$ b. -2147483648..2147483647 c. 0..18446744073709551615
60.	В переменную типа single (вещественный тип, 4 байта) можно записать числа в диапазоне:	a. $-3.4 \cdot 10^{38}..3.4 \cdot 10^{38}$ b. -2147483648..2147483647 c. 0..4294967295
61.	В результате записи числа, имеющего 60 десятичных знаков в переменную типа real (8 байт):	a. Часть знаков будет потеряна b. Все знаки будут сохранены c. Будет потерян старший десятичный разряд
62.	Нормализованная запись числа 10.3 это:	a. $1.03 \cdot 10^1$ b. $10.3 \cdot 10^1$ c. $10.3 \cdot 10^0$
63.	Нормализованная запись числа 100 это:	a. $1 \cdot 10^2$ b. $10 \cdot 10^1$ c. $0.1 \cdot 10^3$ d. $0.1 \cdot 10^{-1}$
64.	Число 1 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.0 - мантисса, 0 - порядок b. 1.0 - мантисса, 1 - порядок c. 0.1 - мантисса, 1 - порядок d. 0.1 - мантисса, 10 - порядок
65.	Число 1.003 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной	a. 1.0000000110001... - мантисса, 0 - порядок b. 1.110001... - мантисса, 1 - порядок c. 1.110001 - мантисса, 101 - порядок d. 1.110001 - мантисса, 111 - порядок



	системе счисления):	
66.	Число 1.03 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.000001111010... - мантисса, 0 - порядок b. 1.0001111010... - мантисса, 101 - порядок c. 1.000011111010... - мантисса, 111 - порядок d. 1.000011111010... - мантисса, 1110 - порядок
67.	Число 1.375 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.011 - мантисса, 0 - порядок b. 1.11 - мантисса, 0 - порядок c. 1.101 - мантисса, 11 - порядок d. 1.101 - мантисса, 101 - порядок
68.	Число 1.5 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.1 - мантисса, 0 - порядок b. 1.101 - мантисса, 0 - порядок c. 1.101 - мантисса, 1 - порядок d. 1.101 - мантисса, 10 - порядок e. 1.1 - мантисса, 10 - порядок
69.	Число 1.625 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.101 - мантисса, 0 - порядок b. 1.101 - мантисса, 1 - порядок c. 1.0101 - мантисса, 1 - порядок d. 1.0111 - мантисса, 1 - порядок
70.	Число 1000000 (в десятичной системе) было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.1110100001001 - мантисса, 10011 - порядок b. 0.1110100001001 - мантисса, 10011 - порядок c. 1.11101001001 - мантисса, 10 - порядок d. 1.101101001001 - мантисса, 10 - порядок
71.	Число 12 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.1 - мантисса, 11 - порядок b. 1.1 - мантисса, 110 - порядок c. 1.1 - мантисса, 10 - порядок d. 1.001 - мантисса, 10 - порядок e. 1.001 - мантисса 111 - порядок
72.	Число 128 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.0 - мантисса, 111 - порядок b. 1.1 - мантисса, 111 - порядок c. 1.0 - мантисса, 11101 - порядок d. 1.0 - мантисса, 10101 - порядок
73.	Число 2 было записано в двоичном представлении в	a. 1.0 - мантисса, 1 - порядок b. 1.0 - мантисса, 10 - порядок



	нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	c. 0.1 - мантисса, 10 - порядок d. 0.1 - мантисса, 1 - порядок
74.	Число 256 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.0 - мантисса, 1000 - порядок b. 1.01 - мантисса, 1000 - порядок c. 1.01 - мантисса, 10 - порядок d. 1.0 - мантисса, 10 - порядок
75.	Число 3.25 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.101 - мантисса, 1 - порядок b. 1.101 - мантисса, 0 - порядок c. 1.101101 - мантисса, 1 - порядок d. 1.101101 - мантисса, 10 - порядок
76.	Число 3.3 было записано в двоичном представлении в нормализованном виде. Выберите правильный вариант (Варианты ответа записаны в двоичной системе счисления):	a. 1.101001100110... - мантисса, 1 - порядок b. 1.111001100110... - мантисса, 1 - порядок c. 1.101101100110... - мантисса, 1 - порядок d. 1.101101100110... - мантисса, 10 - порядок
77.	Команда add esx, eax	a. Вычислит сумму eax и esx. Полученное значение поместит в esx b. Вычислит сумму eax и esx. Полученное значение поместит в eax c. Вычислит сумму eax и esx. Полученное значение поместит в стек
78.	Команда add esx, 1 вычислит:	a. esx=esx+1 b. eax=esx+1 c. esx=1 d. eax=eax+esx+1
79.	Команда add 2, edx	a. Содержит ошибку b. Увеличит значение edx на 2 c. Вычислит edx=edx+edx
80.	Различие команд div и idiv заключается в следующем:	a. Команда div - для чисел без знака, idiv - для чисел со знаком b. Команда div - для 16-битных данных, idiv - для 32-битных c. Команда div - целочисленное умножение, idiv - умножение для вещественных чисел
81.	Команда inc является:	a. Однооперандной командой b. Двухоперандной командой c. Безоперандной командой
82.	Команда inc ah	a. Увеличит значение ah на 1 b. Уменьшит значение ah на 1 c. Обнулит ah d. Запишет в ah значение "1"
83.	Команда mul выполняет:	a. Целочисленное умножение без знака b. Целочисленное умножение со знаком



