

СОГЛАСОВАН

решением Ученого совета
АНО ВО «МБИ
имени Анатолия Собчака»
(протокол от «19» декабря 2024 г. № 6)

АКТУАЛИЗИРОВАН

решением Ученого совета
АНО ВО «МБИ
имени Анатолия Собчака»
(протокол от «25» декабря 2025 г. № 7)

УТВЕРЖДЕН

приказом ректора
АНО ВО «МБИ
имени Анатолия Собчака»
от «27» декабря 2024 г. № 56

УТВЕРЖДЕНА

актуализированная версия
приказом ректора
АНО ВО «МБИ
имени Анатолия Собчака»
от «30» декабря 2025 г. № 59

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ по дисциплине **Объектно-ориентированное программирование**

направление подготовки

38.03.05 Бизнес-информатика

направленность (профиль)

Информационные технологии в бизнесе

уровень образования

высшее образование - бакалавриат

форма обучения

очно-заочная

год набора

2025

Санкт-Петербург
2024

СОДЕРЖАНИЕ

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
2. СТРУКТУРА ФОС ПО ДИСЦИПЛИНЕ.....	3
3. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ	6
4. ШКАЛА ОЦЕНИВАНИЯ РЕЗУЛЬТАТА.....	7
5. ПЕРЕЧЕНЬ ЗАДАНИЙ ПО ДИСЦИПЛИНЕ	8
5.1.ЗАДАНИЯ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ:.....	8
5.2 КОНТРОЛЬНЫЕ ВОПРОСЫ ДЛЯ ТЕКУЩЕЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ	9
5.3 ПРИМЕРНАЯ ТЕМАТИКА РЕФЕРАТОВ.....	23
5.4 ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ.....	25
6. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ	32
7. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ.....	33
7.1.ЗАДАНИЯ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ.....	36
7.2.ЗАДАНИЯ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ.....	36

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1. Фонд оценочных средств предназначен для оценки результатов обучения по учебной дисциплине.

Рабочей программой дисциплины (модуля) предусмотрено формирование следующих компетенций:

Код и наименование компетенции выпускника	Код и наименование индикатора достижения компетенций	Планируемые результаты обучения по дисциплине
ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.1 Знает методы разработки продуктов в сфере информационно-коммуникационных технологий с учетом требований информационной безопасности ОПК-3.2 Знает основы алгоритмизации и программирования ОПК-3.3 Умеет анализировать актуальные угрозы информационной безопасности и учитывать их в процессе создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий ОПК-3.4 Имеет базовые навыки разработки алгоритмов и создания программ на различных языках программирования	<i>Знать:</i> базовые понятия объектно-ориентированного подхода к проектированию и программированию; основные технологии разработки программных продуктов; принципы создания программ для многозадачных операционных систем с помощью визуальных сред программирования и стандартных библиотек классов. <i>Уметь:</i> проводить сравнительный анализ парадигм и технологий программирования и делать обоснованный выбор; проектировать, разрабатывать и тестировать программное обеспечение по техническому заданию в среде визуального программирования; использовать стандартные классы объектно-ориентированных библиотек, пользоваться справочной системой для получения необходимых знаний о стандартных классах. Использует методы и приемы формализации и алгоритмизации поставленных задач, пишет программный код на выбранном объектно-ориентированном языке программирования, выбирает среду программирования <i>Владеть:</i> основными концепциями объектно-ориентированного подхода к программированию; информацией о процессах разработки и жизненном цикле программного обеспечения; инструментарием для разработки программного обеспечения с развитым интерфейсом для многозадачных операционных систем. Способностью руководить процессами разработки программного обеспечения

1.2. *Входной уровень знаний, умений, опыта деятельности, требуемых для формирования компетенции*

- Знать: – основные языки программирования, основные методы разработки программ, стандарты оформления программной документации
- Уметь: Использует основные методы разработки программ. Умеет оформлять программную документацию
- Владеть: Применяет объектно-ориентированное программирование для решения задач в профессиональной деятельности
-

2. СТРУКТУРА ФОС ПО ДИСЦИПЛИНЕ

Оценка проводится методом сопоставления параметров продемонстрированной обучающимся продукта деятельности с заданными эталонами и стандартами по критериям.

Таблица – 1.1. Объекты оценивания и наименование оценочных средств

Номер и наименование разделов/тем	Формы текущего контроля успеваемости / промежуточной аттестации	Объекты оценивания	Вид занятия / Наименование оценочных средств	Форма проведения оценки Устная/ письменная
Тема 1. Понятия: объектно-ориентированной парадигма программирования, платформа, среда разработки.	Текущий контроль	Классы языков программирования. ООП как основная парадигма программирования. Среда разработки. Компиляция и интерпретация. Основные этапы компиляции.	ДП – доклад с презентацией, Т – тест	устная/ письменная
Тема 2. Область применения, терминология ООП.	Текущий контроль	Наиболее часто встречающимися термины в области ООП. Описание преимуществ, и способов моделировать прикладные (бизнес) процессы и предметную область которые предоставляет ООП разработчикам программного обеспечения в процессе проектирования, кодирования и тестирования программных продуктов.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Устная письменная
Тема 3. Классы: основные понятия; Описание класса; Спецификаторы полей и констант класса; Описание полей данных класса.	Текущий контроль	Класс как тип данных, определяемым пользователем. Состав класса. Синтаксис описания базового класса на языке высокого уровня. Особенности применения и синтаксис спецификаторов доступа полей и констант класса; Описание полей данных класса, синтаксис ограничения и требования языка высокого уровня, правила именования полей.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Письменная Устная
Тема 4. Описание основных функциональных элементов класса: Методы, конструкторы, свойства; Виды параметров методов.	Текущий контроль	Методы: виды методов, правила описания и передачи параметров. Классы в C#. Метод Main C#. Среда CLR. Рассмотрение преимуществ использования конструктора при инициализации объекта; Раскрывается использование элемента класса Свойства и как с его помощью определяют характеристики класса в совокупности со способами их задания и получения, то есть методами записи и чтения.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Устная письменная
Тема 5. Иерархии классов Наследование, Абстрактные классы.	Текущий контроль	Иерархии классов, правила наследования различных видов элементов класса, реализация полиморфизма. Механизм Наследования как инструмент упорядочивания и ранжирования классов, то есть объединяя общие для нескольких классов свойства в одном классе и используя его в качестве базового. Виды взаимоотношений между	Доклад с презентацией Тест Коллоквиум Задания Реферат	Устная письменная

