

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Заболотный, Глеб Иванович

Должность: Директор филиала

Дата подписания: 28.10.2023 15:50:13

Уникальный программный ключ:

476db7d4accb36ef8130172be235477473d63457266ce26b7e9e40f733b8b08

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Самарский государственный технический университет»

(ФГБОУ ВО «СамГТУ»)

УТВЕРЖДАЮ:

Директор филиала ФГБОУ ВО
"СамГТУ" в г. Новокуйбышевске

_____ / Г.И. Заболотни

" ____ " _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.03.05 «Технологии программирования»

**Код и направление подготовки
(специальность)**

09.03.01 Информатика и вычислительная
техника

Направленность (профиль)

Информатика и вычислительная техника в
нефтехимическом производстве

Квалификация

Бакалавр

Форма обучения

Заочная

Год начала подготовки

2023

Институт / факультет

Кафедры филиала ФГБОУ ВО "СамГТУ" в г.
Новокуйбышевске

Выпускающая кафедра

кафедра "Информатика и системы
управления" (НФ-ИиСУ)

Кафедра-разработчик

кафедра "Информатика и системы
управления" (НФ-ИиСУ)

Объем дисциплины, ч. / з.е.

108 / 3

**Форма контроля (промежуточная
аттестация)**

Экзамен

Б1.О.03.05 «Технологии программирования»

Рабочая программа дисциплины разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) **09.03.01 Информатика и вычислительная техника**, утвержденного приказом Министерства образования и науки РФ от № 929 от 19.09.2017 и соответствующего учебного плана.

Разработчик РПД:

Доцент, кандидат
экономических наук, доцент

(должность, степень, ученое звание)

А.В Волкодаева

(ФИО)

Заведующий кафедрой

А.В. Волкодаева, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

СОГЛАСОВАНО:

Председатель методического совета
факультета / института (или учебно-
методической комиссии)

Е.Т Демидова, кандидат
юридических наук, доцент

(ФИО, степень, ученое звание)

Руководитель образовательной
программы

А.В. Волкодаева, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы	4
2. Место дисциплины (модуля) в структуре образовательной программы	5
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся	6
4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий	6
4.1 Содержание лекционных занятий	7
4.2 Содержание лабораторных занятий	7
4.3 Содержание практических занятий	7
4.4. Содержание самостоятельной работы	8
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)	9
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения	10
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем	10
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)	11
9. Методические материалы	11
10. Фонд оценочных средств по дисциплине (модулю)	13

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2 Применяет современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	Владеть навыками применения технологий программирования, при решении задач профессиональной деятельности
			Знать методы применения технологий программирования, при решении задач профессиональной деятельности
			Уметь применять технологии программирования, при решении задач профессиональной деятельности
	ОПК-6 Способен разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием	ОПК-6.1 Выявляет потребности организации в компьютерном и сетевом оборудовании	Владеть навыками использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием

			Знать методы использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием
			Уметь использовать технологии программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием
	ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Владеть навыками разработки алгоритмов и программ, пригодных для практического применения
			Знать методы разработки алгоритмов и программ, пригодных для практического применения
			Уметь разрабатывать алгоритмы и программы, пригодные для практического применения
		ОПК-8.3 Тестирует работоспособность программы	Владеть навыками тестирования работоспособности программ
			Знать методы тестирования работоспособности программ
			Уметь тестировать работоспособность программ

2. Место дисциплины (модуля) в структуре образовательной программы

Место дисциплины (модуля) в структуре образовательной программы: **обязательная часть**

Код компетенции	Предшествующие дисциплины	Параллельно осваиваемые дисциплины	Последующие дисциплины
ОПК-2	Информационные технологии и программирование		Выполнение и защита выпускной квалификационной работы; Операционные системы; Учебная практика: технологическая (проектно-технологическая) практика

ОПК-6			Аппаратные средства вычислительной техники; Выполнение и защита выпускной квалификационной работы; Компьютерные сети и коммуникации; Организация производства на предприятиях отрасли
ОПК-8		Языки и методы программирования	Выполнение и защита выпускной квалификационной работы; Языки и методы программирования

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Вид учебной работы	Всего часов / часов в электронной форме	3 семестр часов / часов в электронной форме
Аудиторная контактная работа (всего), в том числе:	6	6
Лекции	2	2
Практические занятия	4	4
Самостоятельная работа (всего), в том числе:	93	93
подготовка к лекциям	30	30
подготовка к практическим занятиям	43	43
подготовка к экзамену	20	20
Контроль	9	9
Итого: час	108	108
Итого: з.е.	3	3

4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы				
		ЛЗ	ЛР	ПЗ	СРС	Всего часов
1	Фундаментальные концепции и принципы программирования	1	0	2	45	48
2	Структуры данных и алгоритмы, их практическое применение	1	0	2	48	51

	Контроль	0	0	0	0	9
	Итого	2	0	4	93	108

4.1 Содержание лекционных занятий

№ занятия	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
3 семестр				
1	Фундаментальные концепции и принципы программирования	Основные парадигмы программирования: структурная и объектно-ориентированная. Механизмы реализации абстракции и управления ресурсами	Работа с файлами. Создание. Чтение. Запись. Работа с файловыми переменными. Наследование.	1
2	Структуры данных и алгоритмы, их практическое применение	Абстрактные типы данных и их реализации. Организация данных в памяти и при работе с внешними ресурсами	Динамические переменные. Операторы управления динамической памятью. Функции управления динамической памятью низкого уровня. Виртуальные функции. Абстрактные классы. Стандартная библиотека шаблонов. Алгоритмы. Адаптеры. Примитивы управления памятью. Основные компоненты. Итераторы. Функциональные объекты. Распределители. Контейнеры. Итераторы потоков	1
Итого за семестр:				2
Итого:				2

4.2 Содержание лабораторных занятий

Учебные занятия не реализуются.

4.3 Содержание практических занятий

№ занятия	Наименование раздела	Тема практического занятия	Содержание практического занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
3 семестр				

1	Фундаментальные концепции и принципы программирования	Основные парадигмы программирования: структурная и объектно-ориентированная Механизмы реализации абстракции и управления ресурсами	Создание бинарных (двоичных) файлов Работа с бинарными (двоичными) файлами. Динамические массивы. Информационные динамические структуры. Создание и работа с однонаправленным списком. Информационные динамические структуры. Двухнаправленный список. Создание. Работа. Информационные динамические структуры. Двухнаправленный список. Удаление списка и его восстановление. Программирование с использованием однонаправленных списков типа «СТЕК». Наследование и виртуальные функции.	2
2	Структуры данных и алгоритмы, их практическое применение	Абстрактные типы данных и их реализации. Организация данных в памяти и при работе с внешними ресурсами	Работа с текстовыми файлами. Структура. Создание. Редактирование данных. Классы и объекты в C++. Перегрузка операций. Стандартная библиотека шаблонов (STL)	2
Итого за семестр:				4
Итого:				4

4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
3 семестр			
Фундаментальные концепции и принципы программирования	подготовка к лекциям	Работа с файлами. Создание. Чтение. Запись. Работа с файловыми переменными. Наследование.	15
Фундаментальные концепции и принципы программирования	подготовка к экзамену	Работа с файлами. Создание. Чтение. Запись. Работа с файловыми переменными. Наследование.	10
Фундаментальные концепции и принципы программирования	подготовка к практическим занятиям	Создание бинарных (двоичных) файлов Работа с бинарными (двоичными) файлами. Динамические массивы. Информационные динамические структуры. Создание и работа с однонаправленным списком. Информационные динамические структуры. Двухнаправленный список. Создание. Работа. Информационные динамические структуры. Двухнаправленный список. Удаление списка и его восстановление. Программирование с использованием однонаправленных списков типа «СТЕК». Наследование и виртуальные функции.	20

Структуры данных и алгоритмы, их практическое применение	подготовка к лекциям	Динамические переменные. Операторы управления динамической памятью. Функции управления динамической памятью низкого уровня. Виртуальные функции. Абстрактные классы. Стандартная библиотека шаблонов. Алгоритмы. Адаптеры. Примитивы управления памятью. Основные компоненты. Итераторы. Функциональные объекты. Распределители. Контейнеры. Итераторы потоков	15
Структуры данных и алгоритмы, их практическое применение	подготовка к экзамену	Динамические переменные. Операторы управления динамической памятью. Функции управления динамической памятью низкого уровня. Виртуальные функции. Абстрактные классы. Стандартная библиотека шаблонов. Алгоритмы. Адаптеры. Примитивы управления памятью. Основные компоненты. Итераторы. Функциональные объекты. Распределители. Контейнеры. Итераторы потоков	10
Структуры данных и алгоритмы, их практическое применение	подготовка к практическим занятиям	Работа с текстовыми файлами. Структура. Создание. Редактирование данных. Классы и объекты в C++. Перегрузка операций. Стандартная библиотека шаблонов (STL)	23
Итого за семестр:			93
Итого:			93

5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)

№ п/п	Библиографическое описание	Ресурс НТБ СамГТУ (ЭБС СамГТУ, IPRbooks и т.д.)
Основная литература		
1	Интеллектуальный анализ данных: учебное пособие / Замятин А.В., Издательский Дом Томского государственного университета: 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 116889	Электронный ресурс
2	Методы хранения и анализа данных: учебное пособие / Кумратова А.М., Ай Пи Ар Медиа: 2022.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 119065	Электронный ресурс
3	Объектно ориентированное программирование на языке Python: учебное пособие / Букунов С.В., Букунова О.В., Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ: 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 117194	Электронный ресурс

4	Объектно-ориентированное программирование: учебное пособие / Логанов С.В., Моругин С.Л., Профобразование, Ай Пи Ар Медиа: 2022.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 118969	Электронный ресурс
Дополнительная литература		
5	Анализ данных: учебно-методическое пособие / Шнарева Г.В., Пономарева Ж.Г., Университет экономики и управления: 2019.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 89482	Электронный ресурс
6	Объектно-ориентированное программирование. В 3-х частях. Ч.1: учебное пособие / Степанов П.П., Кабанов А.А., Никонов В.А., Павлюченко Т.С., Омский государственный технический университет: 2021.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 124850	Электронный ресурс
7	Основы анализа данных: учебное пособие / Гнатюк С.П., Санкт-Петербургский государственный университет промышленных технологий и дизайна: 2019.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 102939	Электронный ресурс

Доступ обучающихся к ЭР НТБ СамГТУ (elib.samgtu.ru) осуществляется посредством электронной информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

При проведении лекционных занятий используется мультимедийное оборудование.

Организовано взаимодействие обучающегося и преподавателя с использованием электронной ин-формационной образовательной среды университета.

№ п/п	Наименование	Производитель	Способ распространения
1	Anaconda 3 (язык Python, jupyter)	Anaconda Equipment (Зарубежный)	Свободно распространяемое
2	Microsoft Office	Microsoft (Зарубежный)	Лицензионное
3	Образовательная платформа «Юрайт»	ООО «ЭЛЕКТРОННОЕ ИЗДАТЕЛЬСТВО ЮРАЙТ» (Отечественный)	Лицензионное
4	МойОфис Образование	ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ» (Отечественный)	Лицензионное
5	Visual Studio Code (VS Code)	Microsoft (Зарубежный)	Свободно распространяемое

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем

№ п/п	Наименование	Краткое описание	Режим доступа
-------	--------------	------------------	---------------

1	eLIBRARY.ru	http://www.eLIBRARY.ru/	Российские базы данных ограниченного доступа
2	КонсультантПлюс (правовые документы) - доступ с ПК в Медиацентре (ауд. 42)	http://www.consultant.ru/	Российские базы данных ограниченного доступа
3	Электронно-библиотечная система IPRbooks	http://www.iprbookshop.ru/	Российские базы данных ограниченного доступа
4	Электронная библиотека изданий СамГТУ	http://irbis.samgtu.local/cgi-bin/irbis64r_01/cgiirbis_64.exe	Российские базы данных ограниченного доступа

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия

Аудитория для лекционных, семинарских и практических занятий, текущего контроля и промежуточной аттестации (с мультимедийным оборудованием) укомплектованы мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Практические занятия

Аудитория для практических и семинарских занятий, текущего контроля и промежуточной аттестации, укомплектованные специализированной мебелью и техническими средствами обучения (проектор, экран, компьютер/ноутбук), с выходом в сеть Интернет и доступом в электронную информационно-образовательную среду СамГТУ. Аудитория оборудована специализированной мебелью: столы и стулья для обучающихся; стол и стул для преподавателя, доска.

- компьютерные классы (ауд. 101, 102, 201, 401, 404).

Самостоятельная работа

Помещения для самостоятельной работы оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде СамГТУ:

- кабинет для текущего контроля и промежуточной аттестации, курсового проектирования, групповых и индивидуальных консультаций ауд. 212;
- кабинет для самостоятельной работы, аудитория 304;
- компьютерные классы (ауд. 101, 102, 111, 201, 401, 404).

9. Методические материалы

Методические рекомендации при работе на лекции

До лекции студент должен просмотреть учебно-методическую и научную литературу по теме лекции с тем, чтобы иметь представление о проблемах, которые будут разбираться в лекции.

Перед началом лекции обучающимся сообщается тема лекции, план, вопросы, подлежащие рассмотрению, доводятся основные литературные источники. Весь учебный материал, сообщаемый преподавателем, должен не просто прослушиваться. Он должен быть активно воспринят, т.е. услышан, осмыслен, понят, зафиксирован на бумаге и закреплён в памяти. Приступая к слушанию нового учебного

материала, полезно мысленно установить его связь с ранее изученным. Следя за техникой чтения лекции (акцент на существенном, повышение тона, изменение ритма, пауза и т.п.), необходимо вслед за преподавателем уметь выделять основные категории, законы и определять их содержание, проблемы, предполагать их возможные решения, доказательства и выводы. Осуществляя такую работу, можно значительно облегчить себе понимание учебного материала, его конспектирование и дальнейшее изучение.