		c. Вычисление целой степени целого числа со знаком d. Вычисление степени числа "2" без знака
84.	После выполнения команды: mul edx в регистре edx будет храниться:	a. Старшая 32-битная часть 64-битного результата умножения b. Старое значение c. Результат умножения edx на eax d. 0
85.	Команда sub 2,eax	a. Содержит ошибку b. Уменьшит значение eax на 2 c. Обнулит eax
86.	Команда sub edx,2	a. Уменьшит значение edx на 2 b. Уменьшит значение edx на 8 c. Вычислит $eax=edx-2$ d. Содержит ошибку
87.	Выберите правильный вариант:	a. Современные процессоры x86 имеют встроенный математический сопроцессор b. Современные процессоры x86 не могут выполнять операции над вещественными числами c. Математический сопроцессор может работать только с целыми числами
88.	Регистры сопроцессора доступны программисту как:	a. Стек с вершиной ST b. R0..R7 c. FEAX, FEBX, FECX, FEDX
89.	Для хранения вещественных данных в своих регистрах сопроцессор использует:	a. 80-битный расширенный вещественный тип b. Тип double c. Тип __int64
90.	Математический сопроцессор способен загружать в свой стек:	a. Целочисленные и вещественные данные, которые он автоматически преобразует в 80-битное расширенное вещественное b. Только вещественные данные c. Только целочисленные данные
91.	Команда FADD без операндов:	a. Сложит ST и ST(1) b. Вызовет ошибку c. Сложит eax и ST
92.	В процессорах x86 флаг CF устанавливается в случае, если:	a. Произошел перенос за пределы разрядной сетки b. Произошло переполнение для чисел со знаком c. Результат выполнения операции равен нулю d. Результат выполнения операции меньше нуля (если считать его числом со знаком)
93.	В процессорах x86 флаг OF устанавливается в случае, если:	a. Произошел перенос за пределы разрядной сетки b. Произошло переполнение для чисел со знаком c. Результат выполнения операции равен нулю d. Результат выполнения операции меньше нуля (если считать его числом со знаком)
94.	В процессорах x86 флаг SF устанавливается в	a. Произошел перенос за пределы разрядной сетки b. Произошло переполнение для чисел со знаком



	случае, если:	с. Результат выполнения операции равен нулю d. Результат выполнения операции меньше нуля (если считать его числом со знаком)
95.	В процессорах x86 флаг ZF устанавливается в случае, если:	a. Произошел перенос за пределы разрядной сетки b. Произошло переполнение для чисел со знаком c. Результат выполнения операции равен нулю d. Результат выполнения операции меньше нуля (если считать его числом со знаком)
96.	Выберите команду, которая модифицирует регистр флагов:	a. add eax, 8 b. mov eax, 5 c. jmp label1
97.	Команда условного перехода jne для процессора Intel 8086 для принятия решения о переходе использует:	a. Регистр флагов b. Регистр ax c. Скрытый служебный регистр
98.	Арифметико-логические устройства (ALU) выполняют:	a. Простые арифметические действия (сложение, вычитание, сравнение) с целыми числами. b. Математические расчеты для чисел с плавающей точкой. c. Балансировку нагрузки на модули оперативной памяти.
99.	Для определения регистра в команде процессора Intel в поле ModR/M используется:	a. 3 бита. b. 5 бит. c. 16 бит. d. 12 бит.
100.	К регистрам общего назначения относится регистр:	a. EBX b. SS c. FLAGS d. EIP
101.	Команда процессора Intel обязательно содержит:	a. Поле "Опкод" b. Поле "Префикс" c. Поле "SIB" d. Поле "ModR/M" e. Поле "Смещение"
102.	Размер, порядок и тип операндов для команды процессора Intel определяет поле:	a. Опкод b. SIB c. "Непосредственное значение"
103.	Регистры al, ax, eax процессора Intel имеют одинаковые коды. Для точного выбора регистра используется:	a. Разрядность операндов b. Значение регистра EBP c. Значение регистра ESI
104.	Технология Hyper-Threading может увеличить эффективность процессора в том случае, если:	a. Вычислительные процессы распараллелены. b. Программы содержат однотипные операции. c. Выполняется только одна программа, имеющая один поток команд.
105.	Технология Hyper-Threading позволяет:	a. Работать с одним процессором как с двумя логическими устройствами.