Номер и наименование разделов/тем	Формы текущего контроля успеваемости / промежуточной аттестации	Объекты оценивания	Вид занятия / Наименование оценочных средств	Форма проведения оценки Устная/ письменная
		классами. Модель включения-делегирования. Рассмотрение абстрактных классов как механизма задания интерфейса для всей иерархии.		
Тема 6. Пользовательские Интерфейсы; Стандартные интерфейсы .NET; Контейнерные классы.	Текущий контроль	Рассматриваются интерфейс как «крайний случай» абстрактного класса. В нем задается набор абстрактных методов, свойств и индексов, которые должны быть реализованы в производных классах. Интерфейс определяет поведение, которое поддерживается реализующими этот интерфейс классами. Основная идея использования интерфейса.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Письменная устная
Тема 7. Делегаты и использование делегатов, операции с делегатами; События, механизм событий.	Текущий контроль	Описание делегата как особого вида класса, предназначенного для хранения ссылок на методы. Использование делегата для поддержки событий, а также как самостоятельная конструкция языка. Паттерн «наблюдатель», операции с делегатами, передача делегатов в методы в качестве параметра; Обработка исключений при вызове делегатов; Описание события как элемент класса, позволяющий ему посылать другим объектам уведомления об изменении своего состояния.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Письменная устная
Тема 8. Разработка многопоточных приложений (домен, приоритеты потоков, параллельное программирование и библиотека TPL).	Текущий контроль	Рассмотрена особенность многопоточных приложений так процесс может состоять из нескольких доменов (частей) приложения, ресурсы которых изолированы друг от друга. В рамках домена может быть запущено несколько потоков выполнения. Поток (thread1) представляет собой часть исполняемого кода программы. В каждом процессе есть первичный поток, исполняющий роль точки входа в приложение. Класс Thread.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Письменная устная
Тема 9. Асинхронное программирование, Асинхронные делегаты.	Текущий контроль	Асинхронные делегаты, вызов делегатов асинхронно с помощью методов BeginInvoke и EndInvoke, способы задания методов обратного вызова.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Письменная устная
Тема 10. Работа с файлами, классы .NET для работы с потоками.	Текущий контроль	В этой лекции рассматривается обмен данными с файлами и их частным случаем — консолью. Обмен данными реализуется с помощью потоков. Поток (stream) как абстрактное понятие, относящееся к любому переносу данных от источника к приемнику.	Доклад с презентацией Тест Коллоквиум Задания	Письменная устная

Номер и наименование разделов/тем	Формы текущего контроля успеваемости / промежуточной аттестации	Объекты оценивания	Вид занятия / Наименование оценочных средств	Форма проведения оценки Устная/ письменная
		Потоки обеспечивают надежную работу как со стандартными, так и с определенными пользователем типами данных, а также единообразный и понятный синтаксис. Поток определяется как последовательность байтов и не зависит от конкретного устройства, с которым производится обмен (оперативная память, файл на диске, клавиатура или принтер).	Реферат	
Тема 11. Создание оконных приложений.	Текущий контроль	Структура Windows-приложения. Главная программа Windows-приложения, инициализация и завершение приложения, цикл обработки сообщений, набор обработчиков событий. Платформа .NET Framework и библиотека классов.	Доклад с презентацией Тест Коллоквиум Задания Реферат	Письменная Устная
Все темы и разделы:	Промежуточная аттестация	Обобщенные результаты обучения по овладению теоретическими знаниями и практическими навыками	Т – тест (1-30)	Письменная
Итоговый контроль по дисциплине	-	Вопросы 1. Эволюция методологий программирования. Парадигмы программирования. 2. Основные принципы объектного подхода. Абстрагирование. 3. Основные принципы объектного подхода. Инкапсуляция. 4. Основные принципы объектного подхода. Модульность. 5. Основные принципы объектного подхода. Иерархия. 6. Основные принципы объектного подхода. Типизация. 7. Основные принципы объектного подхода. Параллелизм. Сохраняемость. 8. Объект с точки зрения ООП. Состояние. Поведение. 9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов. 10. Объект с точки зрения ООП. Взаимоотношения между объектами.	Вопросы к ГИА	Итоговый контроль по дисциплине

3. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ

Оценка знаний, умений, владений выражается в пятибалльной системе.

Таблица 3.1 – Текущий контроль

№ п/п	Виды работ	Критерии оценивания			
		Неудовлетворительно (2 балла)	Удовлетворительно (3 балла)	Хорошо (4 балла)	Отлично (5 баллов)
1	Работа на лекциях	Отсутствие участия студента в работе на занятии	Единичное высказывание	Высказывание суждений, активное участие в работе на занятии	Высказывание неординарных суждений, активное участие в работе на занятии
2	Работа на практических занятиях, решение общих практических задач	Отсутствие участия в обсуждении, решении, неправильное решение	Единичное высказывание, решение с ошибками	Высказывание суждений, активное участие в ходе решения, правильное решение с отдельными замечаниями	Высказывание неординарных суждений, активное участие в ходе решения, правильное решение без ошибок
3	Работа на практических занятиях, решение индивидуальных практических задач	Отсутствие участия в обсуждении, решении, неправильное решение	Единичное высказывание, решение с ошибками	Высказывание суждений, активное участие в ходе решения, правильное решение с отдельными замечаниями	Высказывание неординарных суждений, активное участие в ходе решения, правильное решение без ошибок

Критерии оценивания формулируются для каждой компетенции и отражают деятельность обучающегося, поддающуюся измерению.

Таблица 3.2 – Обобщенные критерии оценивания освоения компетенции

Неудовлетворительно (2 балла)	Удовлетворительно (3 балла)	Хорошо (4 балла)	Отлично (5 баллов)
Компетенция не освоена. Обучающийся не показывает знания, входящие в состав компетенции, не понимает их необходимость и/или не может их применять	Компетенция освоена. Обучающийся показывает общие знания, входящие в состав компетенции, имеет представление об их применении, умение извлекать и использовать основную (важную) информацию из полученных знаний	Компетенция освоена. Обучающийся показывает полноту знаний, демонстрирует умения и навыки решения типовых задач	Компетенция освоена. Обучающийся показывает глубокие знания, демонстрирует умения и навыки решения сложных задач, умение принимать решения, создавать и применять документы, связанные с профессиональной деятельностью; способен самостоятельно решать проблему/задачу на основе изученных методов, приемов и технологий.