Конспектирование лекции позволяет обработать, систематизировать и лучше сохранить полученную информацию с тем, чтобы в будущем можно было восстановить в памяти основные, содержательные моменты. Типичная ошибка, совершаемая обучающимся, дословное конспектирование речи преподавателя. Как правило, при записи «слово в слово» не остается времени на обдумывание, анализ и синтез информации. Отбирая нужную информацию, главные мысли, проблемы, решения и выводы, необходимо сокращать текст, строить его таким образом, чтобы потом можно было легко в нем разобраться. Желательно оставить в рабочих конспектах поля, на которых можно будет делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С окончанием лекции работа над конспектом не может считаться завершенной. Нужно еще восстановить отдельные места, проверить, все ли понятно, уточнить что-то на консультации и т.п. с тем, чтобы конспект мог быть использован в процессе подготовки к практическим занятиям, зачету, экзамену. Конспект лекции – незаменимый учебный документ, необходимый для самостоятельной работы.

Методические рекомендации при подготовке и работе на практическом занятии

Практические занятия по дисциплине проводятся в целях выработки практических умений и приобретения навыков в решении профессиональных задач.

Рекомендуется следующая схема подготовки к практическому занятию:

1. ознакомление с планом практического занятия, который отражает содержание предложенной темы;
2. проработка конспекта лекции;
3. чтение рекомендованной литературы;
4. подготовка ответов на вопросы плана практического занятия;
5. выполнение тестовых заданий, задач и др.

Подготовка обучающегося к практическому занятию производится по вопросам, разработанным для каждой темы практических занятий и (или) лекций. В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы.

Работа студентов во время практического занятия осуществляется на основе заданий, которые выдаются обучающимся в начале или во время занятия. На практических занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительный материал по тематике занятий. Обучающимся необходимо обращать внимание на основные понятия, алгоритмы, определять практическую значимость рассматриваемых вопросов. На практических занятиях обучающиеся должны уметь выполнить расчет по заданным параметрам или выработать определенные решения по обозначенной проблеме. Задания могут быть групповые и индивидуальные. В зависимости от сложности предлагаемых заданий, целей занятия, общей подготовки обучающихся преподаватель может подсказать обучающимся алгоритм решения или первое действие, или указать общее направление рассуждений. Полученные результаты обсуждаются с позиций их адекватности или эффективности в рассмотренной ситуации.

Методические рекомендации по выполнению самостоятельной работы

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению с учетом потребностей и возможностей обучающегося.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения

дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий;
- на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания;
- на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.;
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

10. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств представлен в приложении № 1.

Приложение 1 к рабочей программе дисциплины
Б1.О.03.05 «Технологии программирования»

**Фонд оценочных средств
по дисциплине
Б1.О.03.05 «Технологии программирования»**

Код и направление подготовки (специальность)	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Информатика и вычислительная техника в нефтехимическом производстве
Квалификация	Бакалавр
Форма обучения	Заочная
Год начала подготовки	2023
Институт / факультет	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
Выпускающая кафедра	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Кафедра-разработчик	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Объем дисциплины, ч. / з.е.	108 / 3
Форма контроля (промежуточная аттестация)	Экзамен

**Перечень планируемых результатов обучения по дисциплине (модулю),
соотнесенных с планируемыми результатами освоения образовательной
программы**

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2 Применяет современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	Владеть навыками применения технологий программирования, при решении задач профессиональной деятельности
			Знать методы применения технологий программирования, при решении задач профессиональной деятельности
			Уметь применять технологии программирования, при решении задач профессиональной деятельности
	ОПК-6 Способен разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием	ОПК-6.1 Выявляет потребности организации в компьютерном и сетевом оборудовании	Владеть навыками использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием

			Знать методы использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием
			Уметь использовать технологии программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием
	ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Владеть навыками разработки алгоритмов и программ, пригодных для практического применения
			Знать методы разработки алгоритмов и программ, пригодных для практического применения
			Уметь разрабатывать алгоритмы и программы, пригодные для практического применения
		ОПК-8.3 Тестирует работоспособность программы	Владеть навыками тестирования работоспособности программ
			Знать методы тестирования работоспособности программ
			Уметь тестировать работоспособность программ

Матрица соответствия оценочных средств запланированным результатам обучения

Код индикатора достижения компетенции	Результаты обучения	Оценочные средства	Текущий контроль успеваемости	Промежуточная аттестация
Фундаментальные концепции и принципы программирования				

ОПК-2.2 Применяет современные информационных технологий и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	Владеть навыками применения технологий программирования, при решении задач профессиональной деятельности	оценочные средства промежуточного контроля	Нет	Да
	Уметь применять технологии программирования, при решении задач профессиональной деятельности	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения технологий программирования, при решении задач профессиональной деятельности	оценочные средства промежуточного контроля	Нет	Да
		тест	Да	Нет
	Владеть навыками применения технологий программирования, при решении задач профессиональной деятельности	практические задачи	Да	Нет
	Уметь применять технологии программирования, при решении задач профессиональной деятельности	практические задачи	Да	Нет
ОПК-6.1 Выявляет потребности организации в компьютерном и сетевом оборудовании	Уметь использовать технологии программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	практические задачи	Да	Нет
	Владеть навыками использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	практические задачи	Да	Нет
	Знать методы использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	тест	Да	Нет
	Владеть навыками использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	оценочные средства промежуточного контроля	Нет	Да
	Знать методы использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	оценочные средства промежуточного контроля	Нет	Да
	Уметь использовать технологии программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	оценочные средства промежуточного контроля	Нет	Да
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Знать методы разработки алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками разработки алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Уметь разрабатывать алгоритмы и программы, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да

	Знать методы разработки алгоритмов и программ, пригодных для практического применения	тест	Да	Нет
	Владеть навыками разработки алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Уметь разрабатывать алгоритмы и программы, пригодные для практического применения	практические задачи	Да	Нет
ОПК-8.3 Тестирует работоспособность программы	Владеть навыками тестирования работоспособности программ	практические задачи	Да	Нет
	Уметь тестировать работоспособность программ	практические задачи	Да	Нет
	Знать методы тестирования работоспособности программ	тест	Да	Нет
		оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками тестирования работоспособности программ	оценочные средства промежуточного контроля	Нет	Да
	Уметь тестировать работоспособность программ	оценочные средства промежуточного контроля	Нет	Да
Структуры данных и алгоритмы, их практическое применение				
ОПК-2.2 Применяет современные информационных технологий и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	Владеть навыками применения технологий программирования, при решении задач профессиональной деятельности	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения технологий программирования, при решении задач профессиональной деятельности	оценочные средства промежуточного контроля	Нет	Да
	Уметь применять технологии программирования, при решении задач профессиональной деятельности	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения технологий программирования, при решении задач профессиональной деятельности	тест	Да	Нет
	Владеть навыками применения технологий программирования, при решении задач профессиональной деятельности	практические задачи	Да	Нет
	Уметь применять технологии программирования, при решении задач профессиональной деятельности	практические задачи	Да	Нет
ОПК-6.1 Выявляет потребности организации в компьютерном и сетевом оборудовании	Уметь использовать технологии программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	практические задачи	Да	Нет

	Владеть навыками использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	практические задачи	Да	Нет
	Знать методы использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	тест	Да	Нет
	Уметь использовать технологии программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	оценочные средства промежуточного контроля	Нет	Да
	Знать методы использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием	оценочные средства промежуточного контроля	Нет	Да
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Уметь разрабатывать алгоритмы и программы, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками разработки алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Знать методы разработки алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
		тест	Да	Нет
	Владеть навыками разработки алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Уметь разрабатывать алгоритмы и программы, пригодные для практического применения	практические задачи	Да	Нет
ОПК-8.3 Тестирует работоспособность программы	Уметь тестировать работоспособность программ	практические задачи	Да	Нет
	Владеть навыками тестирования работоспособности программ	практические задачи	Да	Нет
	Знать методы тестирования работоспособности программ	тест	Да	Нет
	Уметь тестировать работоспособность программ	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками тестирования работоспособности программ	оценочные средства промежуточного контроля	Нет	Да
	Знать методы тестирования работоспособности программ	оценочные средства промежуточного контроля	Нет	Да

Типовые задания для промежуточной аттестации по дисциплине
Б1.О.03.05 Технологии программирования
(шифр и наименование дисциплины)

для направления подготовки 09.03.01 Информатика и вычислительная техника

(шифр и наименование направления подготовки, специальности)

2023 ГОД ПРИЕМА

(год приема на образовательную программу)

Контролируемая (ые) компетенция(и):

ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решения задач профессиональной деятельности

ОПК-6 Способен разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием

ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

(шифр и наименование компетенции(й))

Спецификация тестовых заданий

Содержание дисциплины (разделы / темы)	Число заданий									
	закрытые			открытые				комбинированные	всего	
	однозначный выбор варианта ответа	многозначный выбор варианта ответа	задание на сопоставление	задание на установление правильной последовательности	задания на дополнение	задания с развернутым ответом	практико-ориентированные задания	Задания с выбором одного ответа и обоснованием выбора ответа	Задания с выбором нескольких ответов и обоснованием выбора ответов	
Раздел 1. Фундаментальные концепции и принципы программирования	20	6	2	8	7	7	10	8	6	74
Тема 1. Основные парадигмы программирования: структурная и объектно-ориентированная	13	4	2	4	4	6	8	5	5	51
Тема 2. Механизмы реализации абстракции и управления ресурсами	7	2		4	3	1	2	3	1	23
Раздел 2. Структуры данных и алгоритмы, их практическое применение	19	7	4	8	6	12	11	8	9	84
Тема 3. Абстрактные типы данных и их реализации	11	5	4	4	4	3	11	6	5	53
Тема 4. Организация данных в памяти и при работе с внешними ресурсами	8	2		4	2	9		2	4	31
Итого	39	13	6	16	13	19	21	16	15	158

Количество заданий в комплекте оценочных материалов

Код компетенции	Наименование компетенции	Количество заданий
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	54
ОПК-6	Способен разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием	46
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	58

Сценарии выполнения диагностических заданий

Тип задания	Последовательность действий при выполнении задания
Задание закрытого типа с однозначным выбором варианта ответа	1. Внимательно прочитать текст задания. 2. Выбрать единственный вариант ответа из предложенных.
Задание закрытого типа с многозначным выбором вариантов ответа	1. Внимательно прочитать текст задания. 2. Выбрать несколько вариантов ответа из предложенных.
Задание закрытого типа на установление соответствия	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидаются пары элементов. 2. Внимательно прочитать оба списка: список 1 - вопросы, утверждения, факты, понятия и т.д.; список 2 - утверждения, свойства объектов и т.д. 3. Сопоставить элементы списка 1 с элементами списка 2, сформировать пары элементов. 4. Записать буквы вариантов ответа (например, АБВГ)
Задание закрытого типа на установление последовательности	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается последовательность элементов. 2. Внимательно прочитать предложенные варианты ответа. 3. Построить верную последовательность из предложенных элементов. 4. Записать буквы вариантов ответа в нужной последовательности без пробелов и знаков препинания (например, БВА)
Задание открытого типа на дополнение	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается недостающее дополнение. 2. Определить какой информации не хватает. 3. Внесение пропущенного слова. 4. Записать в ответ только дополнение.
Задание открытого типа с развернутым ответом	1. Внимательно прочитать текст задания и понять суть вопроса. 2. Продумать логику и полноту ответа. 3. Записать ответ, используя четкие компактные формулировки. 4. В случае расчетной задачи записать решение и ответ.
Задание комбинированного типа: практико-ориентированные задания	1. Внимательно прочитать текст задания. 2. Выполните указанные в заданиях действия
Задание комбинированного типа с выбором одного ответа и обоснованием выбора ответа	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается только один из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать один ответ, наиболее верный. 4. Записать только букву выбранного варианта ответа. 5. Записать аргументы, обосновывающие выбор ответа
Задание комбинированного типа с выбором нескольких ответов и обоснованием выборов ответов	1. Внимательно прочитать текст задания и понять, что в качестве ответа ожидается несколько из предложенных вариантов. 2. Внимательно прочитать предложенные варианты ответа. 3. Выбрать несколько верных вариантов ответов. 4. Записать последовательно буквы выбранных вариантов без пробелов и знаков препинания (например, АБВ). 5. Записать аргументы, обосновывающие выбор каждого из ответов

Система оценивания заданий

Указания по оцениванию	Результат оценивания (баллы, полученные за выполнение задания / характеристика правильности ответа)
Задание закрытого типа с однозначным выбором варианта ответа считается верным, если правильно определен вариант ответа	За правильный вариант ответа начисляется 1 балл

Задание закрытого типа с многозначным выбором вариантов ответа считается верным, если правильно определены все варианты ответа	За правильный вариант ответа начисляется 1 балл
Задание закрытого типа на установление соответствия считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)	Количество баллов определяется числом пар для сопоставления. За каждое правильно установленное соответствие начисляется 1 балл.
Задание закрытого типа на установление последовательности считается верным, если правильно указана вся последовательность цифр	Максимальный балл определяется количеством элементов в последовательности. В случае ошибки в одном месте - снижение на один балл. За каждое правильно указанное место элемента в последовательности начисляется 1 балл.
Задание открытого типа на дополнение, где предоставляется предложение или фрагмент текста, в котором пропущено одно или несколько слов или фраз. Задача состоит в том, чтобы заполнить пропуски, восстановив тем самым исходный смысл предложения.	2 балла засчитывается, если студент вписал правильный ответ в соответствии с ключом. 1 балл может быть засчитан за близкий к правильному ответ, если он демонстрирует частичное понимание.
Задание открытого типа с развернутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте	Максимальный балл - 4. Студент может получить 4 балла за полный и правильный ответ, логично изложенный и с корректной терминологией, или меньше за неполные или неточно сформулированные ответы. Полнота (1 балл), Правильность (1 балл), Логичность (1 балл), Терминология (1 балл).
Задание комбинированного типа с выбором одного ответа и обоснованием выбора ответа считается верным, если правильно указана цифра и приведены корректные аргументы, используемые при выборе ответа	За правильный выбор ответа начисляется 1 балл. За качественное обоснование - еще 2-3 балла. Критерии оценивания обоснования должны быть четко определены (например, логичность, полнота, использование фактов). Неправильный выбор ответа - 0 баллов, даже если обоснование частично верное.
Задание комбинированного типа с выбором нескольких вариантов ответа и обоснованием выбора ответа считается верным, если правильно указана цифра и приведены корректные аргументы, используемые при выборе ответа	За правильный выбор ответа начисляется 1 балл. За качественное обоснование - еще 2-3 балла. Критерии оценивания обоснования должны быть четко определены (например, логичность, полнота, использование фактов). Неправильный выбор ответа - 0 баллов, даже если обоснование частично верное.

