

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 16:15:01

Уникальный программный ключ:

990bea3aeec548c23bd8699e48288f8d19d0c800

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института
перспективной инженерии,
кандидат тех. наук, доцент

А.А. Порохня

Ф.И.О.

«__» _____ 20__ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля успеваемости и промежуточной аттестации по

Информационные технологии и программирование

название дисциплины (модуля)

Направление подготовки

09.03.02 Информационные системы и технологии

Направленность (профиль)

Прикладное программирование и интернет-технологии

Форма обучения

очная

Год начала обучения

2026

Реализуется в 1, 2 семестрах

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимся установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Информационные технологии и программирование».

3. Разработчик: Березина В.А., ст. преподаватель департамента цифровых, робототехнических систем и электроники.

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Азаров И. В., и.о. директора департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Члены комиссии:

Шагрова Г. В., профессор департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Винокурский Д. Л., доцент департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Представитель _____ организации-работодателя:

(ФИО, должность)

Экспертное заключение: Оценочные материалы (оценочные средства) по дисциплине «Информационные технологии и программирование» составлены в соответствии с требованиями самостоятельно устанавливаемого Федерального государственного образовательного стандарта высшего образования по направлению 09.03.02 Информационные системы и технологии и позволяют оценить уровень сформированности компетенции ОПК-2.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенци(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
<i>Компетенция: ОПК-2</i> Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности				
Результаты обучения по дисциплине (модулю): демонстрирует знания современных информационных технологий и программных средств, в том числе отечественного производства при решении задач профессиональной деятельности <i>Индикатор:</i> ОПК-2ид-2.1	Не знает современных ИТ-средств. Не ориентируется в отличиях зарубежного ПО от отечественного .	Знает основные виды ПО, но имеет смутное представление о реестре отечественного ПО и конкретных продуктах РФ.	Уверенно перечисляет современные ИТ-решения и их российские аналоги. Понимает область их применения.	Демонстрирует глубокие знания рынка ИТ. Свободно ориентируется в функционале и архитектуре передового отечественного ПО.
Результаты обучения по дисциплине (модулю): выбирает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности <i>Индикатор:</i> ОПК-2ид-2.2	Не способен выбрать ПО для решения задачи. Не понимает критериев импортозамещения.	Выбирает ПО по шаблону. Затрудняется аргументировать выбор конкретного отечественного аналога.	Самостоятельно подбирает программный стек, учитывая требования ТЗ и актуальные нормативные акты по ИТ.	Проводит комплексный сравнительный анализ. Обосновывает выбор ПО с учетом безопасности и совместимости систем.
Результаты обучения по дисциплине (модулю): применяет принципы работы современных информационных технологий и программных средств, в том числе отечественного производства <i>Индикатор:</i> ОПК-2ид-2.3	Не понимает принципов работы ОС и прикладных систем. Не знает специфику работы в среде.	Имеет общее представление о логике работы ПО. Путается в механизмах взаимодействия программных компонентов.	Четко описывает принципы работы современных ИТ-решений. Понимает особенности работы в отечественных ОС.	Глубоко владеет теорией функционирования информационных систем. Может объяснить низкоуровневые процессы обработки данных.
Результаты обучения по дисциплине (модулю): использует современные информационные технологий и программные средства, в том числе	Не умеет запускать и настраивать ПО. Работа в отечественных	Решает базовые задачи в ПО под руководством	Уверенно использует прикладное и системное ПО (в т.ч.	Владеет продвинутым функционалом программ. Эффективно

отечественного производства, при решении задач профессиональной деятельности <i>Индикатор:</i> ОПК-2_{ид-2.4}	интерфейсах вызывает непреодолимые трудности.	преподавателя. Часто допускает ошибки при настройке параметров.	отечественно е) для выполнения профессиональных заданий.	решает сложные задачи, оптимизирует настройки под свои нужды.
---	--	--	---	--