4. ШКАЛА ОЦЕНИВАНИЯ РЕЗУЛЬТАТА

Таблица 4.1 – Шкала критериев оценивания компетенций

Оценка	Содержание
Неудовлетворительно (2 балла)	Демонстрирует непонимание проблемы, не восприятие материала. Работа незакончена и/или это плагиат
Удовлетворительно (3 балла)	Демонстрирует частичное понимание проблемы. Большинство требований, предъявляемых, к заданию выполнены. Владение элементами заданного материала. В основном выполненный материал понятен и носит целостный характер
Хорошо (4 балла)	Демонстрирует значительное понимание проблемы обозначенной дисциплиной. Все требования, предъявляемые к заданию выполнены. Содержание выполненных заданий раскрыто и рассмотрено с разных точек зрения

Оценка	Содержание
Отлично (5 баллов)	Демонстрирует полное понимание проблемы. Все требования, предъявляемые к заданию выполнены. Продемонстрировано уверенное владение материалом дисциплины. Выполненные задания носят целостный характер, выполнены в полном объеме, структурированы, представлены различные точки зрения, продемонстрирован творческий подход

Шкалы оценивания и процедуры оценивания результатов обучения по дисциплине регламентируются Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся по программам высшего образования.

5. ПЕРЕЧЕНЬ ЗАДАНИЙ ПО ДИСЦИПЛИНЕ

5.1. ЗАДАНИЯ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ:

Таблица - 5.1 Перечень заданий текущего контроля и их наименование

Наименование оценочных средств	Содержание задания
Доклад с презентацией	1. Понятия объекта и класса. Основные идеи и сферы применения объектноориентированного подхода. 2. Основные принципы ООП: абстракция, инкапсуляция, наследование и полиморфизм. 3. Правила построения объектной модели: абстрагирование, инкапсуляция, модульность. 4. Правила построения объектной модели: иерархичность, типизация, параллелизм, сохраняемость. 5. Состояние, поведение и идентичность объекта. Понятие и виды операций над объектами. 6. Способы создания объектов. Отношения между объектами – связь и агрегация. 7. Виды отношений между классами. 8. Способы проведения объектно-ориентированного анализа.
коллоквиум	11. Классы. Природа классов. Мета модель. Инстанцирование. 12. Классы. Структура класса. Абстрактные классы и интерфейсы. 13. Классы. Отношения между классами. Ассоциация и агрегация. 14. Классы. Иерархии классов. Зависимость. 15. Модель памяти и структура программы. Классы памяти. Ссылки. 16. Средства инкапсуляции. Инкапсуляция и наследование. Друзья. 17. Модульность, отдельная компиляция, пространства имен, using директива. 18. Представление иерархических отношений. Наследование. 19. Представление иерархических отношений. Агрегация. Зависимость по времени жизни. 20. Правила преобразования типов. Параметрический и виртуальный полиморфизм.
реферат	21. Средства реализации состояния объектов; реализация поведения. 22. Перегрузка операторов. 23. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании. 24. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов. 25. Исключения. Обработка исключений.
задания	Создать класс Rectangle (прямоугольник), содержащий следующие члены класса: – поля: a, b целого типа; – конструктор, позволяющий создать экземпляр класса с заданными длинами сторон; 9 – методы, позволяющие вывести длины сторон прямоугольника на экран, <u>рассчитать периметр прямоугольника, рассчитать площадь прямоугольника;</u>
диспут	1. Типизация данных в программировании. 2. Введение в машинное обучение. Методы и подходы в машинном обучении. 3. Синтаксис описания класса. Элементы, входящие в состав класса. Спецификаторы класса. 4. Поля и константы. Методы класса. 5. Конструктор и деструктор класса. 6. Предмет и цель программной инженерии как области знаний. Основные принципы разработки программных продуктов. 7. Уровни требований к программным продуктам. 8. Понятие и основные принципы конфигурационного управления. 9. Виды тестирования программного обеспечения. 10. Верификация и валидация
Тест	Темы тестовых заданий: Тема 1. Понятия: объектно-ориентированной парадигма

Наименование оценочных средств	Содержание задания
	программирования, платформа, среда разработки. Тема 2. Область применения, терминология ООП. Тема 3. Классы: основные понятия; Описание класса; Спецификаторы полей и констант класса; Описание полей данных класса. Тема 4. Описание основных функциональных элементов класса: Методы, конструкторы, свойства; Виды параметров методов. Тема 5. Иерархии классов Наследование, Абстрактные классы. Тема 6. Пользовательские Интерфейсы; Стандартные интерфейсы .NET; Контейнерные классы. Тема 7. Делегаты и использование делегатов, операции с делегатами; События, механизм событий. Тема 8. Разработка многопоточных приложений (домен, приоритеты потоков, параллельное программирование и библиотека TPL). Тема 9. Асинхронное программирование, Асинхронные делегаты. Тема 10. Работа с файлами, классы .NET для работы с потоками. Тема 11. Создание оконных приложений.

5.2 КОНТРОЛЬНЫЕ ВОПРОСЫ ДЛЯ ТЕКУЩЕЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ

Первая контрольная точка - в форме доклада (письменная).

Темы докладов: Темы докладов с презентацией

1. Объектно-ориентированный язык и мышление. Новая парадигма и способ видения мира;
2. Развитие объектно-ориентированного программирования;
3. Объектные возможности языка C#;
4. Объектные возможности языка Python;
5. Smalltalk – язык объектного программирования;
6. Основы объектно-ориентированного программирования на языке Java;
7. Информация, требуемая для разработки конструктора: структура объекта, условия его создания и использования в программе;
8. Области видимости и время существования переменных. Локальные, статические и глобальные переменные;
9. Интерфейсы. Синтаксис интерфейса. Реализация интерфейса;
10. Виртуальные методы, перегрузка методов и стандартных операторов;
11. Абстрактные классы: назначение, построение, использование;
12. Множественное наследование как способ представления объекта с нескольких точек зрения;
13. Повторное использование кода;
14. Понятие рекурсии; математические рекурсивные функции; примеры рекурсивных процедур;
15. UML: диаграмма классов, диаграмма объектов и диаграмма компонентов.