Тестовые задания с ключами ответов

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решения задач профессиональной деятельности					
1.	Какой из перечисленных принципов НЕ относится к парадигме структурного программирования? 1) Использование трех базовых конструкций: следование, ветвление, цикл. 2) Принцип нисходящего проектирования и декомпозиции. 3) Скрытие данных и их внутренней реализации внутри объектов. 4) Минимизация или отказ от оператора безусловного перехода (GOTO).	3	Задание закрытого типа с однозначным выбором варианта ответа	1	1
2.	Какой механизм языка C++ является ключевым для реализации полиморфизма во время выполнения (динамического)? 1) Перегрузка функций (function overloading). 2) Виртуальные функции и механизм позднего связывания. 3) Использование шаблонов (templates). 4) Перегрузка операторов.	2	Задание закрытого типа с однозначным выбором варианта ответа	2	2
3.	Какая структура данных наиболее оптимальна для реализации алгоритма, требующего обработки данных в порядке "последним пришел – первым ушел" (LIFO)? 1) Двухнаправленный список. 2) Очередь (FIFO). 3) Динамический массив. 4) Стек.	4	Задание закрытого типа с однозначным выбором варианта ответа	1	3
4.	Для хранения коллекции уникальных элементов, к которой требуется часто выполнять операцию проверки на принадлежность (т.е. element in collection), какая встроенная структура данных Python будет наиболее эффективна по времени? 1) list (список) 2) tuple (кортеж) 3) dict (словарь), где элементами коллекции являются ключи 4) set (множество)	4	Задание закрытого типа с однозначным выбором варианта ответа	2	3
5.	Что из перечисленного является основной целью использования принципа наследования в объектно-ориентированном	3	Задание закрытого типа с однозначным выбором варианта ответа	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>программировании?</p> <p>1) Ускорение выполнения программы.</p> <p>2) Уменьшение объема занимаемой памяти.</p> <p>3) Создание иерархии классов, позволяющей повторно использовать код (базовый класс) и расширять/специализировать функциональность (производные классы).</p> <p>4) Объединение нескольких функций в одну.</p>				
6.	<p>Какая операция управления динамической памятью в языке C++ непосредственно связана с риском возникновения "утечки памяти" (memory leak)?</p> <p>1) Объявление локальной переменной в стеке.</p> <p>2) Выделение памяти с помощью new без последующего освобождения с помощью delete.</p> <p>3) Освобождение памяти с помощью delete.</p> <p>4) Объявление глобальной переменной.</p>	2	Задание закрытого типа с однозначным выбором варианта ответа	1	2
7.	<p>Что такое "итератор" (iterator) в контексте языка Python?</p> <p>1) Это специальный синтаксис цикла for.</p> <p>2) Это объект, который предоставляет последовательный доступ к элементам какой-либо коллекции (__iter__() и __next__()).</p> <p>3) Это функция для преобразования последовательности, такая как map() или filter().</p> <p>4) Это то же самое, что и "итерируемый объект" (iterable)</p>	2	Задание закрытого типа с однозначным выбором варианта ответа	2	4
8.	<p>Какие из перечисленных принципов являются базовыми конструкциями структурного программирования (по Дейкстре)?</p> <p>1) Следование (sequence).</p> <p>2) Рекурсия (recursion).</p> <p>3) Ветвление (selection).</p> <p>4) Параллельное выполнение (concurrency).</p> <p>5) Цикл (iteration).</p>	1, 3, 5	Задание закрытого типа с многозначным выбором вариантов ответа	1	1
9.	<p>Какие из перечисленных концепций являются основными принципами (столпами) объектно-ориентированного программирования?</p>	1, 3, 5	Задание закрытого типа с многозначным выбором вариантов ответа	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	1) Инкапсуляция. 2) Структурирование. 3) Наследование. 4) Декомпозиция. 5) Полиморфизм.				
10.	Какие из следующих элементов входят в Стандартную библиотеку шаблонов (STL) C++? 1) Контейнеры (например, vector, map). 2) Алгоритмы (например, sort, find). 3) Виртуальные функции. 4) Итераторы. 5) Умные указатели (например, unique_ptr).	1, 2, 4, 5	Задание закрытого типа с многозначным выбором вариантов ответа	2	4
11.	Какие из перечисленных утверждений об абстрактных методах и классах в Python являются верными? 1) Они позволяют реализовать полиморфизм и задать интерфейс, который должны реализовать дочерние классы. 2) Создаются с использованием встроенных ключевых слов abstract и virtual. 3) Абстрактный метод — это метод, объявленный в базовом классе, но не имеющий реализации в нем. 4) Класс становится абстрактным, если он наследуется от класса ABC (Abstract Base Class) и содержит хотя бы один метод, декорированный как @abstractmethod. 5) Нельзя создать экземпляр класса, содержащего абстрактные методы.	1, 3, 4, 5	Задание закрытого типа с многозначным выбором вариантов ответа	2	2
12.	Какие из перечисленных структур данных являются динамическими (размер может изменяться во время выполнения)? 1) Статический массив (например, int arr[100]). 2) Динамический массив (например, созданный через new int[n]). 3) Однонаправленный связный список. 4) Стек, реализованный на статическом массиве. 5) Двухнаправленный связный список.	2, 3, 5	Задание закрытого типа с многозначным выбором вариантов ответа	1	3
13.	Какие из следующих действий могут привести к неопределенному поведению (undefined behavior) в C++? 1) Разыменование нулевого (nullptr)	1, 2, 3, 4, 5	Задание закрытого типа с многозначным выбором вариантов ответа	3	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>указателя.</p> <p>2) Выход за границы массива.</p> <p>3) Двойное освобождение одной и той же области динамической памяти (delete дважды).</p> <p>4) Использование переменной до ее инициализации.</p> <p>5) Чтение из файла, открытого в режиме записи.</p>				
14.	<p>Какие из перечисленных возможностей предоставляет механизм наследования в C++?</p> <p>1) Повторное использование кода базового класса.</p> <p>2) Соккрытие реализации базового класса.</p> <p>3) Создание иерархии типов (отношение "является").</p> <p>4) Увеличение скорости выполнения программы.</p> <p>5) Реализация полиморфного поведения через указатели/ссылки на базовый класс.</p>	1, 3, 5	Задание закрытого типа с многозначным выбором вариантов ответа	2	2
15.	<p>Установите соответствие между базовой конструкцией структурного программирования и её описанием.</p> <p>А) Следование</p> <p>Б) Ветвление</p> <p>В) Цикл</p> <p>1) Выполнение одной из нескольких последовательностей команд в зависимости от условия.</p> <p>2) Многократное повторение одной и той же последовательности команд до выполнения условия.</p> <p>3) Последовательное выполнение команд одна за другой.</p>	А3, Б1, В2	Задание закрытого типа на установление соответствия	1	1
16.	<p>Установите соответствие между принципом ООП и его определением.</p> <p>А) Инкапсуляция</p> <p>Б) Наследование</p> <p>В) Полиморфизм</p> <p>1) Механизм создания нового класса на основе существующего с возможностью переопределения и расширения его функциональности.</p> <p>2) Возможность объектов с одинаковым интерфейсом иметь разную реализацию.</p> <p>3) Объединение данных и методов для работы с ними в единый объект с ограничением прямого доступа к внутреннему состоянию.</p>	А3, Б1, В2	Задание закрытого типа на установление соответствия	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
17.	<p>Установите соответствие между динамической структурой данных и её свойством.</p> <p>А) Однонаправленный список Б) Двухнаправленный список В) Стек (LIFO) Г) Очередь (FIFO)</p> <p>1) Позволяет эффективно добавлять и удалять элементы только с одного конца (вершины). 2) Каждый узел содержит указатель только на следующий элемент. 3) Позволяет эффективно добавлять в конец и удалять из начала. 4) Каждый узел содержит указатели на следующий и предыдущий элементы.</p>	A2, Б4, В1, Г3	Задание закрытого типа на установление соответствия	2	3
18.	<p>Установите соответствие между концепцией объектно-ориентированного программирования в Python и её типичной реализацией или описанием.</p> <p>А) Конкретный (реальный) класс Б) Абстрактный базовый класс (Abstract Base Class, ABC) В) Миксин (Mixin-класс)</p> <p>1) Класс, предназначенный для предоставления определенной функциональности через множественное наследование. Не предназначен для создания самостоятельных экземпляров и часто не имеет собственного состояния (инициализатора). 2) Обычный класс, от которого можно напрямую создавать объекты. Все его методы имеют реализацию. 3) Класс, наследующийся от abc.ABC, содержащий хотя бы один метод, декорированный @abstractmethod. Нельзя создать его экземпляр. Используется для определения общего интерфейса.</p>	A2, Б3, В1	Задание закрытого типа на установление соответствия	2	4
19.	<p>Установите последовательность этапов алгоритма нисходящего проектирования (принцип структурного программирования):</p> <p>А) Декомпозиция подзадач на более простые функции. Б) Тестирование и отладка отдельных модулей. В) Определение главной, самой общей задачи. Г) Кодирование (реализация)</p>	ВАГБД	Задание закрытого типа на установление последовательности	2	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	простейших функций. Д) Интеграция модулей в единую программу.				
20.	Установите последовательность действий для безопасной работы с динамическим массивом (int) в C++: А) Использование массива (обращение к элементам). Б) Освобождение памяти оператором delete[]. В) Проверка указателя на равенство nullptr после выделения. Г) Выделение памяти оператором new int[N].	ГВАБ	Задание закрытого типа на установление последовательности	2	2
21.	Установите последовательность вызова конструкторов и деструкторов при создании и удалении объекта производного класса в C++: А) Конструктор производного класса. Б) Деструктор производного класса. В) Конструктор базового класса. Г) Деструктор базового класса.	ВАБГ	Задание закрытого типа на установление последовательности	2	2
22.	Установите последовательность операций при вставке нового узла в начало однонаправленного списка: А) Присвоить полю next нового узла адрес текущей "головы" списка. Б) Создать новый узел в динамической памяти. В) Записать данные в новый узел. Г) Сделать новый узел "головой" списка.	БВАГ	Задание закрытого типа на установление последовательности	2	3
23.	Установите последовательность этапов работы программы с бинарным файлом для чтения массива структур: А) Чтение данных в массив с помощью fread. Б) Закрытие файла (fclose). В) Определение размера файла и вычисление количества структур. Г) Выделение памяти под массив структур. Д) Открытие файла в режиме "rb" (fopen).	ДВГАБ	Задание закрытого типа на установление последовательности	2	4
24.	Установите последовательность действий для корректной обработки полиморфного объекта через указатель на базовый класс: А) Вызов виртуальных методов через указатель базового класса. Б) Объявление виртуального деструктора в базовом классе.	БГДАВ	Задание закрытого типа на установление последовательности	3	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	В) Удаление объекта через delete по указателю базового класса. Г) Динамическое создание объекта производного класса (new). Д) Присваивание адреса созданного объекта указателю базового типа.				
25.	Установите последовательность операций при работе со стеком (LIFO) на основе однонаправленного списка для добавления и последующего извлечения элемента: А) Извлечь данные из узла в голове списка (операция pop). Б) Удалить узел из головы списка, освободив память. В) Создать новый узел с данными. Г) Добавить новый узел в голову списка (операция push).	ВГАБ	Задание закрытого типа на установление последовательности	2	3
26.	Установите последовательность шагов при перегрузке оператора сложения (+) как метода класса в C++: А) Определить возвращаемый тип (часто объект того же класса). Б) Реализовать логику сложения текущего объекта и объекта-параметра. В) Объявить прототип оператора в публичной секции класса: MyClass operator+(const MyClass& other) const;. Г) Создать внутри метода временный объект для результата и вернуть его.	ВАБГ	Задание закрытого типа на установление последовательности	3	1
27.	Установите последовательность использования компонентов STL для сортировки и вывода содержимого вектора: А) Использовать алгоритм std::sort. Б) Создать вектор и заполнить его данными. В) Использовать итератор std::ostream_iterator для вывода в std::cout. Г) Использовать алгоритм std::copy.	БАГВ	Задание закрытого типа на установление последовательности	2	1
28.	В парадигме структурного программирования любая программа может быть построена с использованием трех базовых управляющих конструкций: следования, ветвления и _____.	цикла (или повтора) рения)	Задание открытого типа на дополнение	1	1
29.	Для работы с файлами в языке Python перед операциями чтения или записи файл, как правило,	открыть	Задание открытого типа на дополнение	1	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	рекомендуется _____ с помощью встроенной функции <code>open()</code>				
30.	Механизм, позволяющий производному классу использовать поля и методы базового класса, называется _____.	наследованием	Задание открытого типа на дополнение	1	1
31.	Функция, объявленная в базовом классе с помощью синтаксиса <code>= 0</code> , называется _____ виртуальной функцией и делает класс абстрактным.	чистой	Задание открытого типа на дополнение	2	2
32.	Указатель, который используется для доступа к элементам контейнеров STL и навигации по ним, называется _____.	итератором	Задание открытого типа на дополнение	2	2
33.	Принцип ООП, заключающийся в объединении данных и методов для работы с ними в единый объект и сокрытии внутренней реализации, называется _____.	инкапсуляцией	Задание открытого типа на дополнение	1	1
34.	Структура данных, в которой добавление и удаление элементов происходит по принципу "первым пришел — первым ушел" (FIFO), называется _____.	очередью	Задание открытого типа на дополнение	1	3
35.	Если память, выделенная с помощью оператора <code>new</code> , не была освобождена оператором <code>delete</code> , возникает ошибка, называемая "_____ памяти".	утечкой	Задание открытого типа на дополнение	1	4
36.	В бинарном файле данные хранятся в том же виде, что и в оперативной памяти, то есть в виде последовательности _____.	байтов	Задание открытого типа на дополнение	1	4
37.	Объясните, в чем состоит принцип «нисходящего проектирования» в структурном программировании.	Нисходящее проектирование — это метод разработки, при котором сложная задача последовательно разбивается на более простые подзадачи.	Задание открытого типа с развернутым ответом	2	1
38.	Опишите главное различие в работе с текстовыми и бинарными файлами на уровне операций ввода-вывода в языке C.	При работе с текстовыми файлами данные интерпретируются как последовательности символов. При работе с бинарными файлами данные	Задание открытого типа с развернутым ответом	2	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		читаются и записываются побайтово, без преобразований, точно как они представлены в памяти..			
39.	Дайте определение полиморфизму в объектно-ориентированном программировании.	Полиморфизм — это возможность объектов с одинаковым интерфейсом (именем метода) иметь разную реализацию.	Задание открытого типа с развернутым ответом	3	1
40.	Что такое "менеджер контекста" (context manager) в языке Python?	Это класс или функция, которые реализуют методы <code>__enter__()</code> и <code>__exit__()</code> и позволяют гарантированно выполнить начальные и завершающие операции.	Задание открытого типа с развернутым ответом	3	2
41.	Объясните, в каких ситуациях наследование является предпочтительным, а в каких лучше использовать композицию (включение объекта одного класса в качестве поля другого).	Наследование предпочтительно, когда между классами существует отношение «является» (is-a), и производный класс — это частный случай или специализация базового. Композиция предпочтительна, когда отношение лучше описывается как «имеет» (has-a) или «включает в себя».	Задание открытого типа с развернутым ответом	3	1
42.	Что такое шаблон (template) в C++?	Шаблон — это механизм C++, позволяющий писать обобщенный код, работающий с	Задание открытого типа с развернутым ответом	2	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		различными типами данных.			
43.	Объясните, какую проблему решает «виртуальный деструктор» в базовом классе C++.	Без виртуального деструктора при удалении объекта производного класса через указатель на базовый класс может возникнуть неполное разрушение объекта.	Задание открытого типа с развернутым ответом	3	2
44.	Вам необходимо сохранить в файл и затем загрузить массив записей о студентах (поля: char name[50]; int group; float gpa;). Напишите фрагмент кода на C для записи всего массива students размером n в бинарный файл "data.bin".	с FILE *f = fopen("data.bin", "wb"); if (f) { fwrite(students, sizeof(Student), n, f); fclose(f); }	Задание комбинированного типа: практико-ориентированное	3	2
45.	В программе часто требуется выполнять операции добавления и удаления элементов только с одного конца коллекции. Определите, какая структура данных (стек, очередь, список) здесь наиболее уместна, и объясните почему.	Стек (LIFO). Потому что стек поддерживает эффективные операции добавления (push) и удаления (pop) только с одного конца (вершины), что полностью соответствует условию задачи.	Задание комбинированного типа: практико-ориентированное	2	1
46.	В программе создается динамический массив целых чисел, размер которого определяется пользователем. Напишите на C++ код для безопасного создания такого массива с обработкой возможного исключения std::bad_alloc.	cpp int size; std::cin >> size; int *arr = nullptr; try { arr = new int[size]; } catch (const std::bad_alloc& e) { std::cerr << "Ошибка выделения памяти: " << e.what() << std::endl; } arr delete[] arr;	Задание комбинированного типа: практико-ориентированное	3	2
47.	При обходе дерева файловой системы названия файлов добавляются в коллекцию. Важно,	file_names = set() file_names.add("document.txt")	Задание комбинированного	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	чтобы имена не повторялись, и чтобы можно было быстро проверить наличие файла. Напишите фрагмент кода Python, который: создаёт коллекцию для хранения уникальных имён файлов; добавляет имя файла "document.txt" в эту коллекцию; проверяет, содержится ли в коллекции имя "README.md", и выводит сообщение о результате.	if "README.md" in file_names: print("Файл найден.") else: print("Файл не найден.")	о типа: практико-ориентированное		
48.	В программе моделируется работа простого калькулятора, поддерживающего операции сложения и вычитания. Опишите, как с помощью указателя на базовый класс можно вызвать операцию вычисления, не зная на этапе компиляции, сложение это или вычитание.	Создается указатель типа Operation*, которому присваивается адрес объекта Addition или Subtraction. Вызов op->execute(x, y) приведет к выполнению правильной операции благодаря механизму позднего связывания.	Задание комбинированного типа: практико-ориентированное	3	1
49.	При считывании данных из бинарного файла программа работает корректно на одной машине (x86, Windows), но "ломается" на другой (ARM, Linux), выдавая неверные значения полей структур. Назовите одну из наиболее вероятных причин такой платформенной зависимости при работе с бинарными файлами.	Разное выравнивание структур в памяти (alignment) 2. Разный порядок байтов (endianness)	Задание комбинированного типа: практико-ориентированное	3	2
50.	При интеграции отечественного СУБД-совместимого ПО с legacy-системой требуется часто считывать и обрабатывать поток транзакций в порядке их поступления. Какую абстрактную структуру данных нужно заложить в основу буфера для этого потока? А) Стек (LIFO). Б) Очередь (FIFO). В) Множество (Set). Г) Ассоциативный массив (Map).	Б. Ключевое требование — "в порядке их поступления". Это классическое определение структуры данных "Очередь", работающей по принципу FIFO (First-In-First-Out).	Задание комбинированного типа с выбором одного ответа и обоснованием	1	1
51.	Разрабатывается отечественная платформа для создания бизнес-приложений. Требуется, чтобы новые модули (расчет налогов,	Б. Полиморфизм, основанный на абстрактных интерфейсах,	Задание комбинированного типа с выбором	3	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>формирование отчетов) могли легко подключаться к ядру без его перекомпиляции.</p> <p>Какой архитектурный принцип ООП является ключевым для этого?</p> <p>А) Инкапсуляция данных внутри классов.</p> <p>Б) Полиморфизм и работа через абстрактные интерфейсы (API).</p> <p>В) Глубокое наследование.</p> <p>Г) Использование дружественных функций.</p>	<p>позволяет ядру системы вызывать методы, не зная конкретной реализации.</p> <p>Новый модуль реализует этот стандартный интерфейс и регистрируется в системе. Ядро работает с ним через указатель на интерфейс.</p>	<p>одного ответа и обоснованием</p>		
52.	<p>Для обеспечения долгосрочной сохранности архивных данных в отечественном цифровом хранилище выберите наиболее устойчивый к устареванию софта формат файла.</p> <p>А) Простой текстовый формат с четкой спецификацией (например, CSV, XML).</p> <p>Б) Бинарный формат, оптимизированный под конкретную версию программы.</p> <p>В) Документ проприетарного офисного пакета.</p> <p>Г) Исполняемый файл (.exe).</p>	<p>А. Простые, открытые текстовые форматы с публичной спецификацией не зависят от конкретного ПО, операционной системы или аппаратной архитектуры. Данные в таком формате можно прочитать помощью любого текстового редактора или парсера, написанного на любом языке.</p>	<p>Задание комбинированного типа с выбором одного ответа и обоснованием</p>	2	2
53.	<p>Для отечественной системы документооборота необходимо выбрать форматы экспорта данных, которые гарантируют долгосрочную читаемость и независимость от конкретного ПО. Какие форматы этому соответствуют?</p> <p>А) Формат ODF (OpenDocument Format) — открытый стандарт для офисных документов.</p> <p>Б) Простой текстовый формат (TXT) или Markdown.</p> <p>В) Формат PDF/A (специализация PDF для долгосрочного архивирования).</p> <p>Г) Документ в проприетарном формате последней версии популярного офисного пакета.</p>	<p>АБВ</p> <p>А) ODF: Является открытым международным стандартом. Его спецификация публична, что позволяет создавать совместимые приложения вне зависимости от вендоров.</p> <p>Б) Текстовый формат/TXT/Markdown: Простой, читаемый любым редактором.</p> <p>Наиболее</p>	<p>Задание комбинированного типа с выбором нескольких ответов и обоснованием</p>	2	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	Д) Бинарный файл внутреннего формата программы.	устойчив к устареванию, так как не содержит сложной разметки. В) PDF/A: Специально создан для долгосрочного архивного хранения. Г) и Д) Являются проприетарными и привязывают пользователя к конкретному ПО или версии, что противоречит принципу независимости и долгосрочной сохранности.			