		b. Выполнять инструкции программы в любом порядке. c. Уменьшить время выполнения любой программы в два раза.
106.	В процессорах Intel для разрешения/запрета на прерывания используется флаг:	a. IF b. OF c. CF d. ZF
107.	В процессоре Intel программные прерывания вызываются инструкцией:	a. int b. call c. cli d. out
108.	Для осуществления системных вызовов обычно используются:	a. Программные прерывания b. Ложные прерывания c. Команды математического сопроцессора
109.	В процессоре Intel команды cli, sti используются для:	a. Установки/снятия флага прерываний b. Вызова программных прерываний c. Добавления обработчика прерываний
110.	Прерывания позволяют:	a. Внешним устройствам "прерывать" работу процессора b. Процессору "прерывать" работу внешних устройств c. Пользователю синхронизировать тактовый генератор процессора
111.	Прерывания, которые можно запрещать установкой соответствующих битов в регистре процессора, называются:	a. Маскируемые прерывания b. Немаскируемые прерывания c. Контролируемые прерывания
112.	Прерывания, которые обрабатываются всегда, независимо от запретов на другие прерывания, называются:	a. Немаскируемые прерывания b. Маскируемые прерывания c. Аппаратные прерывания
113.	При возникновении прерывания управление передается:	a. Обработчику прерывания b. Контроллеру устройства c. Программе, которая вызвала прерывание
114.	Для CISC характерно:	a. Небольшое количество регистров общего назначения b. Небольшое количество инструкций, ориентированных на типовой код, генерируемый компиляторами c. Использование исключительно регистрового способа адресации
115.	Для RISC характерно:	a. Минимальный набор инструкций b. Небольшое количество регистров общего назначения c. Множество способов адресации памяти
116.	Процессоры x86 внешне выглядят как:	a. CISC b. RISC c. EPIC
117.	Процессор Core i7 имеет архитектуру:	a. x86-64 b. IA-64 c. MISC



118.	Процессор Itanium2 имеет архитектуру:	a. IA-64 b. x86-64 c. CISC
119.	Выберите наиболее правильный вариант. Какому типу памяти соответствует "оперативная память" современного компьютера:	a. DRAM b. SRAM c. Flash
120.	Регистр процессора представляет собой:	a. Ячейку памяти, расположенную непосредственно в процессоре b. Генератор импульсов для синхронизации всех операций c. Буферную DRAM память между процессором и оперативной памятью
121.	Кэш процессора:	a. Служит для увеличения скорости обмена данными с оперативной памятью b. В современных процессорах не применяется c. В современных ЭВМ обычно устанавливается как отдельная микросхема на материнской плате
122.	Какой исполнительный блок процессора обычно отвечает за целочисленное сложение:	a. ALU b. FPU c. AVX
123.	Особенностью SRAM памяти является:	a. Доступ к ячейке всегда занимает одно и то же время b. Для доступа к ячейке памяти требуется сложный контроллер, поддерживающий подзарядку и перезарядку ячеек c. Это всегда энергонезависимая память
124.	Согласно принципам Фон Неймана:	a. Данные и программы хранятся в одной и той же памяти. b. Вычислительная машина имеет два вида памяти: для хранения программ и данных. c. Программа считывается процессором напрямую с жесткого диска.
125.	Отличие гарвардской архитектуры от Фон неймановской состоит в следующем:	a. Вычислительная машина имеет два вида памяти: для хранения программ и данных. b. Операции над вещественными числами реализуются через простейшие целочисленные операции. c. Для представления данных используется троичная система счисления.
126.	Современный персональный компьютер имеет архитектуру, наиболее похожую на:	a. Фон Неймановскую b. Гарвардскую c. Архитектуру MISD
127.	Современный микроконтроллер имеет архитектуру, наиболее похожую на:	a. Гарвардскую b. Фон Неймановскую c. Архитектуру SIMD
128.	Выберите правильное утверждение:	a. Для каждой архитектуры ЭВМ необходим свой язык ассемблера. b. Каждая команда на ассемблере строго соответствует конкретной машинной команде.