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Вопросы для собеседования:	
1		1. Системы контроля. Принципы работы Git и отличия состояний Modified, Staged и Committed. 2. Коллективная разработка. Назначение механизма Pull Request и его роль в процессе код-ревью. 3. Стандарты разметки. Синтаксис Markdown для оформления документации и вставки медиафайлов. 4. Безопасность проекта. Использование файла .gitignore для исключения служебных данных и секретов. 5. Семантическая верстка. Преимущества тегов header и main перед универсальными контейнерами div. 6. Доступность интерфейсов. Роль атрибута alt у изображений и его влияние на работу скринридеров. 7. Блочная модель. Компоненты Box Model и их влияние на расчет итоговых размеров элементов. 8. Единицы измерения. Различия между относительными и абсолютными величинами в CSS-стилях. 9. Гибкая компоновка. Принципы работы Flexbox и разница между осями выравнивания контента. 10. Облачная публикация. Алгоритм развертывания статического сайта с помощью сервиса GitHub Pages. 11. Документация как код. Принципиальное отличие Typst от классических визуальных редакторов. 12. Автоматизация оформления. Применение функций set и show для управления стилем документа. 13. Интерпретируемые языки. Особенности выполнения кода Python и его отличие от компилируемого ПО. 14. Динамическая типизация. Механизм определения типов данных в Python и связанные с этим риски.	ОПК-2

		15. Форматирование строк. Преимущества f-строк перед конкатенацией при выводе сложных данных. 16. Логические операции. Принцип работы операторов and, or и механизм ленивых вычислений. 17. Управляющие циклы. Критерии выбора между конструкциями while и for в зависимости от задачи. 18. Линейные коллекции. Отличия списков от кортежей и причины экономии памяти в неизменяемых типах. 19. Ассоциативные массивы. Структура словаря dict и эффективность поиска данных по уникальному ключу. 20. Механизм срезов. Использование синтаксиса slice для извлечения подстрок и разворота коллекций. 21. Области видимости. Взаимодействие локальных и глобальных переменных внутри функций Python. 22. Проектирование функций. Использование аргументов по умолчанию для создания гибких интерфейсов. 23. Обработка исключений. Применение блоков try-except для повышения отказоустойчивости программ. 24. Менеджеры контекста. Использование оператора with для безопасного управления ресурсами файлов. 25. Интеграция технологий. Процесс автоматической подготовки данных в Python для отчетов в Typst. 26. Контроль версий. Назначение команды git checkout и принципы работы с ветками проекта. 27. Стиливая специфика. Понятие каскадности в CSS и приоритеты различных типов селекторов. 28. Математическая верстка. Синтаксис написания сложных формул в Typst для технических отчетов. 29. Алгоритмическая сложность. Примеры эффективного использования встроенных методов списков и словарей. 30. Модульная архитектура. Принципы импорта стандартных библиотек и создание собственных модулей. 31. Инструментарий сборки. Роль компилятора, линковщика и препроцессора в создании	
--	--	---	--

		исполняемого файла. 32. Стандарты языка. Ключевые отличия между современными стандартами C++17/C++20 и классическим C++98. 33. Типизация данных. Особенности статической типизации и правила неявного приведения типов в выражениях. 34. Пространства имен. Назначение оператора scope resolution (::) и риски использования using namespace std. 35. Ввод-вывод. Механизмы работы потоков iostream и использование манипуляторов для форматирования консоли. 36. Логика ветвления. Особенности работы оператора switch-case и использование тернарного оператора в коде. 37. Управление циклами. Различия в логике выполнения конструкций while и do-while при проверке условий. 38. Передача аргументов. Сравнение передачи параметров в функцию по значению, указателю и ссылке. 39. Константная безопасность. Использование модификатора const для защиты данных от непреднамеренного изменения. 40. Перегрузка функций. Принципы создания функций с одинаковыми именами и алгоритм выбора нужной версии компилятором. 41. Адресная арифметика. Связь между указателями и массивами, а также операции смещения по адресам памяти. 42. Динамическая память. Жизненный цикл объектов в куче и ответственность программиста при использовании new и delete. 43. Утечки ресурсов. Причины возникновения memory leaks и методы их обнаружения в сложных программах. 44. Символьные массивы. Особенности строк в стиле C и роль терминирующего нуля в конце последовательности. 45. Объектная декомпозиция. Принципиальное различие между структурами struct и классами class в языке C++. 46. Инкапсуляция данных. Применение модификаторов public, private и protected для управления доступом к полям. 47. Конструкторы классов. Порядок инициализации полей объекта и назначение списков инициализации.	
--	--	--	--