54.	<p>При проектировании протокола обмена для отечественной распределенной системы IoT-датчиков необходимо учесть ограничения устройств (малая память, низкая скорость). Какие из подходов к управлению данными в памяти будут здесь эффективны?</p> <p>А) Использование статических массивов фиксированного размера вместо динамических выделений.</p> <p>Б) Отказ от сложных структур данных (деревья, графы) в пользу простых списков или массивов.</p> <p>В) Активное использование сборки мусора (Garbage Collection) для упрощения кода.</p> <p>Г) Структурное программирование: простые функции, минимизация глубины вызовов (стек).</p> <p>Д) Хранение всех данных в виде объектов со сложными иерархиями наследования.</p>	<p>АБГ</p> <p>А) Статические массивы: Избегают накладных расходов на выделение/освобождение динамической памяти и фрагментации кучи. Б) Простые структуры: Список и массивы потребляют меньше памяти на служебные данные (указатели) и проще в обработке. Г) Структурное программирование: Позволяет писать компактный, эффективный код с предсказуемым использованием стека.</p>	Задание комбинированного типа с выбором нескольких ответов и обоснованием	3	1
ОПК-6 Способен разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием					
55.	При составлении технического задания (ТЗ) на ПО для управления	2	Задание закрытого типа с	2	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	сетевым оборудованием, какой принцип структурного программирования является наиболее важным для описания алгоритмов работы системы? 1) Использование сложной многоуровневой рекурсии. 2) Четкая последовательность, ветвление и циклы для описания детерминированных процессов (например, диагностики сети). 3) Обязательное использование оператора goto для гибкости. 4) Полное отсутствие какой-либо структуры в описании логики.		однозначным выбором варианта ответа		
56.	В ТЗ на сетевое оборудование вы описываете классы устройств (маршрутизаторы, коммутаторы, точки доступа). Какой принцип ООП наилучшим образом отражает эту модель? 1) Создание иерархии классов (наследование), где общие свойства вынесены в базовый класс "Сетевое устройство". 2) Отказ от какой-либо классификации, описание каждого устройства с нуля. 3) Использование только глобальных переменных для хранения характеристик всех устройств. 4) Принцип "одна программа — одна длинная функция".	1	Задание закрытого типа с однозначным выбором варианта ответа	2	1
57.	При оценке стоимости владения в бизнес-плане необходимо обрабатывать потоки финансовых данных (ежемесячные платежи, амортизация). Какую концепцию ООП удобно использовать для моделирования таких потоков? 1) Создание класса "ФинансовыйПоток" с методами для расчета NPV, IRR и полями для хранения данных. 2) Использование множества независимых, не связанных друг с другом переменных. 3) Запись всех данных в одну глобальную строковую переменную. 4) Ручной расчет на бумаге без формализации модели.	1	Задание закрытого типа с однозначным выбором варианта ответа	3	1
58.	Для управления динамическим списком заявок на оборудование в системе учета, описанной в ТЗ,	1	Задание закрытого типа с однозначным	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>оптимальнее всего предусмотреть в проекте:</p> <p>1) Использование динамических информационных структур (например, списков), так как количество заявок заранее неизвестно и меняется.</p> <p>2) Статический массив под 1000000 заявок на все случаи жизни.</p> <p>3) Отказ от хранения истории заявок.</p> <p>4) Хранение каждой заявки в отдельном файле на диске без индексации.</p>		выбором варианта ответа		
59.	<p>Разрабатывая бизнес-план, вы создаете несколько сценариев (пессимистичный, оптимистичный). Какая концепция ООП позволяет эффективно работать с этими сценариями как с объектами?</p> <p>1) Полиморфизм: создание базового класса "Сценарий" с виртуальным методом calculate(), который по-разному реализован в производных классах.</p> <p>2) Копирование одного и того же кода расчета для каждого сценария с ручным изменением цифр.</p> <p>3) Использование только одного, "среднего" сценария.</p> <p>4) Описание сценариев в презентации PowerPoint без формальной модели.</p>	1	Задание закрытого типа с однозначным выбором варианта ответа	3	1
60.	<p>Для хранения и быстрого поиска нормативов и спецификаций оборудования в системе, описанной в ТЗ, оптимальной структурой данных будет:</p> <p>1) Использование ассоциативных контейнеров (словарей, map), где ключом является код или название норматива.</p> <p>2) Несортированный однонаправленный список, где поиск осуществляется перебором с начала.</p> <p>3) Стек (LIFO), куда новые нормативы кладутся сверху.</p> <p>4) Очередь (FIFO), куда нормативы добавляются в конец.</p>	1	Задание закрытого типа с однозначным выбором варианта ответа	2	3
61.	<p>В бизнес-плане оснащения лаборатории для моделирования различных сценариев загрузки вычислительных ресурсов могут быть использованы знания о структурах данных. Какие из</p>	2, 3, 5	Задание закрытого типа с многозначным выбором варианта ответа	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>перечисленных структур являются динамическими и подходят для хранения прогнозных данных с меняющимся объемом?</p> <p>1) Статический массив фиксированного размера (int workload[365]).</p> <p>2) Однонаправленный связный список.</p> <p>3) Стек (LIFO) на основе динамической памяти.</p> <p>4) Бинарный файл конфигурации.</p> <p>5) Двухнаправленный связный список.</p>				
62.	<p>При проектировании объектной модели в ТЗ для системы учета IT-активов (компьютеры, принтеры, серверы) можно применить принципы ООП. Какие из следующих концепций ООП будут здесь наиболее полезны?</p> <p>1) Инкапсуляция: объединение данных об активе (серийный номер, инвентарный номер) и методов работы с ним в единый класс.</p> <p>2) Наследование: создание иерархии классов (например, СетевоеОборудование -> Маршрутизатор).</p> <p>3) Полиморфизм: возможность единообразно обрабатывать разные типы активов в отчетах.</p> <p>4) Использование исключительно глобальных переменных для хранения всех данных.</p> <p>5) Отказ от какой-либо структуры в пользу сплошного текстового описания.</p>	1, 2, 3	Задание закрытого типа с многозначным выбором варианта ответа	3	1
63.	<p>При разработке финансовой модели в бизнес-плане для расчета ROI (возврата на инвестиции) эффективно использовать объектно-ориентированный подход. Какие из следующих сущностей логично представить в виде отдельных классов?</p> <p>1) Класс "Инвестиция" с полями: сумма, дата, тип оборудования.</p> <p>2) Класс "Экономия" с методами расчета снижения затрат.</p> <p>3) Единая таблица в Excel без выделения сущностей.</p> <p>4) Класс "СценарийРасчета" с виртуальным методом execute().</p> <p>5) Набор разрозненных глобальных переменных в скрипте.</p>	1, 2, 4	Задание закрытого типа с многозначным выбором варианта ответа	3	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
64.	<p>В ТЗ на систему резервного копирования критически важных конфигураций сетевого оборудования необходимо предусмотреть надежное хранение. Какие из перечисленных требований к структуре и формату данных для резервных копий являются обоснованными?</p> <p>1) Данные должны сохраняться в бинарном формате для максимальной скорости записи на диск.</p> <p>2) Резервные копии должны быть читаемы человеком (текстовый формат) для аварийного восстановления без специализированного ПО.</p> <p>3) Файлы резервных копий должны иметь четкую структуру и версионирование.</p> <p>4) Все копии должны храниться только в зашифрованном виде, недоступном для чтения даже администратором.</p> <p>5) Копии должны быть привязаны к конкретной версии железа и не подлежать переносу.</p>	2, 3	Задание закрытого типа с многозначным выбором варианта ответа	3	2
65.	<p>При анализе потоков данных в бизнес-плане (затраты, доходы) удобно использовать структуры, позволяющие эффективно добавлять и обрабатывать записи. Какие из перечисленных структур данных хорошо подходят для моделирования такого последовательного потока, где важен порядок по времени?</p> <p>1) Очередь (FIFO) для обработки событий в хронологическом порядке.</p> <p>2) Стек (LIFO), так как последние данные всегда важнее.</p> <p>3) Однонаправленный список для последовательного хранения и итерации.</p> <p>4) Ассоциативный массив (словарь), где ключом является случайное число.</p> <p>5) Множество (set), так как оно гарантирует уникальность сумм платежей.</p>	1, 3	Задание закрытого типа с многозначным выбором варианта ответа	2	3
66.	<p>При планировании этапов внедрения оборудования в бизнес-плане, сама последовательность этапов может быть описана как</p>	1, 2, 3	Задание закрытого типа с многозначным	2	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>алгоритм. Какие конструкции структурного программирования логично использовать при таком описании?</p> <p>1) Следование: Последовательное описание этапов (монтаж, настройка, тестирование).</p> <p>2) Ветвление: Описание условий перехода к следующему этапу ("если тестирование провалено, вернуться к настройке").</p> <p>3) Цикл: Указание на итеративный процесс (например, "повторять тестирование до достижения критериев").</p> <p>4) Рекурсия: Описание процесса как самовывзывающегося.</p> <p>5) GOTO: Использование перекрестных ссылок между разными разделами документа.</p>		выбором варианта ответа		
67.	<p>Установите соответствие между типом динамической структуры данных и ситуацией её применения в бизнес-плане или ТЗ.</p> <p>А) Однонаправленный список</p> <p>Б) Стек (LIFO)</p> <p>В) Очередь (FIFO)</p> <p>1) Моделирование списка заявок на обслуживание оборудования, которые обрабатываются в порядке поступления.</p> <p>2) Моделирование цепочки зависимых задач (этапов проекта), где каждая задача ссылается на следующую.</p> <p>3) Моделирование истории выполненных команд в системе для возможности отмены (операция "Undo").</p>	A2, Б3, В1	Задание закрытого типа на установление соответствия	2	3
68.	<p>Установите соответствие между принципом ООП и его отражением в структуре ТЗ на комплексное ПО.</p> <p>А) Инкапсуляция</p> <p>Б) Наследование</p> <p>В) Полиморфизм</p> <p>1) В ТЗ указано, что модули "ОтчетБазовый" и "ОтчетРасширенный" должны иметь общую часть в описании формата вывода.</p> <p>2) В ТЗ указано, что внутренняя логика расчета в модуле "Анализатор" скрыта, а наружу предоставлен только интерфейс с методами загрузитьДанные() и получ</p>	A2, Б1, В3	Задание закрытого типа на установление соответствия	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	итьРезультат(). 3) В ТЗ указано, что модули "ОповещениеПоEmail" и "ОповещениеВTelegram" должны вызываться единообразно через метод отправить() общего интерфейса.				
69.	Установите соответствие между видом класса в ООП и его аналогом при описании сущностей в ТЗ. А) Конкретный класс Б) Абстрактный класс В) Интерфейс (абстрактный класс только с чистыми виртуальными методами) 1) Детальное техническое описание конкретного устройства (например, "Коммутатор модели X: порты, скорость"). 2) Общее описание обязательных функций, которые должен реализовать любой модуль определённого типа (например, "все модули отчетов должны генерировать PDF"). 3) Общее описание категории оборудования с указанием общих характеристик, но без конкретной модели (например, "Беспроводная точка доступа стандарта 802.11ac").	A1, B3, B2	Задание закрытого типа на установление соответствия	3	3
70.	Установите соответствие между типом контейнера STL и задачей в рамках разработки бизнес-плана или модели. А) std::vector Б) std::map (или std::unordered_map) В) std::set 1) Хранение списка ежемесячных платежей за обслуживание, где важен порядок и возможен быстрый доступ по индексу. 2) Хранение справочника норм амортизации, где доступ к норме осуществляется по коду типа оборудования. 3) Хранение уникальных идентификаторов установленного ПО для исключения дублирования в смете.	A1, B2, B3	Задание закрытого типа на установление соответствия	2	3
71.	Установите соответствие между этапом работы с файлом и его описанием в разделе ТЗ "Требования к данным". А) Открытие файла Б) Чтение/запись данных	A3, B1, B2	Задание закрытого типа на установление соответствия	1	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>В) Закрытие файла</p> <p>1) Процесс извлечения или сохранения структурированной информации согласно заданному формату.</p> <p>2) Освобождение ресурсов и гарантированная запись всех данных на физический носитель.</p> <p>3) Инициализация доступа к файлу с указанием режима (чтение, запись) и проверка прав.</p>				
72.	<p>Установите соответствие между операцией над связным списком и бизнес-процессом в рамках оснащения, который она может моделировать.</p> <p>А) Добавление элемента в начало</p> <p>Б) Удаление элемента по значению</p> <p>В) Поиск элемента</p> <p>1) Внеплановая поставка срочного оборудования, которое должно быть учтено в первую очередь.</p> <p>2) Проверка, есть ли конкретная модель ноутбука в текущей заявке на закупку.</p> <p>3) Списание устаревшего оборудования из реестра активов.</p>	А1, Б3, В2	Задание закрытого типа на установление соответствия	2	3
73.	<p>Установите последовательность действий при корректной работе с динамической структурой данных (например, списком закупаемого оборудования) в программе-калькуляторе сметы:</p> <p>А) Освобождение памяти, занятой списком.</p> <p>Б) Добавление нового элемента (позиции оборудования) в список.</p> <p>В) Выделение памяти для нового элемента списка и инициализация его полей.</p> <p>Г) Итерация по списку для расчета общей стоимости.</p>	БВГА	Задание закрытого типа на установление последовательности	2	3
74.	<p>Установите последовательность этапов обработки запроса в модуле учета, спроектированном с использованием принципа инкапсуляции:</p> <p>А) Возврат результата (или ошибки) вызывающему коду.</p> <p>Б) Внутренняя валидация и обработка данных внутри объекта.</p> <p>В) Создание объекта-модели запроса с инкапсулированными данными.</p>	ГВБА	Задание закрытого типа на установление последовательности	2	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	Г) Получение входных данных (например, ID оборудования).				
75.	<p>Установите последовательность операций при безопасном чтении конфигурации из бинарного файла в программе настройки:</p> <p>А) Закрытие файла.</p> <p>Б) Проверка контрольной суммы (checksum) считанных данных.</p> <p>В) Открытие файла в режиме "rb" (read binary).</p> <p>Г) Чтение блока данных в структуру в памяти.</p> <p>Д) Проверка, что файл существует и доступен для чтения.</p>	ДВГБА	Задание закрытого типа на установление последовательности	3	2
76.	<p>Установите последовательность этапов разработки бизнес-плана, где используется итеративная (циклическая) модель уточнения:</p> <p>А) Сбор требований и анализ рынка оборудования.</p> <p>Б) Запуск пилотного проекта и сбор обратной связи.</p> <p>В) Корректировка финансовой модели и технических требований.</p> <p>Г) Разработка первоначальной финансовой модели и ТЗ.</p> <p>Д) Формирование итогового пакета документов.</p>	АГБВД	Задание закрытого типа на установление последовательности	2	4
77.	<p>Установите последовательность шагов при добавлении нового типа оборудования в существующую объектно-ориентированную модель каталога в ТЗ:</p> <p>А) Определение, является ли новый тип специализацией существующего класса (наследование) или отдельной сущностью.</p> <p>Б) Документирование новых или измененных интерфейсов (API) для интеграции.</p> <p>В) Описание атрибутов (полей) и поведения (методов) нового класса.</p> <p>Г) Анализ общих свойств с уже описанным в ТЗ оборудованием.</p>	ГАВБ	Задание закрытого типа на установление последовательности	3	2
78.	<p>Установите последовательность формирования итогового ТЗ на основе модульного (структурного) подхода:</p> <p>А) Детализация технических требований для каждого модуля/подсистемы.</p> <p>Б) Формирование общего описания системы (назначение, цели).</p> <p>В) Согласование и утверждение</p>	БГАДВ	Задание закрытого типа на установление последовательности	2	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	итогового документа. Г) Декомпозиция системы на ключевые модули (учет, мониторинг, отчетность). Д) Описание интерфейсов взаимодействия между модулями.				
79.	Для моделирования потока входящих заказов в систему обслуживания, где важно соблюдать порядок "первым пришел — первым обслужен", в python следует использовать структуру данных _____.	очередь (queue)	Задание открытого типа на дополнение	1	3
80.	Принцип ООП, позволяющий в ТЗ задать общие требования к группе устройств (например, "все устройства должны иметь метод getSerialNumber()"), а в конкретных спецификациях уточнять детали, называется _____.	наследование	Задание открытого типа на дополнение	1	1
81.	В объектно-ориентированной модели ТЗ система оповещения может отправлять сообщения по email, SMS или в мессенджер. Способность вызывать метод send() для любого из этих каналов, не зная конкретной реализации, обеспечивается принципом _____.	полиморфизм	Задание открытого типа на дополнение	2	1
82.	При расчете сметы в бизнес-плане, когда точное количество позиций оборудования заранее неизвестно и может меняться, для хранения списка позиций в программе-расчетнике целесообразно использовать _____ массив или связный список.	динамический	Задание открытого типа на дополнение	2	4
83.	Класс в ООП, который объединяет данные (поля) об устройстве и методы работы с ними (например, calculateDepreciation()), скрывая внутреннюю логику расчетов, демонстрирует принцип _____.	инкапсуляции	Задание открытого типа на дополнение	2	4
84.	Для гарантированного освобождения ресурсов (например, закрытия файла, сетевого соединения или блокировки) при любом развитии событий, в том числе при возникновении исключений, в Python используется паттерн _____.	менеджеры контекста (context managers, конструкция with)	Задание открытого типа на дополнение	3	2
85.	В ТЗ на систему мониторинга, где события нужно обрабатывать в обратном порядке (последнее пришедшее — первым на анализ),	стек (или LIFO)	Задание открытого типа на дополнение	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	для их временного хранения подходит структура данных _____.				
86.	Объясните, как применение принципа нисходящего проектирования (top-down design) из структурного программирования может улучшить процесс разработки технического задания (ТЗ) на комплексную систему управления IT-активами.	<p>Принцип нисходящего проектирования предлагает начинать с общей, самой верхнеуровневой задачи и последовательно декомпозировать её на подзадачи. При разработке ТЗ это означает:</p> <ol style="list-style-type: none"> 1. Определение главной цели системы 2. Разбиение этой цели на основные подсистемы/модули 3. Детализация требований к каждому модулю 	Задание открытого типа с развернутым ответом	3	1
87.	Что такое "умный указатель" (std::unique_ptr) в контексте C++?	std::unique_ptr — это "умный указатель", который обеспечивает исключительное владение динамически выделенным объектом и гарантирует его автоматическое уничтожение (вызов delete) при выходе unique_ptr из области видимости.	Задание открытого типа с развернутым ответом	3	2
88.	Объясните разницу между отношениями "является" (is-a) и "имеет" (has-a) в контексте ООП.	Отношение "является" (is-a) описывает наследование и выражает специализацию. Это значит, что производный класс — это частный случай базового.	Задание открытого типа с развернутым ответом	3	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		Отношение "имеет" (has-a) описывает композицию или агрегацию. Это значит, что один класс включает в себя другой класс как часть.			
89.	Задача: В ТЗ на систему учета необходимо описать формат хранения списка сотрудников (ФИО, табельный номер). Требуется простота чтения человеком. 1) Предложите тип файла (текстовый/бинарный) и простой формат для этой цели. 2) Приведите пример двух строк в этом формате.	Текстовый файл, формат CSV (значения, разделенные запятыми). Пример строк: Иванов Иван Иванович, 12547 Петрова Анна Сергеевна, 12548	Задание комбинированного типа: практико-ориентированное	1	4
90.	При проектировании структуры программы для бизнес-плана требуется обработать неизвестное заранее количество статей расходов. Назовите структуру данных, которая лучше всего подходит для хранения такого динамического списка в памяти. Кратко объясните одно преимущество ее использования по сравнению со статическим массивом.	Динамический массив (std::vector в C++). Позволяет гибко добавлять/удалять элементы, не теряя памяти (в отличие от статического массива "с запасом").	Задание комбинированного типа: практико-ориентированное	1	1
91.	В программе-калькуляторе сметы часто требуется отменять последнее введенное значение. Назовите структуру данных, идеально подходящую для реализации функции "Отмена" (Undo). Объясните принцип ее работы.	Стек (LIFO). "Последним пришел — первым ушел". Операция отмены удаляет последнее добавленное значение.	Задание комбинированного типа: практико-ориентированное	1	1
92.	В системе учета требуется хранить пары "код оборудования — его описание" для быстрого поиска описания по коду. Назовите контейнер STL, оптимальный для этой задачи. Напишите строку кода на C++ для объявления такого контейнера (типы выберите условно).	std::map или std::unordered_map. std::map<std::string, std::string> equipmentCatalog;	Задание комбинированного типа: практико-ориентированное	2	3
93.	При моделировании работы call-центра в ТЗ описаны входящие звонки, которые должны обрабатываться в порядке поступления.	Очередь (FIFO). enqueue (добавить в конец) и dequeue (взять из начала).	Задание комбинированного типа: практико-ориентированное	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	Назовите структуру данных, моделирующую эту очередь звонков. Назовите две основные операции с этой структурой.				
94.	В программе для бизнес-плана расчет для разных сценариев ("Базовый", "Оптимистичный") выполняется по разным формулам, но вызывается из одного места. Назовите принцип/механизм ООП, позволяющий это реализовать. Какой ключевой модификатор используется в базовом классе для соответствующего метода?	Полиморфизм (через виртуальные функции). virtual	Задание комбинированного типа: практико-ориентированное	2	1
95.	Для хранения истории изменений цен в калькуляторе бизнес-плана (чтобы можно было откатиться) оптимальна структура данных, работающая по принципу... Варианты: А) LIFO (Last In, First Out) — "последним пришел, первым ушел". Б) FIFO (First In, First Out) — "первым пришел, первым ушел". В) Произвольный доступ по ключу. Г) Хранение только последнего значения.	А. Функция отмены действия (undo) требует отменять последние изменения, что соответствует принципу LIFO, реализуемому структурой "стек".	Задание комбинированного типа с выбором и обоснованием	2	1
96.	В ТЗ указано, что модуль "Аутентификатор" должен предоставлять строго определенный метод проверитьПароль(логин, пароль) и скрывать внутреннюю базу данных и алгоритмы хеширования. Какой принцип ООП здесь задействован? А) Наследование. Б) Полиморфизм. В) Инкапсуляция. Г) Абстракция.	В. Инкапсуляция — это объединение данных (база паролей, хеши) и методов работы с ними в одной единице (классе) и сокрытие внутренней реализации (как именно хранятся и сравниваются пароли) от внешнего кода.	Задание комбинированного типа с выбором и обоснованием	2	1