		с. Программа, написанная на ассемблере, является универсальной, может использоваться для любой архитектуры и операционной системы.
129.	Архитектура x86 накладывает следующее ограничение на операнды:	a. Оба операнда команды не могут одновременно быть переменными в оперативной памяти. b. Оба операнда команды не могут одновременно быть регистрами. c. Нельзя использовать 8-битные операнды.
130.	Для использования косвенно-регистравой адресации в ассемблере x86 необходимо использовать:	a. Квадратные скобки [] b. Точку с запятой ; c. Регистр ebx без каких-либо операторов, в котором не может храниться ничего, кроме адреса.
131.	Выберите допустимую команду:	a. MOV CH,BL b. MOV 3,AL c. MOV a,b (где a и b - переменные типа int)
132.	Команда mov eax,[ebx]	a. Скопирует в регистр eax значение переменной в оперативной памяти, адрес которой хранится в ebx b. Поместит в eax порядковый номер регистра ebx c. Скопирует в регистр eax значение регистра ebx d. Обнулит регистр eax
133.	Команда mov eax,[x] где x- переменная в оперативной памяти:	a. Скопирует в регистр eax значение переменной x b. Скопирует в регистр eax значение переменной , адрес которой хранится в переменной x c. Скопирует в регистр eax адрес переменной x d. Обнулит регистр eax
134.	Выберите недопустимую команду:	a. mov ah,eax b. mov eax,ecx c. mov al,ch
135.	Выберите недопустимую команду:	a. mov 0,eax b. mov eax,0 c. mov eax,-1
136.	Выберите команду, где один из операндов является непосредственным значением:	a. mov ax,5 b. mov ax,bx c. mov eax,[ebx]
137.	После выполнения команды mov eax,-5 в регистре eax будет храниться:	a. 32-битное число -5 в дополнительном коде b. 16-битное число -5 в прямом коде c. 0 d. Старое значение, которое хранилось до выполнения команды
138.	После выполнения команд: mov al,0;mov ah,1 в регистре ax будет храниться значение:	a. 256 b. 1 c. 0 d. -1
139.	После выполнения команд: mov al,0;mov ah,2 в регистре ax будет	a. 512 b. 1 c. 0



	храниться значение:	d. -1
140.	Для адресации устройства на шине программы как правило используют:	a. Порты ввода-вывода. b. Физический номер слота, в котором установлено устройство. c. Серийный номер устройства.
141.	Для записи в порт ввода-вывода процессор Intel может использовать команду:	a. out b. call c. ret d. sti
142.	Для соединения x86 процессора с внутренними устройствами используется шина:	a. FSB (Front Side Bus) b. AGP (Accelerated Graphics Port) c. LPC(Low Pin Count)
143.	Для чтения из порта ввода-вывода процессор Intel может использовать команду:	a. in b. ret c. cli d. call
144.	Разрядность шины данных определяет:	a. Максимальный объем информации, которая может быть получена или передана за один раз. b. Тактовую частоту шины. c. Тактовую частоту устройств, подключенных к шине.
145.	Компьютерная шина это:	a. Подсистема, которая передаёт данные между функциональными блоками компьютера b. Подсистема для хранения данных регистров процессора c. Подсистема, определяющая набор команд процессора
146.	Перспективный способ передачи данных это:	a. Последовательная передача b. Параллельная передача c. Симплексная передача
147.	Команда call:	a. Помещает адрес возврата в стек и передает управление подпрограмме b. Возвращает управление предыдущей функции c. Только сохраняет адрес возврата в специальном регистре
148.	Команда ret:	a. Извлекает из стека адрес возврата и передает управление по этому адресу b. Передает управление по адресу в регистрах c. Прекращает выполнение программы
149.	Локальные переменные функции хранятся:	a. В стеке b. Всегда в основной памяти (не в стеке) c. Всегда в регистрах
150.	Преимуществом подпрограммы является:	a. Возможность повторного использования программного кода b. Код, выполненный с использованием подпрограмм выполняется быстрее c. При использовании подпрограмм экономится память стека
151.	При вызове подпрограммы, адрес возврата:	a. Сохраняется в стеке автоматически b. Хранить не нужно c. Сохраняется в регистрах процессора