		48. Деструкторы объектов. Правила автоматического освобождения ресурсов и очистки памяти при удалении экземпляра. 49. Контекст объекта. Назначение скрытого указателя <code>this</code> и способы его применения внутри методов класса. 50. Иерархия наследования. Механизм расширения функционала базового класса в производных классах. 51. Динамический полиморфизм. Использование виртуальных функций для реализации позднего связывания объектов. 52. Чистая виртуальность. Понятие абстрактного класса и создание интерфейсов для проектирования систем. 53. Виртуальные деструкторы. Необходимость пометки деструктора базового класса как <code>virtual</code> для корректного удаления потомков. 54. Обобщенное программирование. Принципы создания шаблонов функций и классов для работы с любыми типами данных. 55. Контейнеры STL. Сравнение внутреннего устройства и производительности векторов (<code>vector</code>) и списков (<code>list</code>). 56. Ассоциативные коллекции. Механизм хранения данных в <code>std::map</code> и использование ключей для быстрого поиска. 57. Механизм итераторов. Универсальный способ навигации по различным контейнерам стандартной библиотеки. 58. Файловые потоки. Особенности работы с библиотекой <code>fstream</code> для сохранения и чтения структурированных данных. 59. Обработка исключений. Использование блоков <code>try-catch</code> для перехвата ошибок и обеспечения стабильности приложения. 60. Идиома RAII. Принцип управления ресурсами через привязку их времени жизни к области видимости объекта.	
		Задания закрытого типа:	
61.	2	Какая команда <code>git</code> используется для создания связи локального репозитория с удаленным на <code>github</code>? 1) <code>git init</code> 2) <code>git remote add origin</code> 3) <code>git commit -m</code>	ОПК-2

		4) git push	
62.	2	Для чего предназначен файл .gitignore в корне проекта? 1) Для хранения паролей и секретных ключей 2) Для указания файлов, которые не должны отслеживаться Git 3) Для настройки цветовой темы VS Code 4) Для автоматической публикации сайта	ОПК-2
63.	3	Какой атрибут тега отвечает за текстовое описание изображения на случай его отсутствия? 1) title 2) src 3) alt 4) href	ОПК-2
64.	2	Какое css-свойство определяет внутренние отступы внутри блока? 1) margin 2) padding 3) border 4) display	ОПК-2
65.	3	Какое значение свойства display включает режим гибкой верстки flexbox? 1) block 2) inline 3) flex 4) grid	ОПК-2
66.	2	В какой формат компилируется документ typst по умолчанию для просмотра? 1) HTML 2) PDF 3) DOCX 4) LATEX	ОПК-2

67.	2	Каким символом в typst выделяется блок математических формул? 1) # 2) \$ 3) @ 4) &	ОПК-2
68.	3	Какой результат вернет функция type(3.14) в python? 1) <class 'int'> 2) <class 'str'> 3) <class 'float'> 4) <class 'list'>	ОПК-2
69.	4	Какая функция в python используется для получения данных от пользователя через консоль? 1) print() 2) output() 3) get() 4) input()	ОПК-2
70.	2	Что выведет код: print(10 // 3)? 1) 3.333 2) 3 3) 1 4) 0	ОПК-2
71.	3	Какой из операторов python означает 'не равно'? 1) not = 2) <> 3) != 4) ==	ОПК-2
72.	2	Какое ключевое слово используется для добавления альтернативного условия, если первое ложно?	ОПК-2

		1) else if 2) elif 3) if not 4) case	
73.	2	Что делает функция range(5) в цикле for? 1) Генерирует последовательность от 1 до 5 2) Генерирует последовательность от 0 до 4 3) Создает список из пяти случайных чисел 4) Повторяет цикл бесконечно	ОПК-2
74.	3	Какая из коллекций в python является неизменяемой (immutable)? 1) Список (list) 2) Словарь (dict) 3) Кортеж (tuple) 4) Множество (set)	ОПК-2
75.	3	Как обратиться к последнему элементу списка 'my_list' в python? 1) my_list[last] 2) my_list[0] 3) my_list[-1] 4) my_list.end()	ОПК-2
76.	4	Какой метод используется для добавления нового элемента в конец списка? 1) add() 2) push() 3) insert() 4) append()	ОПК-2
77.	2	Каким символом разделяются ключ и значение в словаре python? 1) = 2) : 3) ->	ОПК-2