97.	В ТЗ на систему контроля версий конфигураций описано, что можно добавлять, но не удалять записи об изменениях. Какую структуру данных это напоминает, и какую операцию она не поддерживает в чистом виде? А) стек (добавление — push, отсутствует операция удаления из середины). Б) Очередь (поддерживает и добавление, и удаление). В) Множество (поддерживает	А. Система, где можно только добавлять записи и просматривать/откатываться к более ранним, но не удалять произвольные записи из истории, близка к стеку. В	Задание комбинированного типа с выбором и обоснованием	2	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	удаление). Г) Массив (поддерживает удаление).	классическом стеке нет операции удаления произвольного элемента.			
98.	<p>Для временного хранения команд, отправляемых на сетевое оборудование в процессе настройки (с возможностью повтора последней команды), подходят структуры данных, основанные на следующих принципах организации:</p> <p>А) FIFO (First In, First Out). Б) LIFO (Last In, First Out). В) Произвольный доступ по индексу. Г) Доступ только к последнему добавленному элементу. Д) Гарантия уникальности всех элементов.</p>	<p>БГ</p> <p>Б) LIFO: Принцип "последняя команда — первая для повтора" соответствует структуре данных "стек", которая работает по принципу LIFO.</p> <p>Г) Доступ только к последнему добавленному элементу: Это свойство стека. Для повтора команды нужен доступ именно к последней отправленной команде, что совпадает с данным свойством.</p>	Задание комбинированного типа с выбором нескольких ответов и обоснованием	2	2
99.	<p>При разработке ТЗ на ПО для визуализации иерархии подразделений компании (дерева) программист выбирает подход к хранению структуры в памяти. Какие из следующих способов организации данных теоретически пригодны для представления древовидной структуры?</p> <p>А) Массив с вычислением индексов родителя/потомков. Б) Структура узла с указателями на дочерние узлы (например, список детей). В) Односвязный список (каждый узел указывает только на один следующий). Г) Очередь (FIFO). Д) Стек (LIFO).</p>	<p>АБ</p> <p>А) Массив: Дерево может быть представлено в массиве, где для узла с индексом i потомки имеют индексы $2*i+1$ и $2*i+2$. Б) Узлы с указателями: узел содержит данные и список/массив указателей на свои дочерние узлы. Это наиболее гибкий и распространенный способ.</p>	Задание комбинированного типа с выбором нескольких ответов и обоснованием	3	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
100.	<p>В ТЗ указано, что система должна вести подробный журнал аудита (лог) всех изменений в конфигурациях. Какие из предложенных форматов/подходов к ведению лога являются практичными и рекомендуемыми?</p> <p>А) Запись в оперативную память без сохранения на диск.</p> <p>Б) Запись в структурированном текстовом формате (например, каждая строка в JSON).</p> <p>В) Запись в простой текстовый файл с разделителями (CSV).</p> <p>Г) Использование специализированной системы/библиотеки логирования (например, spdlog для C++).</p> <p>Д) Запись в бинарный файл, являющийся точной копией внутренних структур программы.</p>	<p>БВГ Б)</p> <p>Структурированный текст (JSON): Позволяет легко добавлять новые поля, удобен для последующего машинного парсинга и анализа.</p> <p>В) Текстовый CSV: Прост для импорта в табличные процессоры или базы данных для составления отчетов.</p> <p>Г)</p> <p>Специализированные библиотеки: Обеспечивают ротацию логов, разные уровни логирования, асинхронную запись, что важно для надежной работы в production-среде.</p>	Задание комбинированного типа с выбором нескольких ответов и обоснованием	2	2
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения					
101.	<p>При разработке алгоритма для чтения конфигурационного файла, где порядок строк не важен, а важна скорость поиска значения по ключу, какую структуру данных практичнее всего использовать в программе?</p> <p>1) Односвязный список.</p> <p>2) Ассоциативный массив (словарь, std::map или std::unordered_map).</p> <p>3) Стек (LIFO).</p> <p>4) Очередь (FIFO).</p>	2	Задание закрытого типа с однозначным выбором варианта ответа	1	3
102.	<p>Вы разрабатываете функцию для безопасного копирования строки с ограничением по максимальной длине (чтобы избежать переполнения буфера). Какой принцип структурного программирования является ключевым для реализации такой проверки?</p>	2	Задание закрытого типа с однозначным выбором варианта ответа	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	1) Использование рекурсии. 2) Ветвление (проверка условия: если длина источника > размера буфера, то обрезать или выдать ошибку). 3) Исключительно последовательное выполнение без проверок. 4) Использование глобальных переменных для передачи параметров.				
103.	Для реализации системы плагинов в программе, где каждый плагин должен предоставлять функцию <code>init()</code> и <code>execute()</code> , оптимально использовать механизм ООП, основанный на: 1) Множественном наследовании от нескольких совершенно разных классов. 2) Абстрактном базовом классе (интерфейсе) с чистыми виртуальными методами <code>init()</code> и <code>execute()</code> . 3) Дружественных функциях, имеющих доступ к приватным данным ядра. 4) Единой глобальной структуре для всех данных плагинов.	2	Задание закрытого типа с однозначным выбором варианта ответа	2	1
104.	При проектировании алгоритма обхода файловой системы и сбора списка всех файлов с расширением <code>.txt</code> наиболее подходящей управляющей конструкцией является: 1) Линейная последовательность инструкций. 2) Рекурсия или цикл для обработки вложенных каталогов. 3) Безусловный переход (<code>GOTO</code>) между участками кода. 4) Множество независимых условных операторов.	2	Задание закрытого типа с однозначным выбором варианта ответа	1	2
105.	В программе требуется хранить коллекцию уникальных идентификаторов сессий пользователей, к которым часто выполняется запрос "принадлежит ли этот ID коллекции?". Какой контейнер STL наиболее эффективен для этой практической задачи? 1) <code>std::vector</code> 2) <code>std::list</code> 3) <code>std::set</code> или <code>std::unordered_set</code> 4) <code>std::deque</code>	3	Задание закрытого типа с однозначным выбором варианта ответа	2	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
106.	<p>При разработке сетевого сервиса важно гарантировать освобождение сокета (сетевого соединения) при любом исходе (ошибка, корректное завершение). Какой идиоме C++ это соответствует?</p> <p>1) RAII (Resource Acquisition Is Initialization): выделение ресурса в конструкторе, освобождение — в деструкторе.</p> <p>2) Явный вызов close(socket) в конце каждой функции.</p> <p>3) Использование сборщика мусора.</p> <p>4) Игнорирование проблемы, так как ОС все почистит.</p>	1	Задание закрытого типа с однозначным выбором варианта ответа	2	2
107.	<p>Вы разрабатываете алгоритм валидации пароля. Какие из перечисленных проверок являются практичными и необходимыми для повышения безопасности?</p> <p>1) Проверка минимальной длины (например, ≥ 8 символов).</p> <p>2) Наличие хотя бы одной цифры.</p> <p>3) Наличие хотя бы одного спецсимвола (например, !, @, #).</p> <p>4) Проверка, что пароль является палиндромом.</p> <p>5) Проверка, что пароль не совпадает с логином.</p>	1, 2, 3, 5	Задание закрытого типа с многозначным выбором варианта ответа	2	3
108.	<p>При реализации поиска пути в лабиринте (например, алгоритм Lee или поиск в ширину BFS) для представления карты и организации вычислений можно использовать различные структуры данных. Какие из перечисленных подходят для этой задачи в Python?</p> <p>1) Двумерный список (list of lists), если размер известен и невелик.</p> <p>2) Массив NumPy (numpy.array) для эффективной работы с большими данными.</p> <p>3) Множество (set) для хранения всех ячеек лабиринта без учёта их координат.</p> <p>4) Очередь (collections.deque) для реализации волнового алгоритма (очередь фронта волны).</p> <p>5) Стек (list) в качестве основной структуры для хранения всей карты лабиринта.</p>	1, 2, 4	Задание закрытого типа с многозначным выбором варианта ответа	3	3
109.	Для создания расширяемой системы рендеринга графических примитивов (Круг, Квадрат,	1, 3, 4	Задание закрытого типа с многозначным	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>Треугольник) целесообразно использовать принципы ООП. Какие из следующих шагов являются правильными?</p> <p>1) Создать абстрактный базовый класс Shape с чистой виртуальной функцией draw().</p> <p>2) Создать один класс Graphic с огромным методом switch по типу фигуры.</p> <p>3) Отнаследовать от Shape конкретные классы Circle, Square, Triangle и реализовать в них draw().</p> <p>4) Хранить фигуры в контейнере указателей на Shape и отрисовывать в цикле вызовом shape->draw().</p> <p>5) Для каждой новой фигуры переписывать основной цикл отрисовки.</p>		выбором варианта ответа		
110.	<p>При чтении данных из бинарного файла, созданного на другой платформе, могут возникнуть проблемы. Какие из перечисленных факторов следует учесть при разработке кроссплатформенного кода чтения?</p> <p>1) Размер фундаментальных типов (например, int может быть 2, 4 или 8 байт).</p> <p>2) Порядок байтов (endianness: little-endian vs big-endian).</p> <p>3) Имя файла и его расположение в файловой системе.</p> <p>4) Выравнивание (padding) полей в структурах.</p> <p>5) Цветовую схему интерфейса программы.</p>	1, 2, 4	Задание закрытого типа с многозначным выбором варианта ответа	3	3
111.	<p>Вы пишете программу-агрегатор новостей, которой необходимо хранить ленту статей. Какие из требований к структуре данных для хранения ленты являются разумными?</p> <p>1) Быстрое добавление новой статьи в начало или конец ленты.</p> <p>2) Возможность быстрого удаления произвольной статьи из середины.</p> <p>3) Эффективный последовательный обход (для вывода пользователю).</p> <p>4) Обязательная сортировка статей по ID при каждом добавлении.</p> <p>5) Гарантия, что все заголовки статей уникальны.</p>	1, 3	Задание закрытого типа с многозначным выбором варианта ответа	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
112.	<p>При проектировании класса "Умный буфер" для работы с бинарными данными, который должен самостоятельно управлять памятью, необходимо реализовать определенные методы. Какие из перечисленных обязательны для обеспечения корректной работы и безопасности?</p> <p>1) Конструктор (выделяющий память или принимающий готовые данные). 2) Деструктор (освобождающий память). 3) Конструктор копирования и/или оператор присваивания (для правильного глубокого копирования). 4) Виртуальный метод для сериализации в XML. 5) Метод, возвращающий адрес участка памяти на Луне.</p>	1, 2, 3	Задание закрытого типа с многозначным выбором варианта ответа	3	3
113.	<p>Разрабатывая алгоритм сжатия повторяющихся последовательностей в строке (например, RLE), вы определяете основные этапы. Какие из следующих шагов являются частью такого алгоритма?</p> <p>1) Последовательный проход по символам входной строки. 2) Подсчет серий одинаковых подряд идущих символов. 3) Случайная перестановка символов в строке. 4) Формирование выходной строки в формате "символ-количество". 5) Шифрование результата с закрытым ключом.</p>	1, 2, 4	Задание закрытого типа с многозначным выбором варианта ответа	2	3
114.	<p>Установите соответствие между задачей и рекомендуемой структурой данных для её эффективной реализации в программе.</p> <p>Список 1:</p> <p>А) Кэширование 5 последних выполненных команд для отмены (Undo). Б) Обработка запросов на печать в порядке их поступления. В) Хранение тегов статей блога, где каждый тег должен быть уникальным. Г) Представление матрицы фиксированного размера для математических расчетов.</p> <p>Список 2:</p> <p>1) Двумерный статический массив.</p>	А3, Б2, В4, Г1	Задание закрытого типа на установление соответствия	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	2) Очередь (FIFO). 3) Стек (LIFO). 4) Множество (std::set).				
115.	<p>Установите соответствие между принципом ООП и его практическим применением при разработке программы интернет-магазина.</p> <p>Список 1:</p> <p>А) Инкапсуляция Б) Наследование В) Полиморфизм</p> <p>Список 2:</p> <p>1) Создание класса Electronics, наследующего от Product, для добавления специфичного поля warrantyMonths. 2) Расчет итоговой стоимости заказа через метод calculateTotal(), который внутренне суммирует цены позиций, скрывая детали от внешнего кода. 3) Использование единого метода processPayment() для обработки платежей с разных платформ (карта, электронный кошелек) через общий интерфейс PaymentGateway.</p>	A2, Б1, В3	Задание закрытого типа на установление соответствия	2	3
116.	<p>Установите соответствие между этапом работы с файлом и его корректной реализацией в языке С.</p> <p>Список 1:</p> <p>А) Открытие файла для записи новых данных (старые удаляются). Б) Открытие файла для добавления данных в конец. В) Открытие файла для чтения. Г) Корректное завершение работы с файлом.</p> <p>Список 2:</p> <p>1) fopen("data.txt", "r") 2) fopen("data.txt", "w") 3) fopen("data.txt", "a") 4) fclose(filePtr)</p>	A2, Б3, В1, Г4	Задание закрытого типа на установление соответствия	1	3
117.	<p>Установите соответствие между видом указателя в С++ и сценарием его практического применения.</p> <p>Список 1:</p> <p>А) "Сырой" указатель (T*) Б) Умный указатель std::unique_ptr В) Умный указатель std::shared_ptr</p> <p>Список 2:</p> <p>1) Временная передача владения объектом в функцию, которая должна стать его единственным владельцем. 2) Работа с legacy-кодом или API C-</p>	A2, Б1, В3	Задание закрытого типа на установление соответствия	3	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	библиотек, которые требуют простой указатель. 3) Реализация кэша объектов, к которым могут обращаться несколько независимых модулей программы.				
118.	<p>Установите соответствие между алгоритмической задачей и базовой управляющей конструкцией, которая в ней доминирует.</p> <p>Список 1:</p> <p>А) Проверка введенного пользователем пароля на соответствие критериям.</p> <p>Б) Вывод на экран всех элементов массива.</p> <p>В) Поиск максимального элемента в списке.</p> <p>Г) Последовательное выполнение инструкций по сборке устройства.</p> <p>Список 2:</p> <p>1) Следование (sequence).</p> <p>2) Ветвление (selection).</p> <p>3) Цикл (iteration).</p>	A2, Б3, В3, Г1	Задание закрытого типа на установление соответствия	1	3
119.	<p>Установите соответствие между встроенной структурой данных Python и её ключевой характеристикой, важной для выбора на практике.</p> <p>А) list (список)</p> <p>Б) set (множество)</p> <p>В) dict (словарь)</p> <p>Г) collections.deque (двусторонняя очередь)</p> <p>1) Быстрые операции добавления/удаления с обоих концов, но медленный произвольный доступ к элементам в середине.</p> <p>2) Быстрый произвольный доступ по индексу, но медленные вставка/удаление в начале списка.</p> <p>3) Хранение уникальных элементов с очень быстрой проверкой принадлежности (операция in), но без сохранения порядка (в версиях Python < 3.7) и без индексов.</p> <p>4) Хранение пар "ключ-значение" с очень быстрым доступом по ключу, в современных версиях Python (3.7+) сохраняет порядок добавления.</p>	A2, Б3, В4, Г1	Задание закрытого типа на установление соответствия	2	3
120.	Установите логическую последовательность шагов разработки функции, которая считывает целые числа из файла, пока не встретит ноль, и возвращает	БАГВБД	Задание закрытого типа на установление последовательности	1	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	их сумму: А) Открыть файл для чтения. Б) Объявить переменные для числа и суммы. В) В цикле считывать числа, добавляя их к сумме. Г) Проверить успешность открытия файла. Д) Закрыть файл и вернуть сумму. Е) Задать условие выхода из цикла (считанное число == 0).				
121.	Установите последовательность действий при корректном создании и использовании однонаправленного списка для хранения логга событий: А) Создать новый узел, записать в него данные. Б) "Прицепить" новый узел к концу списка. В) Объявить структуру Node с полями data и next. Г) Инициализировать "голову" списка значением nullptr. Д) Пройти по списку от головы для вывода всех событий.	ВГАБД	Задание закрытого типа на установление последовательности	2	3
122.	Установите последовательность этапов при реализации полиморфного поведения в программе рисования фигур: А) В цикле вызвать shape->draw() для каждого указателя в векторе. Б) Создать вектор указателей на базовый класс Shape. В) Реализовать конкретные классы Circle, Rectangle, наследующие Shape. Г) Определить абстрактный базовый класс Shape с чистой виртуальной функцией draw(). Д) Положить в вектор адреса объектов Circle и Rectangle.	ГВБДА	Задание закрытого типа на установление последовательности	3	3
123.	Установите последовательность шагов безопасной работы с динамическим массивом в C++: А) Использование массива (запись, чтение). Б) Освобождение памяти оператором delete[]. В) Проверка указателя на nullptr после выделения. Г) Выделение памяти оператором new type[size].	ГВАБ	Задание закрытого типа на установление последовательности	2	3
124.	Установите последовательность обработки исключения при работе с	ВДАГБ	Задание закрытого типа	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>файлом в C++:</p> <p>А) Выполнить операции чтения/записи в try-блоке.</p> <p>Б) Закрыть файл в блоке finally или деструкторе (RAII).</p> <p>В) Объявить объект потока (std::ifstream).</p> <p>Г) В блоке catch обработать возможное исключение std::ios_base::failure.</p> <p>Д) Открыть файл с помощью метода .open().</p>		на установление последовательности		
125.	<p>Установите последовательность вызова специальных функций класса при копировании объекта Matrix через присваивание:</p> <p>А) Деструктор старого объекта (левого операнда).</p> <p>Б) Конструктор копирования временного объекта.</p> <p>В) Оператор присваивания копированием (operator=).</p> <p>Г) Создание временного объекта как копии правого операнда.</p> <p>(Сценарий: Matrix a; Matrix b; a = b;)</p>	ВГБА	Задание закрытого типа на установление последовательности	3	3
126.	<p>Установите последовательность этапов алгоритма бинарного поиска в отсортированном массиве:</p> <p>А) Сравнить искомый элемент с элементом в середине текущего диапазона.</p> <p>Б) Если элемент не найден, вернуть специальное значение (например, -1).</p> <p>В) Определить начальный диапазон поиска (весь массив).</p> <p>Г) Если элемент в середине равен искомому, вернуть его индекс.</p> <p>Д) В зависимости от сравнения, сузить диапазон поиска до левой или правой половины.</p> <p>Е) Повторять шаги А, Г, Д, пока диапазон не станет пустым.</p>	ВАГДЕБ	Задание закрытого типа на установление последовательности	2	3
127.	<p>Для реализации алгоритма, который обрабатывает задачи в порядке их поступления (например, очередь на печать), следует использовать структуру данных _____.</p>	очередь (или FIFO)	Задание открытого типа на дополнение	1	3
128.	<p>В объектно-ориентированном программировании механизм, позволяющий классу-наследнику использовать или переопределять методы и поля базового класса, называется _____.</p>	наследование	Задание открытого типа на дополнение	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
129.	Чтобы гарантировать, что выделенная динамически память будет освобождена даже при возникновении исключения в C++, используется идиома _____.	RAII (или умные указатели)	Задание открытого типа на дополнение	2	2
130.	Функция для чтения данных из текстового файла в C, которая прекращает чтение при достижении конца файла или ошибки, называется _____.	fscanf (или fgetc)	Задание открытого типа на дополнение	1	3
131.	Контейнер стандартной библиотеки C++, который хранит элементы в отсортированном порядке и обеспечивает быстрый поиск за $O(\log n)$, называется _____.	set (или map, std::set)	Задание открытого типа на дополнение	2	3
132.	Принцип структурного программирования, который разбивает сложную задачу на более мелкие подзадачи (функции), называется _____.	декомпозиция (или модульность)	Задание открытого типа на дополнение	1	1
133.	Объясните, почему для хранения большого количества элементов, к которым требуется частый доступ по уникальному ключу (например, ID пользователя), на практике предпочтительнее использовать хэш-таблицу (std::unordered_map), а не линейный список (std::list).	Хэш-таблица обеспечивает среднее время доступа к элементу $O(1)$ (константное), так как вычисляет индекс по ключу. Линейный список требует последовательного перебора ($O(n)$), что при больших объемах данных приводит к неприемлемым задержкам. Практически это означает мгновенный поиск пользователя по ID независимо от размера базы.	Задание открытого типа с развернутым ответом	2	3
134.	Что такое деструктор в Python и в чём состоит ключевая особенность его вызова при наследовании?	В Python деструктор — это метод <code>__del__(self)</code> . При наследовании деструкторы не вызываются автоматически для цепочки родительских	Задание открытого типа с развернутым ответом	2	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		классов — если дочерний класс переопределяет <code>__del__</code> , он должен явно вызвать <code>super().__del__()</code> , чтобы обеспечить корректное освобождение ресурсов базового класса.			
135.	Опишите краткий алгоритм (шаги) проверки строки на палиндром (читается одинаково слева направо и справа налево) с использованием цикла.	1) Инициализировать два индекса: $i = 0$ (начало), $j = \text{len} - 1$ (конец). 2) В цикле, пока $i < j$, сравнивать символы <code>str[i]</code> и <code>str[j]</code> . 3) Если символы не равны — вернуть <code>false</code> (не палиндром). 4) Увеличить i , уменьшить j . 5) Если цикл завершился — вернуть <code>true</code> (палиндром).	Задание открытого типа с развернутым ответом	2	3
136.	Какая основная практическая проблема возникает при использовании бинарных файлов для хранения структур данных, если программа будет работать на компьютерах с разной архитектурой (например, x86 и ARM)?	Основная проблема — разный порядок байтов (endianness) и возможное разное выравнивание (padding) полей структур. Это приводит к некорректному чтению числовых значений и смещению полей, что вызывает ошибки. Решение: использовать платформо-независимую сериализацию (например, текстовый	Задание открытого типа с развернутым ответом	2	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		формат или преобразование в сетевой порядок байтов).			
137.	Объясните, в чем практическая выгода от использования принципа инкапсуляции при разработке класса «Банковский счет».	Инкапсуляция позволяет скрыть внутреннее состояние счета (баланс, история операций) и предоставить доступ только через строго определенные методы (например, deposit(), withdraw(), getBalance()). Это предотвращает прямое и неконтролируемое изменение баланса, обеспечивает валидацию операций (нельзя снять больше, чем есть) и ведение аудита, что критично для безопасности финансовых приложений.	Задание открытого типа с развернутым ответом	2	1
138.	Что такое «утечка памяти» (memory leak) и как она влияет на долгосрочную работу серверного приложения?	Утечка памяти — ситуация, когда программа выделяет память динамически (например, через new или malloc), но не освобождает ее (через delete или free). В серверном приложении, работающем непрерывно, даже небольшие утечки накапливаются, приводя к постепенному исчерпанию всей доступной	Задание открытого типа с развернутым ответом	2	2