152.	Соглашение вызова fastcall:	a. Использует регистры для хранения параметров b. Не может использоваться на современных процессорах c. Обычно не используется, так как при этом время работы функции увеличивается
153.	Соглашение вызова определяет:	a. Порядок передачи параметров b. Время работы функции c. Последовательность вызова функциями друг друга
154.	Команда mov esx, eax	a. Скопирует содержимое eax в esx b. Скопирует содержимое esx в eax c. Поменяет местами значения eax и esx d. Содержит ошибку
155.	Выберите 8-битный регистр:	a. BL b. AX c. ECX
156.	Выберите 8-битный регистр:	a. BH b. BX c. RBX
157.	Выберите 16-битный регистр:	a. CX b. EBX c. RCX
158.	Выберите 16-битный регистр:	a. DX b. DL c. DH
159.	Выберите 32-битный регистр:	a. EBX b. RBX c. DH
160.	Выберите 32-битный регистр:	a. ECX b. AX c. DL
161.	Выберите 64-битный регистр:	a. RBX b. EAX c. BH
162.	Выберите 64-битный регистр:	a. RAX b. BX c. AL
163.	Для использования ассемблерных вставок в среде Visual Studio необходимо использовать ключевое слово:	a. __asm b. inline c. cdecl
164.	Выберите правильный вариант:	a. При входе в ассемблерную вставку регистры могут принимать произвольные значения. b. При входе в ассемблерную вставку регистры обнуляются. c. Для использования регистров в ассемблерной вставке необходимо



		их инициализировать, объявив как локальные переменные.
165.	Выберите правильный вариант:	<p>a. При выходе из ассемблерной вставки необходимо восстановить значения регистров esp и ebp, если они были модифицированы.</p> <p>b. При выходе из ассемблерной вставки необходимо восстановить все регистры, которые были модифицированы.</p> <p>c. В ассемблерной вставке можно использовать любые регистры для любых целей, при выходе из ассемблерной вставки восстанавливать их необязательно.</p>
166.	Локальные переменные функций C/C++ доступны внутри ассемблерной вставки только если:	<p>a. Не модифицировался регистр ebp.</p> <p>b. Значение переменной было скопировано в регистр общего назначения до входа в ассемблерную вставку.</p> <p>c. Переменная объявлена без ключевого слова "static".</p>
167.	Ключевое слово __asm используется:	<p>a. В компиляторах Microsoft.</p> <p>b. В языках C/C++ для любого компилятора.</p> <p>c. В старых 16-битных компиляторах. В 32/64-битных компиляторах оно не используется.</p>



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

Экзамен проводится в виде тестирования. Студент должен ответить на вопросы закрытого типа, которые предполагают выбор вариантов ответа, а также на вопросы открытого типа, которые не предполагают вариантов ответа, правильный ответ требуется написать самостоятельно. Всего 20 тестовых вопросов. Продолжительность теста – 35 минут.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Тест формируется в системе электронного обучения MOODLE.

Максимальный балл за тест — 100 баллов.

Оценка	Отлично/ Зачтено	Хорошо/ зачтено	Удовлетворитель- но/зачтено	Неудовлетворительно / незачтено
Баллы	100-90 баллов	89-75 баллов	74-60 баллов	59-0 баллов
Уровень освоения проверяемых компетенций	высокий	средний	базовый	недостаточный

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

При подведении итогов учитываются результаты только промежуточной аттестации:

0-59 баллов – неудовлетворительно/незачтено;

60-74 баллов – удовлетворительно/зачтено;

75-89 баллов – хорошо/зачтено;

90-100 баллов – отлично/зачтено;

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке отлично:
 - предполагает формирование компетенций на высоком уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки отлично;
 - студент умеет применять на практике знания, полученные в рамках изучения дисциплины



- формируются навыки использования теоретических и практических разделов дисциплины для решения задач профессиональной деятельности;
- 2. Средний уровень соответствует оценке хорошо:
 - предполагает формирование компетенций на среднем уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки хорошо;
 - студент умеет применять знания, полученные в рамках изучения дисциплины, для решения задач профессиональной деятельности;
- 3. Базовый уровень соответствует оценке удовлетворительно:
 - предполагает формирование компетенций на базовом уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки удовлетворительно;
- 4. Недостаточный уровень соответствует оценке неудовлетворительно.