		4) -	
78.	3	Какое ключевое слово используется для создания (объявления) функции? 1) function 2) func 3) def 4) void	ОПК-2
79.	3	Для чего используется оператор return в функциях? 1) Для прерывания цикла 2) Для вывода текста на экран 3) Для возврата результата выполнения функции 4) Для импорта модуля	ОПК-2
80.	3	Какой режим открытия файла в функции open() используется только для чтения? 1) 'w' 2) 'a' 3) 'r' 4) 'x'	ОПК-2
81.	2	Какой менеджер контекста гарантирует закрытие файла после работы с ним? 1) try/except 2) with 3) global 4) for	ОПК-2
82.	2	Какой оператор используется для выделения памяти в куче (heap) в c++? 1) malloc 2) new 3) alloc 4) push	ОПК-2

83.	3	Какое ключевое слово используется для освобождения памяти, выделенной под одиночный объект через new? 1) free 2) remove 3) delete 4) clear	ОПК-2
84.	2	В чем основное отличие struct от class в c++ по умолчанию? 1) В struct нельзя использовать функции 2) В struct тип доступа по умолчанию public, а в class — private 3) В class нельзя использовать указатели 4) struct занимает меньше места в памяти	ОПК-2
85.	2	Какая директива препроцессора необходима для работы со стандартным вводом-выводом (cin, cout)? 1) #include <stdio.h> 2) #include <iostream> 3) #include <string> 4) #include <fstream>	ОПК-2
86.	3	Какой модификатор доступа делает члены класса видимыми в самом классе и в его потомках, но скрытыми для внешнего кода? 1) public 2) private 3) protected 4) friend	ОПК-2
87.	3	Что означает спецификатор 'virtual' перед методом базового класса? 1) Метод нельзя вызывать 2) Метод будет выполняться только в базовом классе 3) Метод может быть переопределен в производных классах для обеспечения полиморфизма	ОПК-2

		4) Метод является статическим	
88.	3	Каким символом обозначается деструктор класса? 1) ! 2) & 3) ~ 4) *	ОПК-2
89.	3	Какая библиотека stl предоставляет реализацию динамического массива? 1) <list> 2) <array> 3) <vector> 4) <deque>	ОПК-2
90.	1	Что такое чисто виртуальная функция (pure virtual function)? 1) Функция, у которой нет тела и которая приравнена к 0 2) Функция, работающая только с целыми числами 3) Функция, помеченная словом const 4) Функция, объявленная в секции public	ОПК-2
91.	3	Какой оператор используется для доступа к членам объекта через указатель на этот объект? 1) 2) :: 3) -> 4) &	ОПК-2
92.	3	Какое ключевое слово используется для создания шаблона функции или класса? 1) generic 2) typename 3) template 4) auto	ОПК-2
93.	3	Какой метод вектора (std::vector) добавляет элемент в конец?	ОПК-2

		1) add() 2) append() 3) push_back() 4) insert()	
94.	2	Какое исключение будет сгенерировано методом .at() при обращении к индексу вне границ вектора? 1) std::runtime_error 2) std::out_of_range 3) std::bad_alloc 4) std::logic_error	ОПК-2
95.	2	Какое пространство имен содержит стандартные компоненты c++, такие как cout и vector? 1) core 2) std 3) main 4) cpp	ОПК-2
96.	2	Что произойдет, если в классе с виртуальными методами не объявить деструктор виртуальным? 1) Программа не скомпилируется 2) При удалении через указатель на базовый класс деструктор потомка не будет вызван 3) Объект нельзя будет создать 4) Память будет очищена в два раза быстрее	ОПК-2
97.	2	Какой заголовочный файл нужен для работы с файловыми потоками (чтение/запись файлов)? 1) <iostream> 2) <fstream> 3) <file> 4) <stdio>	ОПК-2

98.	3	Каким значением инициализируется указатель, если он ни на что не указывает (в современном c++)? 1) 0 2) NULL 3) nullptr 4) void	ОПК-2
99.	2	Что делает оператор '::'? 1) Разыменовывает указатель 2) Разрешает область видимости (scope resolution) 3) Сравнивает два объекта 4) Создает ссылку	ОПК-2
100.	3	Как передать аргумент в функцию так, чтобы функция работала с оригиналом переменной, но не могла его изменить? 1) По значению (int x) 2) По ссылке (int &x) 3) По константной ссылке (const int &x) 4) По указателю (int *x)	ОПК-2
101.	3	Какой контейнер stl лучше всего подходит для быстрого поиска значения по уникальному ключу? 1) std::vector 2) std::list 3) std::map 4) std::stack	ОПК-2
102.	3	Что делает функция-член .close() у файлового потока? 1) Удаляет файл с диска 2) Очищает содержимое файла 3) Завершает работу с файлом и освобождает системный ресурс 4) Скрывает файл	ОПК-2