Методические рекомендации:

Обучающийся может подготовить не более пяти докладов с презентацией. Критерии оценки одного доклада с презентацией: 5 баллов выставляется обучающемуся, если он перечисляет все существенные характеристики обозначенного в вопросе предмета и возможные варианты дальнейшего развития решения проблемы, если это возможно; 4 балла, если студент раскрыл только часть основных положений вопроса, продемонстрировал неточность в представлениях о предмете вопроса; 3 балла, если студент обозначил общую траекторию ответа, но не смог конкретизировать основные компоненты; 2 балла, если студент не продемонстрировал знаний основных понятий, представлений об изучаемом предмете.

Методические указания по написанию доклада с презентацией. Требования, предъявляемые к докладу с презентацией:

1. Объем доклад не должен превышать 5-8 страниц. Печать производится через 1,5 интервала, размер шрифта 14, с выравниванием по ширине. Левое поле листа 30 мм, правое – 10 мм,

верхнее – 20 мм, нижнее 20 мм. Текст должен оформляться абзацами с отступом 1,25 см.

2. Эссе должно восприниматься как единое целое, идея должна быть ясной и понятной.
3. Необходимо писать коротко и ясно. Эссе не должно содержать ничего лишнего, должно включать только ту информацию, которая необходима для раскрытия вашей позиции, идеи.
4. Доклад должен иметь грамотное композиционное построение, быть логичным, четким по структуре.
5. Каждый абзац доклад должен содержать только одну основную мысль.
6. Доклад должен показывать, что его автор знает и осмысленно использует теоретические понятия, термины, обобщения, мировоззренческие идеи.
7. Доклад должен содержать убедительную аргументацию заявленной по проблеме позиции.

контрольная точка - в форме теста (письменная).

1. Дайте определение верификации как вида деятельности
 - а. это процесс доказательства того, что программная система соответствует ожиданиям пользователя или заказчика
 - б. это процесс доказательства того, что программная реализация системы и требования на систему соответствуют друг другу и проектным стандартам
 - в. это процесс поиска и исправления ошибок в проектной документации и программной реализации системы
 - г. это процесс поиска и документирования дефектов программной реализации разрабатываемой системы
2. Модульное тестирование предназначено для
 - а. проверки функционирования каждого независимого программного модуля
 - б. для максимальной изоляции побочного влияния на функционирование модуля со стороны остальных частей системы
 - в. для тестирования модуля в условиях отсутствия воздействия со стороны пользователя
 - г. проверки функционирования одного замкнутого участка программного кода
3. Какие цели и задачи достигаются и решаются в процессе управления конфигурациями?
 - а. управление изменениями данных
 - б. обеспечение совместной работы коллектива разработчиков
 - в. обеспечение целостности технологических процессов
 - г. обеспечение целостности данных
4. Укажите основные свойства роли «Тестировщик»
 - а. создает отчеты о найденных дефектах
 - б. участвует в разработке функциональных требований
 - в. устраняет обнаруженные дефекты в системе
 - г. выявляет дефекты в системе
5. Что такое управление версиями?
 - а. автоматизированный процесс трансформации исходных текстов ПО в пакет исполняемых модулей
 - б. одна из задач конфигурационного управления
 - в. управление версиями файлов
 - г. В системе багтрекинга приоритет задачи, созданной по сигнатуре, может определяться:
 - а. IP-адресом источника
 - б. Комбинацией числа ошибок с данной сигнатурой и числа уникальных IP-адресов
 - в. Давностью возникновения ошибки
7. Для разрешения технического конфликта между членами команда действенным способом может являться

- а. введение общего третьего подхода со взятием ответственности за его внедрение на менеджера
 - б. декларативный выбор менеджером одной из двух рассматриваемых стратегий
 - в. проведение публичных дебатов на тему конфликтного вопроса
8. Роль лидера команды — определять в команде
- а. видение решаемой задачи
 - б. способы коммуникации
 - в. прядок взаимодействия
9. Добавление нового человека в команду
- а. вынуждает повторно проводить предыдущий этап модели для нового человека
 - б. вынуждает повторно проводить процесс по всем этапам модели для нового человека
 - в. не сильно влияет на общий процесс в модели производительности команды
10. Информация, к которой нельзя ограничить доступ:
- а. информация о работе государственных органов
 - б. информация об окружающей среде
 - в. персональные данные субъекта
 - г. информация о здоровье субъекта
- Фонд оценочных средств для проведения промежуточной аттестации
- Страница 4 из 14
11. Какой Федеральный закон устанавливает эквивалентность документа, подписанного собственноручно, и электронного сообщения с ЭЦП?
- а. о персональных данных
 - б. о лицензировании отдельных видов деятельности
 - в. об информации, информационных технологиях и о защите информации
 - г. о безопасности
12. На сколько групп подразделяются классы защищенности автоматизированной системы обработки информации?
- Ответ: 3
13. Каждый пользователь ЛВС должен иметь:
- а. уникальный идентификатор и пароль
 - б. права доступа, позволяющие настраивать свое рабочее место
 - в. свой съемный накопитель информации
 - г. права доступа, позволяющие настраивать антивирусную защиту
14. Персональные данные, над которыми были произведены действия, в результате которых невозможно определить их принадлежность конкретному субъекту ПД, называются:
- а. общедоступными
 - б. общеизвестными
 - в. обезличенными
 - г. специальными
15. Кто должен своевременно обнаруживать факты несанкционированного доступа к персональным данным?
- а. субъект персональных данных
 - б. оператор персональных данных
 - в. регулятор
 - г. контролер
16. Какие утверждения относительно классов WindowListener и WindowEvent верны?
- а. данные классы можно использовать для отслеживания изменения количества компонент в контейнере
 - б. событие WindowListener отслеживает нажатие клавиш клавиатуры
 - в. событие WindowEvent сигнализирует об изменении состояния окна
 - г. данные классы можно использовать для инициирования закрытия окна приложения

17. Для чего нужны классы-обертки?

- а. использование классов-оберток ускоряет работу приложения
- б. переменная класса-обертки может принимать значения null
- в. классы-обертки предоставляют набор статических методов для работы с примитивными типами
- г. при использовании коллекций нельзя использовать значения примитивных типов

18. Почему объектно-ориентированное программирование пришло на смену процедурному программированию?