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		оперативной памяти, замедлению работы и в конечном итоге — к аварийному завершению процесса из-за нехватки памяти (OOM).			
139.	При разработке игры требуется реализовать систему инвентаря, где предметы можно добавлять и удалять в произвольном порядке. Объясните, почему для хранения предметов в этом случае <code>std::vector</code> может быть не лучшим выбором, и предложите более подходящий контейнер STL.	<code>std::vector</code> хранит элементы в непрерывной памяти. Удаление предмета из произвольной позиции (не с конца) требует сдвига всех последующих элементов, что имеет сложность $O(n)$ и неэффективно при частых операциях. Более подходящий выбор — <code>std::list</code> (двусвязный список) или <code>std::deque</code> . <code>std::list</code> обеспечивает удаление/вставку в любом месте за $O(1)$, если известен итератор на элемент, что идеально для частого изменения состава инвентаря.	Задание открытого типа с развернутым ответом	3	3
140.	Нужно написать функцию, которая принимает массив целых чисел и его размер, а возвращает индекс максимального элемента. 1) Назовите базовую управляющую конструкцию, которая будет ядром этой функции. 2) Приведите первый шаг алгоритма (после объявления переменных).	Цикл (for или while). Присвоить предполагаемому индексу максимума значение 0 (предположим, что первый	Задание комбинированного типа: практико-ориентированное	1	1