103.	3	Какое слово используется для перехвата исключения? 1) throw 2) try 3) catch 4) except	ОПК-2
104.	1	Каков размер типа char в байтах согласно стандарту? 1) 1 2) 2 3) 4 4) 8	ОПК-2
105.	3	Что возвращает оператор 'sizeof'? 1) Значение переменной 2) Адрес переменной 3) Размер типа или объекта в байтах 4) Количество элементов в массиве	ОПК-2
106.	3	Какое ключевое слово гарантирует, что метод класса не будет изменять его поля? 1) static 2) final 3) const 4) immutable	ОПК-2
		Задания открытого типа:	
107.	git	Как называется распределенная система контроля версий, используемая в курсе?	ОПК-2
108.	readme	Как называется файл, содержащий описание проекта и отображаемый на главной странице репозитория github?	ОПК-2
109.	commit	Какая команда git используется для фиксации проиндексированных изменений в истории проекта?	ОПК-2

110.	markdown	Как называется язык облегченной разметки текста, используемый для оформления документации в vs code?	ОПК-2
111.	html	Как называется корневой элемент (тег) любого корректного html-документа?	ОПК-2
112.	a	Какой тег используется для создания гиперссылок в веб-разработке?	ОПК-2
113.	head	Как называется секция html-документа, в которой подключаются внешние стили и шрифты?	ОПК-2
114.	css	Какая технология отвечает за визуальное оформление и стилизацию веб-страниц?	ОПК-2
115.	margin	Как называется внешнее поле (отступ) элемента от других блоков в блочной модели css?	ОПК-2
116.	pages	Как называется сервис github для бесплатного хостинга и публикации статических сайтов?	ОПК-2
117.	typst	Как называется современная система верстки, позиционируемая как более простая альтернатива latex?	ОПК-2
118.	компиляция	Как называется процесс преобразования исходного кода (например, .tex или .py) в результат (pdf или выполнение)?	ОПК-2
119.	int	Как называется тип данных в python для хранения целых чисел?	ОПК-2
120.	if	Какое ключевое слово используется в python для создания условий?	ОПК-2
121.	for	Как называется цикл в python, который чаще всего используется для обхода элементов коллекции?	ОПК-2
122.	list	Как называется изменяемый тип данных в python для хранения упорядоченного набора элементов?	ОПК-2
123.	tuple	Как называется неизменяемый аналог списка в языке python?	ОПК-2

124.	dict	Как называется структура данных, хранящая информацию в формате пар 'ключ-значение'?	ОПК-2
125.	срез	Как называется механизм в python для извлечения подстроки или части списка по индексам?	ОПК-2
126.	def	Какое ключевое слово используется для определения функции в python?	ОПК-2
127.	результат	Как называется значение, которое функция отправляет обратно в место своего вызова?	ОПК-2
128.	math	Как называется стандартная библиотека python для работы с математическими функциями (sin, sqrt)?	ОПК-2
129.	исключение	Как называется механизм в python для перехвата и обработки ошибок во время выполнения?	ОПК-2
130.	with	Какой оператор (менеджер контекста) гарантирует автоматическое закрытие файла после работы?	ОПК-2
131.	терминал	Как называется встроенный в vs code инструмент для ввода команд и запуска скриптов?	ОПК-2
132.	компиляция	Как называется процесс преобразования исходного кода c++ в объектный код, выполняемый специальной программой?	ОПК-2
133.	амперсанд	Какой оператор используется для получения адреса переменной в памяти?	ОПК-2
134.	конструктор	Как называется функция, которая имеет то же имя, что и класс, и вызывается автоматически при создании объекта?	ОПК-2
135.	чистая	Как называется функция, помеченная ключевым словом virtual и приравненная к нулю в базовом классе?	ОПК-2
136.	стрелка	Какой оператор в c++ используется для доступа к членам класса через указатель?	ОПК-2