- а. использование ООП позволяет создавать не зависящие от платформы программы
- б. ООП позволяет уменьшить сложность создания ПО
- в. программы, написанные с использованием ООП, выполняются на компьютере быстрее
- г. ООП позволяет увеличить надежность создаваемого ПО

19. Какие утверждения относительно полей класса верны?

- а. final-поле может быть проинициализировано в конструкторе
- б. идентификатор при объявлении становится составным именем поля
- в. числовое поле ссылочного типа имеет значение по умолчанию null
- г. поля не могут иметь модификатор доступа protected

20. Предположим, вы моделируете автомобиль, описывая его свойства в формате Java-класса. Какие из следующих полей

нужно объявить динамическими (не static)?

- а. максимально допустимая масса для этого класса автомобилей
- б. дата начала выпуска автомобилей
- в. идентификационный номер (VIN)
- г. дата выпуска автомобиля

21. Какие утверждения относительно класса BufferedOutputStream верны?

- а. предоставляет возможность производить многократную запись небольших блоков данных без обращения к устройству вывода
- б. инициирование передачи содержимого буфера на устройство вывода осуществляется самим классом

BufferedOutputStream

- в. инициирование передачи содержимого буфера на устройство вывода можно осуществить явным образом
- г. запись с использованием буфера значительно быстрее, чем без него

22. Which of the following can fill in the blank in this code to make it compile? (Отметьте все варианты)

```
public class Ant {  
    _____ void method() { }  
}
```

- а. default
- б. final
- в. private
- г. Public
- д. String
- е. zzz:

Фонд оценочных средств для проведения промежуточной аттестации

Страница 5 из 14

23. Что из перечисленных конструкций будет скомпилировано без ошибок?

- а. final static void method4() { }
- б. public final int void method() { }
- в. private void int method() { }
- г. static final void method3() { }
- д. void final method() { }

e. `void public method() { }`

24. Какой из перечисленных методов будет успешно компилироваться? (Отметьте все варианты)

a. `public void methodA() { return;}`

б. `public void methodB() { return null;}`

в. `public void methodD() { }`

г. `public int methodD() { return 9;}`

д. `public int methodE() { return 9.0;}`

е. `public int methodF() { return;}`

ж. `public int methodG() { return null;}`

25. Что из перечисленных конструкций будет скомпилировано без ошибок? (Отметьте все варианты)

a. `public void moreA(int... nums) { }`

б. `public void moreB(String values, int... nums) { }`

в. `public void moreC(int... nums, String values) { }`

г. `public void moreD(String... values, int... nums) { }`

д. `public void moreE(String[] values, ...int nums) { }`

е. `public void moreF(String... values, int[] nums) { }`

ж. `public void moreG(String[] values, int[] nums) { }`

26. Given the following method, which of the method calls return 2? (Отметьте все варианты)

```
public int howMany(boolean b, boolean... b2) {  
    return b2.length;  
}
```

a. `howMany();`

б. `howMany(true);`

в. `howMany(true, true);`

г. `howMany(true, true, true);`

д. `howMany(true, {true});`

е. `howMany(true, {true, true});`

ж. `howMany(true, new boolean[2]);`

27. Какие утверждения верны? (Отметьте все варианты)

a. Package private access is more lenient than protected access.

б. A public class that has private fields and package private methods is not visible to classes outside the package.

в. You can use access modifiers so only some of the classes in a package see a particular package private class.

г. You can use access modifiers to allow read access to all methods, but not any instance variables.

д. You can use access modifiers to restrict read access to all classes that begin with the word Test.

28. Given the following `my.school.ClassRoom` and `my.city.School` class definitions, which line numbers in `main()` generate a

compiler error? (Отметьте все варианты)

1: `package my.school;`

2: `public class Classroom {`

3: `private int roomNumber;`

4: `protected String teacherName;`

5: `static int globalKey = 54321;`

6: `public int floor = 3;`

7: `Classroom(int r, String t) {`

8: `roomNumber = r;`

9: `teacherName = t; } }`

1: `package my.city;`

2: `import my.school.*;`

3: `public class School {`

```

4: public static void main(String[] args) {
5: System.out.println(Classroom.globalKey);
6: Classroom room = new Classroom(101, ""Mrs. Anderson");
7: System.out.println(room.roomNumber);
8: System.out.println(room.floor);
9: System.out.println(room.teacherName); } }

```

a. None, the code compiles fine.

б. Line 5

Фонд оценочных средств для проведения промежуточной аттестации

Страница 6 из 14

в. Line 6

г. Line 7

д. Line 8

е. Line 9

29. Какие утверждения верны? (Отметьте все варианты)

a. Encapsulation uses package private instance variables.

б. Encapsulation uses private instance variables.

в. Encapsulation allows setters.

г. Immutability uses package private instance variables.

д. Immutability uses private instance variables.

е. Immutability allows setters.

30. Which are methods using JavaBeans naming conventions for accessors and mutators?

(Отметьте все варианты)

a. public boolean getCanSwim() { return canSwim;}

б. public boolean canSwim() { return numberWings;}

в. public int getNumWings() { return numberWings;}

г. public int numWings() { return numberWings;}

д. public void setCanSwim(boolean b) { canSwim = b;}

31. Что выведет (в консоль) следующий код?

```
1: package rope;
```

```
2: public class Rope {
```

```
3: public static int LENGTH = 5;
```

```
4: static {
```

```
5: LENGTH = 10;
```

```
6: }
```

```
7: public static void swing() {
```

```
8: System.out.print("swing ");
```

```
9: }
```

```
10: }
```

```
1: import rope.*;
```

```
2: import static rope.Rope.*;
```

```
3: public class Chimp {
```

```
4: public static void main(String[] args) {
```

```
5: Rope.swing();
```

```
6: new Rope().swing();
```

```
7: System.out.println(LENGTH);
```

```
8: }
```

```
9: }
```

Ответ: swing swing 10

32. Какие утверждения корректны по отношению к данному коду?(Отметьте все варианты)

```
1: public class Rope {
```

```
2: public static void swing() {
```

```
3: System.out.print("swing ");
```

```

4: }
5: public void climb() {
6: System.out.println("climb ");
7: }
8: public static void play() {
9: swing();
10: climb();
11: }
12: public static void main(String[] args) {
13: Rope rope = new Rope();
14: rope.play();
15: Rope rope2 = null;
16: rope2.play();
17: }
18: }

```

a. The code compiles as is.