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		элемент — пока максимальный).			
141.	<p>Вы разрабатываете класс FileLogger для записи логов. Объект этого класса должен автоматически открывать файл при создании и закрывать его при уничтожении.</p> <p>1) Какой идиоме C++ это соответствует?</p> <p>2) В каком методе класса следует разместить код закрытия файла?</p>	<p>RAII (Resource Acquisition Is Initialization).</p> <p>В деструкторе класса (~FileLogger()).</p>	Задание комбинированного типа: практико-ориентированное	2	2
142.	<p>Для программы расчета стоимости заказа в интернет-магазине требуется хранить пары "код товара — цена". По коду товара нужно быстро находить цену.</p> <p>1) Назовите подходящий контейнер STL.</p> <p>2) Напишите объявление такого контейнера для строкового кода и целочисленной цены.</p>	<p>std::map или std::unordered_map.</p> <p>std::map<std::string, int></p> <p>priceCatalog;</p>	Задание комбинированного типа: практико-ориентированное	2	2
143.	<p>Алгоритм должен читать числа из текстового файла input.txt до конца файла.</p> <p>1) Какую функцию/метод C/C++ можно использовать для проверки достижения конца файла?</p> <p>2) В каком режиме нужно открыть файл?</p>	<p>feof() (в C) или stream.eof() (в C++).</p> <p>Режим чтения ("r" в C, std::ios::in в C++).</p>	Задание комбинированного типа: практико-ориентированное	1	2
144.	<p>В системе бронирования билетов новые заявки должны обрабатываться в порядке поступления.</p> <p>1) Назовите структуру данных, моделирующую эту очередь.</p> <p>2) Назовите операцию для добавления новой заявки.</p>	<p>Очередь (FIFO).</p> <p>push (или enqueue).</p>	Задание комбинированного типа: практико-ориентированное	1	1
145.	<p>Вы пишете функцию, которая делит два числа. Необходимо обработать случай деления на ноль.</p> <p>1) Какую конструкцию структурного программирования вы используете для проверки делителя?</p> <p>2) Что должна сделать функция, если делитель равен нулю?</p>	<p>Ветвление (оператор if).</p> <p>Вернуть специальное значение (например, 0 или INF), выбросить исключение или вывести сообщение об ошибке.</p>	Задание комбинированного типа: практико-ориентированное	1	1
146.	<p>В программе для учета книг в библиотеке каждая книга имеет уникальный ISBN. Для быстрого поиска книги по ISBN в памяти программы оптимальнее всего</p>	<p>Б.</p> <p>Ассоциативный контейнер (unordered_map) обеспечивает</p>	Задание комбинированного типа с выбором одного ответа и обоснованием	2	3