137.	куча	Как называется область памяти, в которой размещаются динамические объекты, созданные через оператор new?	ОПК-2
138.	наследование	Как называется механизм, позволяющий одному классу перенимать свойства и методы другого класса?	ОПК-2
139.	override	Какое ключевое слово используется для переопределения виртуальной функции в производном классе для контроля компилятором?	ОПК-2
140.	this	Как называется скрытый указатель внутри методов класса, который ссылается на текущий экземпляр объекта?	ОПК-2
141.	инкапсуляция	Как называется свойство объектно-ориентированного программирования, заключающееся в скрытии внутренней реализации объекта?	ОПК-2
142.	vector	Какая стандартная библиотека с++ предоставляет контейнер для работы с динамическими массивами?	ОПК-2
143.	итераторы	Как называются объекты-посредники в stl, которые используются для навигации по элементам контейнеров?	ОПК-2
144.	throw	Какое ключевое слово используется для генерации (выброса) исключения в блоке кода?	ОПК-2
145.	bool	Как называется тип данных в с++, который может принимать только значения true или false?	ОПК-2
146.	nullptr	Как называется специальное значение для указателя, который ни на что не ссылается (введено в с++11)?	ОПК-2
147.	перегрузка	Как называется возможность функций иметь одинаковые имена, но разные наборы аргументов?	ОПК-2

148.	абстрактный	Как называется класс, на основе которого нельзя создать объект из-за наличия в нем чисто виртуальных функций?	ОПК-2
149.	fstream	Какая библиотека используется для работы с файловыми потоками ввода и вывода?	ОПК-2
150.	разрешение	Как называется оператор '::', используемый для указания принадлежности функции к классу или пространству имен?	ОПК-2
151.	raii	Как называется идиома управления ресурсами, основанная на захвате ресурса в конструкторе и освобождении в деструкторе?	ОПК-2
152.	list	Какой контейнер stl реализует структуру данных 'двусвязный список'?	ОПК-2
153.	шаблон	Как называется шаблонный механизм, позволяющий писать функции и классы, работающие с любыми типами данных?	ОПК-2
154.	const	Какое ключевое слово гарантирует, что метод класса не будет изменять состояние его полей?	ОПК-2
155.	деструкция	Как называется процесс удаления объекта и вызова его специальной завершающей функции?	ОПК-2
156.	std	Какое пространство имен является стандартным для всех базовых компонентов c++?	ОПК-2

2. Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала (контрольных точек), оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

3. Критерии оценивания компетенций заданий открытого типа

Оценка **«отлично»** выставляется студенту, если являются верными результаты демонстрации знаний современных информационных технологий и программных средств, в том числе отечественного производства, выбора современных информационных технологий и программных средств отечественного производства, применения принципов работы современных информационных технологий и программных средств, использования современных информационных технологий и программных средств при решении профессиональных задач; представленный документ не содержит ошибок и решены все задачи. Компетенция ОПК-2 освоена на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если являются верными результаты демонстрации знаний современных информационных технологий и программных средств, в том числе отечественного производства, выбора современных информационных технологий и программных средств отечественного производства, применения принципов работы современных информационных технологий и программных средств, использования современных информационных технологий и программных средств при решении профессиональных задач; отчетный документ содержит незначительные ошибки, не влияющие на общий результат и не представлены отдельные задачи, не влияющие на общий результат. Компетенция ОПК-2 освоена на среднем уровне.

Оценка **«удовлетворительно»** выставляется студенту, если в основном верными являются результаты демонстрации знаний современных информационных технологий и программных средств, в том числе отечественного производства, выбора современных информационных технологий и программных средств отечественного производства, применения принципов работы современных информационных технологий и программных средств, использования современных информационных технологий и программных средств при решении профессиональных задач; отчетный документ содержит ошибки, не позволяющие оценить качество разработки; представлены не все данные и задачи решены не в полном объеме. Компетенция ОПК-2 освоена на минимальном уровне.

Оценка **«неудовлетворительно»** выставляется студенту, если являются неверными результаты демонстрации знаний современных информационных технологий и программных средств, в том числе отечественного производства, выбора современных информационных технологий и программных средств отечественного производства, применения принципов работы современных информационных технологий и программных средств, использования современных информационных технологий и программных средств при решении профессиональных задач; представленный документ содержит много грубых ошибок, не представлены ключевые данные и задачи решены недостаточно полно. Компетенция ОПК-2 не сформирована.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка **«отлично»** выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенция ОПК-2 освоена на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенция ОПК-2 освоена на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенция ОПК-2 освоена на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенция ОПК-2 не сформирована.