б. There is exactly one compiler error in the code.

в. If the lines with compiler errors are removed, the output is climb climb.

г. If the lines with compiler errors are removed, the output is swing swing.

33. What modifiers are implicitly applied to all interface methods? (Отметьте все варианты)

Фонд оценочных средств для проведения промежуточной аттестации

Страница 7 из 14

a. protected

б. public

в. static

г. void

д. abstract

е. default

34. Что выведет (в консоль) следующий код?

```

1: class Mammal {
2: public Mammal(int age) {
3: System.out.print("Mammal");
4: }
5: }
6: public class Platypus extends Mammal {
7: public Platypus() {
8: System.out.print("Platypus");
9: }
10: public static void main(String[] args) {
11: new Mammal(5);
12: }
13: }

```

a. Platypus

б. Mammal

в. PlatypusMammal

г. MammalPlatypus

д. Compile error in line 8.

е. Compile error in line 11.

35. Что можно вставить на место подчеркивания, чтобы код успешно компилировался?

(Отметьте все варианты)

```

public interface CanHop {}
public class Frog implements CanHop {
public static void main(String[] args) {

```

```
frog = new TurtleFrog();
}
}
```

```
public class BrazilianHornedFrog extends Frog {}
public class TurtleFrog extends Frog {}
```

- а. Frog
- б. TurtleFrog
- в. BrazilianHornedFrog
- г. CanHop
- д. Object
- е. Long

36. Which statement(s) are correct about the following code? (Отметьте все варианты)

```
public class Rodent {
    protected static Integer chew() throws Exception {
        System.out.println("Rodent is chewing");
        return 1;
    }
}
```

```
public class Beaver extends Rodent {
    public Number chew() throws RuntimeException {
        System.out.println("Beaver is chewing on wood");
        return 2;
    }
}
```

- а. It will compile without issue.
- б. It fails to compile because the type of the exception the method throws is a subclass of the type of exception the parent method throws.
- в. It fails to compile because the return types are not covariant.
- г. It fails to compile because the method is protected in the parent class and public in the subclass.
- д. It fails to compile because of a static modifier mismatch between the two methods.

37. Which of the following may only be hidden and not overridden? (Отметьте все варианты)

- а. private instance methods
- б. protected instance methods

Фонд оценочных средств для проведения промежуточной аттестации

Страница 8 из 14

- в. public instance methods
- г. static methods
- д. public variables
- е. private variables

38. Выберите корректное утверждение о данном коде:

```
1: interface HasExoskeleton {
2:     abstract int getNumberOfSections();
3: }
4: abstract class Insect implements HasExoskeleton {
5:     abstract int getNumberOfLegs();
6: }
7: public class Beetle extends Insect {
8:     int getNumberOfLegs() { return 6; }
9: }
```

- а. It compiles and runs without issue.
- б. Compile error in line 2.
- в. Compile error in line 4.

г. Compile error in line 7.

д. It compiles but throws an exception at runtime.

39. Which of the following statements about polymorphism are true? (Отметьте все варианты)

а. A reference to an object may be cast to a subclass of the object without an explicit cast.

б. If a method takes a superclass of three objects, then any of those classes may be passed as a parameter to the method.

в. A method that takes a parameter with type `java.lang.Object` will take any reference.

г. All cast exceptions can be detected at compile-time.

д. By defining a public instance method in the superclass, you guarantee that the specific method will be called in the parent class at runtime.

40. Which statements are true for both abstract classes and interfaces? (Отметьте все варианты)

а. All methods within them are assumed to be abstract.

б. Both can contain public static final variables.

в. Both can be extended using the `extend` keyword.

г. Both can contain default methods.

д. Both can contain static methods.

е. Neither can be instantiated directly.

ж. Both inherit `java.lang.Object`.

Инструкция по выполнению. Выбрать один правильный ответ

Критерии оценивания: • 5 баллов выставляется, если обучающийся ответил правильно на 100-85% заданий теста; • 4 балла выставляется, если обучающийся ответил на 84-69 % заданий; • 3 балла выставляется, если обучающийся ответил на 68-50% заданий; 2 балла выставляется, если обучающийся ответил менее, чем на 50 % заданий.

Контрольная точка в форме диспута (устная).

1. Классификация методов проектирования программных продуктов;
2. Место и роль ООП в теории и практике разработки программных систем;
3. Объектно-ориентированное проектирование программных продуктов. Объектно-ориентированный анализ;
4. Средства визуального создания программ и объектно-ориентированное программирование;
5. Объектный подход за пределами программирования;
6. Основные средства анализа и моделирования предметной области в языке UML;
7. Событийно-управляемое, параллельное и распределенное программирование;
8. Компонентно-ориентированный подход. Компоненты и их свойства;
9. Параллельное программирование. Потоки;
10. Сложность разработки ПО. Участники процесса разработки. Проблемы разработки ПО;
11. Методика тестирования объектно-ориентированных программ;
12. Библиотеки подпрограмм: понятие и виды. Схема вызова библиотек. Статическое и динамическое связывание;
13. Организация файловой структуры программы для эффективной разработки, отладки и модификации;
14. Поддержка и сопровождение сложного программного продукта;
15. Организация командной работы над проектом

Критерии оценивания:

Представление точки зрения. Оценивается чёткость речи, использование замечаний, убедительность предложений, а также владение собой: непринуждённость, установление

зрительного контакта с аудиторией, уважительное отношение к оппонентам.

Стратегия выступления. Оценивается организация речи и риторика: тщательность и целенаправленность аргументации, использование времени, убедительные формулировки.

Содержание. Оценивается качество доказательств и эффективность опровержений. Например, логическая и фактическая обоснованность аргументов, защита от контраргументов.

Командная работа. Оценивается сплочённость команды, их поведение: хорошая ли работа вместе, проявление уважения к оппонентам.

Обмен мнениями. Оценивается, предлагали ли участники конструктивные советы оппонентам, с какой вежливостью они делились отзывами.

Глубина поставленных и рассмотренных вопросов. Также важно оценить, насколько верными были ответы и качество этих ответов.

Активность и глубина подготовки отдельных подгрупп, участников и занятия в целом.

Контрольная точка в форме коллоквиума (устная).