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	использовать: А) Массив структур Book, поиск перебором. Б) Ассоциативный контейнер, где ключ — ISBN (std::unordered_map<std::string, Book>). В) Односвязный список книг. Г) Стек из книг.	поиск по ключу в среднем за $O(1)$, что значительно быстрее перебора в массиве ($O(n)$) или списке для больших каталогов. Это практическое решение для задачи частого поиска.			
147.	При разработке драйвера устройства важно гарантировать, что ресурсы (память, дескрипторы) будут освобождены даже в случае аварийного завершения функции. Какой подход в C++ для этого наиболее надежен? А) Использование объектов с деструкторами (принцип RAII). Б) Размещение всех инструкций освобождения в конце функции. В) Надежда на то, что операционная система все почистит. Г) Использование оператора goto для перехода к блоку очистки.	А. Принцип RAII гарантирует вызов деструктора при выходе из области видимости (даже при выбросе исключения), что обеспечивает надежное и автоматическое освобождение ресурсов. Это стандартная и безопасная практика в C++.	Задание комбинированного типа с выбором одного ответа и обоснованием	2	4
148.	Алгоритму необходимо обработать все элементы коллекции, но порядок обработки не важен. Какую управляющую конструкцию следует выбрать? А) Последовательность условных операторов (if-else). Б) Цикл (for, while, range-based for). В) Рекурсивный вызов функции. Г) Оператор безусловного перехода (GOTO).	Б. Цикл — это базовая и эффективная конструкция для итерации по элементам коллекции. Он явно выражает намерение "сделать что-то для каждого элемента", код становится понятным и предсказуемым.	Задание комбинированного типа с выбором одного ответа и обоснованием	1	4
149.	Для хранения истории выполненных команд в графическом редакторе (чтобы реализовать Undo) оптимальна структура, работающая по принципу: А) FIFO (First In, First Out). Б) LIFO (Last In, First Out). В) Случайный доступ по индексу. Г) Доступ по ключу-хэшу.	Б. Операция Undo отменяет последнюю выполненную команду, затем предпоследнюю и т.д. Это напрямую соответствует принципу LIFO,	Задание комбинированного типа с выбором одного ответа и обоснованием	1	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
		который реализуется структурой данных "стек".			
150.	<p>Вы разрабатываете модуль, который должен одинаково работать с разными типами документов (PDF, DOCX). Новые форматы могут добавляться позже. Какой принцип ООП является основой для такого дизайна?</p> <p>А) Полиморфизм (работа через общий базовый класс или интерфейс Document).</p> <p>Б) Инкапсуляция (скрытие данных каждого формата).</p> <p>В) Наследование (создание длинной цепочки классов).</p> <p>Г) Использование глобальных переменных для хранения типа документа.</p>	<p>А. Полиморфизм позволяет основному коду вызывать общие методы (например, open(), save()), не зная конкретного типа документа. Это дает возможность добавлять новые форматы (новые классы), не изменяя существующий код обработки, что соответствует принципу открытости/закрытости.</p>	Задание комбинированного типа с выбором одного ответа и обоснованием	2	4
151.	<p>При проектировании высоконагруженного сервера выбор сделан в пользу языка C++. Важно избежать ситуаций, когда поведение программы становится непредсказуемым из-за ошибок программиста. Какая из перечисленных проблем наиболее опасна в этом контексте?</p> <p>А) Использование длинных, но корректных имен переменных.</p> <p>Б) Неопределенное поведение (UB), например, выход за границы массива.</p> <p>В) Наличие комментариев в коде.</p> <p>Г) Использование стандартной библиотеки STL.</p>	<p>Б. Неопределенное поведение (UB) — самая опасная категория ошибок. Оно делает всю программу непредсказуемой: она может работать, падать или вести себя странно в зависимости от компилятора, оптимизаций или фазы луны. Для сервера, требующего стабильности 24/7, это недопустимо.</p>	Задание комбинированного типа с выбором одного ответа и обоснованием	2	4
152.	<p>Требуется считать из бинарного файла массив структур SensorData. Файл был создан на той же машине. Для максимальной скорости и простоты следует использовать:</p> <p>А) Чтение каждой структуры по полям в цикле.</p> <p>Б) Чтение всего блока данных сразу одной операцией (fread или read).</p>	<p>Б. Чтение всего массива структур одной операцией ввода-вывода (fread(&array, sizeof(SensorData), count, file)) является наиболее</p>	Задание комбинированного типа с выбором одного ответа и обоснованием	2	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	В) Конвертация файла в текстовый формат и чтение как текста. Г) Ручной разбор каждого байта.	эффективным по скорости, так сводит к минимуму число системных вызовов и накладных расходов. Это стандартная практика для работы с бинарными данными, если формат файла соответствует структуре в памяти.			
153.	Вы разрабатываете алгоритм сортировки списка пользователей по возрасту. Какие из перечисленных принципов структурного программирования будут использованы в реализации классического алгоритма сортировки (например, "пузырьком" или "быстрой")? А) Следование (последовательность шагов: сравнение, обмен). Б) Ветвление (сравнение двух элементов: if (a > b)). В) Цикл (повторение проходов по массиву). Г) Рекурсия (используется в быстрой сортировке, но не во всех алгоритмах). Д) Безусловный переход (GOTO).	АБВ А) Следование — основа любого алгоритма: шаги выполняются в определенном порядке. Б) Ветвление — необходимо для сравнения элементов и принятия решения об обмене. В) Цикл — обязателен для многократного прохода по данным.	Задание комбинированного типа с выбором нескольких ответов и обоснованием	2	4
154.	При проектировании системы учета сотрудников с помощью ООП вы создаете класс Employee. Какие из следующих элементов целесообразно сделать закрытыми (private)? А) Поле salary (зарплата). Б) Метод calculateTax() (расчет налога). В) Поле internalId (внутренний служебный идентификатор). Г) Метод printInfo() (вывод информации для отчета). Д) Конструктор класса.	АВ А) Зарплата — критические финансовые данные, прямой доступ к которым извне класса нежелателен (инкапсуляция). В) Служебный ID — внутренняя реализация, которая не должна быть частью публичного интерфейса.	Задание комбинированного типа с выбором нескольких ответов и обоснованием	2	4
155.	Для программы, которая читает большой текстовый файл построчно	АВД А) и	Задание комбинированного	2	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	и ищет определенные слова, важна эффективность по памяти. Какие из подходов к чтению файла являются память-эффективными? А) Чтение файла построчно в буфер фиксированного размера. Б) Загрузка всего файла целиком в одну строку (std::string). В) Использование потока ввода (std::ifstream) с методом getline(). Г) Одновременное открытие файла в 10 разных потоков для ускорения. Д) Обработка данных по мере чтения, без сохранения всего файла в памяти.	В) Построчное чтение обрабатывает данные порциями, не загружая весь файл в память. Д) Позволяет не накапливать данные, если их можно обработать сразу.	о типа с выбором нескольких ответов и обоснованием		
156.	Вы пишете функцию для безопасной обработки ввода пользователя. Какие из перечисленных ошибок необходимо предусмотреть и обработать? А) Ввод нечислового значения, когда ожидается число. Б) Ввод значения, выходящего за допустимый диапазон (например, отрицательный возраст). В) Пользователь ввел слишком осмысленный и правильный ответ. Г) Ввод строки, превышающей размер буфера (риск переполнения буфера). Д) Пользователь использовал заглавные буквы.	АБГ А) Ошибка формата ввода — частая проблема, приводит к сбою программы. Б) Логическая ошибка данных — может нарушить работу алгоритма. Г) Переполнение буфера — серьезная уязвимость безопасности.	Задание комбинированного типа с выбором нескольких ответов и обоснованием	2	4
157.	В графическом редакторе требуется реализовать возможность отмены действий (Undo). Какие из перечисленных структур данных теоретически могут быть использованы для хранения истории команд? А) Стек (LIFO). Б) Массив или вектор с индексом текущей позиции. В) Очередь (FIFO). Г) Односвязный список без возможности возврата назад. Д) Двусвязный список (для возможности Redo).	АБД А) Стек — классическая структура для Undo (отмена последнего действия). Б) Массив/вектор — позволяет хранить историю и перемещаться по ней по индексу. Д) Двусвязный список — позволяет удобно перемещаться вперед и назад по истории (Undo/Redo).	Задание комбинированного типа с выбором нескольких ответов и обоснованием	2	4
158.	Разрабатывая игровой движок, вы создаете иерархию классов для	АБГ А) Наследование	Задание комбинированного	2	4

№ задания	Содержание задания	Ответ на задание	Тип задания	Уровень сложности (балл)	№ Темы
	<p>игровых объектов: GameObject -> Character -> Player. Какие преимущества ООП вы получаете от такого дизайна?</p> <p>А) Повторное использование кода (общие поля/методы в базовых классах).</p> <p>Б) Возможность обработки всех объектов через указатель на GameObject (полиморфизм).</p> <p>В) Упрощение математических расчетов в игре.</p> <p>Г) Более четкая организация и понимание кода за счет моделирования отношения "является".</p> <p>Д) Автоматическое увеличение производительности игры.</p>	<p>позволяет избежать дублирования кода (например, координаты, здоровье в GameObject).</p> <p>Б) Полиморфизм позволяет единообразно обновлять, отрисовывать все объекты в цикле.</p> <p>Г) Иерархия отражает логическую структуру: Игрок (Player) является Персонажем (Character), который является Игровым Объектом (GameObject).</p>	о типа с выбором нескольких ответов и обоснованием		

Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процессы формирования компетенций

Характеристика процедуры текущего контроля успеваемости и промежуточной аттестации по дисциплине

Оценивание знаний, умений, навыков и опыта деятельности проводятся на основе сведений, приводимых в матрице соответствия оценочных средств запланированным результатам обучения.

Цель текущего контроля успеваемости и промежуточной аттестации по учебным дисциплинам в семестре – проверка приобретаемых обучающимися знаний, умений, навыков в контексте формирования установленных образовательной программой компетенций в течение семестра.

Шкала оценивания:

«Отлично» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно»: студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки

«неудовлетворительно», допускается оценка «удовлетворительно»: обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «удовлетворительно», «хорошо» и «отлично»: обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно» – выставляется, если при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.

Ответы и решения, обучающихся оцениваются по следующим общим критериям: распознавание проблем; определение значимой информации; анализ проблем; аргументированность; использование стратегий; творческий подход; выводы; общая грамотность.

Обучающиеся обязаны сдавать все задания в сроки, установленные преподавателем. Оценка

«Удовлетворительно» по дисциплине, может выставляться и при неполной сформированности компетенций в ходе освоения отдельной учебной дисциплины, если их формирование предполагается продолжить на более поздних этапах обучения, в ходе изучения других учебных дисциплин.

Текущий контроль осуществляется через систему оценки преподавателем всех видов работ обучающихся, предусмотренных рабочей программой дисциплины и учебным планом.

Критерии оценки теста.

Количество верных ответов:

80-100% -оценка «отлично»: обучающийся демонстрирует глубокое знание учебно-программного материала, умение свободно выполнять задания, усвоивший взаимосвязь основных понятий дисциплины; способный самостоятельно приобретать новые знания и умения; способный самостоятельно использовать углубленные знания;

71-85% -оценка «хорошо»: обучающийся демонстрирует полное знание учебно-программного материала, успешно выполняющий предусмотренные программой задания, показывающий систематический характер знаний по дисциплине и способный к их самостоятельному пополнению и обновлению в ходе дальнейшего обучения в вузе и в будущей профессиональной деятельности;

50-70% -оценка «удовлетворительно»: обучающийся обнаруживает знание основного учебного программного материала в объеме, необходимом для дальнейшего обучения, выполняющего задания, предусмотренные программой, допустившим неточности в ответе, но обладающим необходимыми знаниями для их устранения;

менее 50% -оценка «неудовлетворительно»: обучающийся демонстрирует пробелы в знаниях основного учебного программного материала, допускает принципиальные ошибки в выполнении предусмотренных программой заданий.

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить сформированность планируемых результатов обучения, а также уровень освоения материала обучающимися.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». возможно использовать балльно-рейтинговые оценки.

Основанием для определения оценки на зачете служит уровень освоения обучающимся материала и формирования компетенция, предусмотренных учебным планом.

Успеваемость на зачете определяется оценками: «зачтено»; «не зачтено».

Оценка	Критерии оценивания	Балльно-рейтинговая оценка
«Зачтено»	Обучающийся освоил компетенции дисциплины на 51-100 % и показал хорошие знания изученного учебного материала, логично и последовательно изложил и полностью раскрыл смысл предлагаемого вопроса; продемонстрировал умение применить теоретические знания для решения практической задачи; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	51-100
«Не зачтено»	Обучающийся освоил компетенции дисциплины менее чем на 51% и при ответе на предлагаемый вопрос выявились существенные пробелы в знаниях учебного материала, неумение с помощью преподавателя получить правильное решение практической задачи; не в полном объеме выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	0- 50

Основанием для определения оценки на экзамене служит уровень освоения обучающимся учебного материала, умение решать практические задачи и формирования компетенция, предусмотренных учебным планом.

Успеваемость на экзамене определяется оценками: «отлично»; «хорошо»; «удовлетворительно»; «не удовлетворительно».

Оценка	Критерии оценивания	Балльно-рейтинговая оценка
«Отлично»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 86-100 %, показал глубокие знания учебного материала, логично и последовательно изложил содержание ответов на вопросы билета; продемонстрировал умение иллюстрировать теоретические положения конкретными примерами и свободно выполнять экзаменационные задания; усвоил основную и ознакомился с дополнительной литературой; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	86-100
«Хорошо»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 61-85 %, показал глубокие знания учебного материала, логично и последовательно изложил содержание ответов на вопросы билета, но допустил несущественные неточности; продемонстрировал умение иллюстрировать теоретические положения конкретными примерами и выполнять экзаменационные задания; усвоил основную и ознакомился с дополнительной литературой; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	61-85
«Удовлетворительно»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 51-60 %, показал знания учебного материала в объеме, необходимом для дальнейшего освоения учебных программ, но допустил погрешности в изложении ответов на вопросы билета и при выполнении экзаменационных заданий; ознакомился с основной литературой, рекомендованной программой; справился с контрольными заданиями, предусмотренными рабочей программой дисциплины	51-60
«Не удовлетворительно»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования менее чем на 51 %, обнаружил пробелы в знаниях учебного материала, допустил принципиальные ошибки в	0-50

	выполнении контрольных заданий, предусмотренных рабочей программой дисциплины	
--	---	--

Интегральная оценка

Критерии	Традиционная оценка	Балльно-рейтинговая оценка
5	5	86 - 100
4	4	61-85
3	3	51-60
2 и 1	2, Незачет	0-50
5, 4, 3	Зачет	51-100