Вопросы для коллоквиума

Вопросы к первому коллоквиуму

А) Теоретическая часть.

1. Принципы объектно-ориентированного программирования, их суть. Определение класса. Объект. Скрытие информации. Чем член-функции отличаются от обычных.

Какие существуют типы доступа и чем они отличаются друг от друга. Неявный указатель `this`.

2. Чем член-функции отличаются от обычных. Операция `::`. Чем отличаются член-функции, определенные внутри класса, от функций, определенных вне класса.

3. Конструкторы и деструкторы. Назначение конструктора. Особенности конструктора.

Деструктор. Особенности деструктора.

4. Конструктор копирования. В каком случае необходим конструктор копирования и почему. Чем конструктор копирования отличается от перегрузки операции `=`.

5. Перегрузка операций. Перечислите правила перегрузки операций. Перегрузки операций `+` и `+=`. Чем отличается операция `+` от операции `+=`.

6. Перегрузка операций. Перечислите правила перегрузки операций. Перегрузка операций `[]`, `()`.

7. Перегрузка операций. Перечислите правила перегрузки операций.

Особенности

перегрузки операции (тип). Перегрузка операции `++`.

8. Перегрузка операций. Перечислите правила перегрузки операций. Перегрузка операции `=`. Когда необходимо перегружать операцию `=`. Чем конструктор копирования отличается от перегрузки операции `=`.

9. Дружественность. Что может быть другом класса. Перегрузка операций потокового ввода `>>` и вывода `<<`. Что общего и в чем разница между перегрузкой операции

потокового вывода `<<` и перегрузкой операции потокового ввода `>>`. Можно ли перегрузить операции потокового ввода/вывода как методы класса, ответ аргументировать. Обязательно ли использовать ссылку при перегрузке потоковых операций.

10. Массивы объектов. Какие конструкторы можно использовать для инициализации

статических объектов. Можно ли явно инициализировать массив объектов, определенных в динамической памяти. Как работает конструктор для массива объектов. Как работает деструктор для массива объектов.

11. Агрегированные классы. Каким образом можно определить конструктор

агрегированного класса, если член-данное в нем – указатель на объект другого (используемого) класса? Каким образом агрегированный класс может использовать член-данные используемого класса из части `private`.

12. Агрегированные классы. Каким образом агрегированный класс может использовать член-данные используемого класса из части `private`.

13. Функции- и классы-шаблоны. Что такое порожденная функция. Использование функций-шаблонов и классов-шаблонов. Описать работу компилятора.

14. Статические член-данные класса: как объявляется статическое член-данное в классе, его отличие от остальных член-данных; как инициализируется статическое членданное, как влияет тип доступа на инициализацию статического член-данного; можно

ли изменять статическое член-данное в `main`; как можно получить доступ к статическому член-данному в `main`.

15. Статические методы класса: как объявляются статические методы в классе, их назначение; как определяются статические методы вне класса; особенности статических методов класса

Вопросы ко второму коллоквиуму

А) Теоретическая часть.

1. Базовый и порожденный классы. Тип доступа `protected`. Типы наследования `public`, `protected` и `private`. Ограничения наследования.

2. Простое наследование. Принцип доминирования в иерархии наследования.

3. Конструктор порожденного класса. Его вид. Деструктор порожденного класса.

4. Стандартные преобразования при наследовании.

5. Множественное наследование. Прямые и не прямые базовые классы. Виртуальный базовый класс. Особенности инициализации его ч/данных.

6. Полиморфизм. Раннее и позднее связывание.

7. Виртуальные функции. Чистые виртуальные функции и абстрактный базовый класс.

8. Правила определения виртуальных функций: описание, определение, виртуальные функции порожденного класса.

9. Правила определения виртуальных функций: виртуальные и не виртуальные вызовы.

10. Правила определения виртуальных функций: виртуальные операции, приватные функции.

11. Механизм позднего связывания.

12. Библиотека STL – БСШ. Определение контейнера. Структура БСШ.

13. Библиотека стандартных шаблонов. Шаблоны `vector`, `list`, `set`, `stack`, `queue`

14. Класс «Факультет» (классы `Person`, `Student`, `Teacher`, `Skills`, `Faculty`)

1. Эволюция методологий программирования. Парадигмы программирования.

2. Основные принципы объектного подхода. Абстрагирование.

3. Основные принципы объектного подхода. Инкапсуляция.

4. Основные принципы объектного подхода. Модульность.

5. Основные принципы объектного подхода. Иерархия.

6. Основные принципы объектного подхода. Типизация.

7. Основные принципы объектного подхода. Параллелизм. Сохраняемость.

8. Объект с точки зрения ООП. Состояние. Поведение.

9. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.

10. Объект с точки зрения ООП. Взаимоотношения между объектами.

11. Классы. Природа классов. Мета модель. Инстанцирование.

12. Классы. Структура класса. Абстрактные классы и интерфейсы.

13. Классы. Отношения между классами. Ассоциация и агрегация.
14. Классы. Иерархии классов. Зависимость.
15. Модель памяти и структура программы. Классы памяти. Ссылки.
16. Средства инкапсуляции. Инкапсуляция и наследование. Друзья.
17. Модульность, раздельная компиляция, пространства имен, using директива.
18. Представление иерархических отношений. Наследование.
19. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.
20. Правила преобразования типов. Параметрический и виртуальный полиморфизм.
21. Средства реализации состояния объектов; реализация поведения.
22. Перегрузка операторов.
23. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.
24. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов.
25. Исключения. Обработка исключений

Методические рекомендации:

Обучающийся может подготовить не более двух ответов на вопросы к коллоквиуму. Критерии оценивания одного ответа: 5 баллов выставляется обучающемуся, если он перечисляет все существенные характеристики обозначенного в вопросе предмета и возможные варианты дальнейшего развития решения проблемы, если это возможно; 4 балла, если студент раскрыл только часть основных положений вопроса, продемонстрировал неточность в представлениях о предмете вопроса; 3 балла, если студент обозначил общую траекторию ответа, но не смог конкретизировать основные компоненты; 2 балла, если студент не продемонстрировал знаний основных понятий, представлений об изучаемом предмете

Контрольная точка - в форме заданий (письменная).

Вариант 1 Задание 1. Анализ причин сложности современных программных систем. Основы структурного и объектного подходов к программированию. Преимущества и недостатки объектного подхода. Задание 2. Уровни доступа к членам класса. Задание 3. Конструктор с параметрами. Пример программы. Вариант 2 Задание 1. Основные принципы ООП: абстракция и инкапсуляция. Преимущества инкапсуляции. Задание 2. Объявление объектов. Создание массивов объектов. Пример программы. Задание 3. Конструктор со списком инициализаторов. Пример программы. Вариант 3 Задание 1. Понятие класса и объекта в ООП. Состояние и поведение объекта. Задание 2. Указатели на объекты класса. Динамические объекты и массивы объектов. Пример программы. Задание 3. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы. Вариант 4 Задание 1. Суть наследования и его виды. Задание 2. Синтаксис и семантика конструктора. Указатель this. Пример программы. Задание 3. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы. 25 Вариант 5 Задание 1. Мономорфизм и полиморфизм. Задание 2. Свойства конструкторов. Задание 3. Синтаксис и семантика деструктора. Пример программы. Вариант 6 Задание 1. Структура класса C++. Пример программы. Задание 2. Конструктор по умолчанию. Пример программы. Задание 3. Свойства деструкторов

Вариант 1 Задание 1. Статические переменные и статические функции класса (static). Назначение. Пример программы. Задание 2. Методы класса, дружественные другому классу. Пример программы. Задание 3. Виртуальные базовые классы. Пример программы. Вариант 2 Задание 1. Константные переменные и функции класса (const). Пример программы. Задание 2. Дружественные классы. Пример программы. Задание 3. Замещение членов базового класса. Пример программы. Вариант 3 Задание 1. Функции, дружественные одному классу. Пример программы. Задание 2. Объявление классов при простом наследовании. Спецификаторы наследуемого доступа. Пример программы.

Задание 3. Назначение перегрузки функций. Пример программы. Вариант 4 Задание 1. Функции, дружественные нескольким классам. Пример программы. Задание 2. Конструкторы и деструкторы производных классов (очередность вызова). Пример программы. Задание 3. Перегрузка конструкторов. Пример программы.

Вариант 1 Задание 1. Назначение перегрузки операторов. Правила перегрузки операторов. Задание 2. Механизмы раннего и позднего связывания. Задание 3. Синтаксис шаблонного класса. Пример программы. Задание 4. Ассоциативные контейнеры. Пример. Вариант 2 Задание 1. Перегрузка бинарных операторов. Пример программы. Задание 2. Виртуальные функции. Таблица виртуальных функций. Пример программы. Задание 3. Специализация шаблонов функций и классов. Пример программы. Задание 4. Итераторы. Обратные итераторы. Пример. Вариант 3 Задание 1. Перегрузка унарных операторов. Пример программы. Задание 2. Чисто виртуальные функции и абстрактные классы. Пример программы. Задание 3. Достоинства и недостатки шаблонов. Задание 4. Алгоритмы библиотеки STL. Пример. Вариант 4 Задание 1. Перегрузка оператора доступа к членам класса. Пример программы. Задание 2. Назначение виртуальных деструкторов. Пример программы. Задание 3. Назначение и состав библиотеки STL. Задание 4. Иерархия классов VCL-библиотеки. Вариант 5 Задание 1. Перегрузка и неоднозначность. Пример программы. Задание 2. Назначение шаблонных функций. Задание 3. Последовательные контейнеры. Пример. Задание 4. Классы командных кнопок, флажков и радиокнопок VCL-библиотеки. Вариант 6 Задание 1. Виды и реализации полиморфизма. Задание 2. Конкретизация шаблонной функции. Пример программы. Задание 3. Контейнеры – адаптеры. Пример. Задание 4. Обработка событий средствами VCL-библиотеки

Практическая часть.

1. Класс List (список). Может быть однонаправленный или двунаправленный, с фиктивной головой или нет.

2 независимых класса – узел (Node) и список (могут быть друзьями).

Член-данные для списка: указатель на голову, указатель на хвост (для двунаправленного)

Методы: List (); List (int); List (int*, int) (из массива делаем список); List (const List &);

~List(); Node* FindKey(int) (на входе – ключ, на выходе – указатель на узел, если ключ в списке нашли, NULL – если ключа в списке нет); функции ввода/вывода (можно только перегрузить потоковые); сортировка; добавление элемента (в голову, хвост, на позицию, после ключа), удаление элемента (из головы, хвоста, с позиции, по ключу), нахождение max/min; проверка пустоты списка; очистка списка.

Перегрузка операторов: = (присвоение); [] ; +(List) (в голову, в хвост); ==, != (сравнение списков); потокового ввода/вывода.

2. Классы Student и Teacher, наследники класса Person.

а) Определить родительский класс Person (член-данные: string fio, address; int year; методы: конструкторы; void Input() (ввод данных с клавиатуры); void Show() (вывод данных на консоль).

б) Член-данные класса Student: int group, kurs; string spec.

Методы класса Student: конструкторы; void Input() (ввод данных с клавиатуры); void Show() (вывод данных на консоль), Get_group(); Get_kurs();

Перегрузить потоковый вывод.

в) Член-данные класса Teacher: string kafedra, prof.

Методы класса Student: конструкторы; void Input() (ввод данных с клавиатуры); void Show() (вывод данных на консоль), Get_Kafedra(); Get_Prof();

Перегрузить потоковый вывод.

3. Агрегированный класс Faculty.

Член-данные класса Faculty: vector<Student> s; vector <Teacher> t; string

long_name, short_name;

Методы: конструкторы; добавление студента/преподавателя; удаление студента/преподавателя; вывод (всех, всех студентов, всех преподавателей, конкретной группы, конкретного курса, конкретной кафедры); получить все данные студента/преподавателя; вывод общего количества студентов/преподавателей.

4. БСШ, класс vector.

Задать массив случайных целых чисел и

а) выбрать из него в другой массив уникальные числа (т.е. встречающиеся только один раз); б) выбрать из него в другой массив числа, кратные 5.

Задать массив строк и удалить из него самую длинную строку.

Задать массив чисел Фибоначчи и выбрать из него в другой массив четные числа.

5. БСШ, класс list.

Задать список фамилий и

а) Добавить перед самой длинной фамилией фамилию fam; б) отсортировать его по возрастанию; в) найти и удалить самую длинную фамилию.

Задать 2 упорядоченных списка фамилий, слить их в один общий упорядоченный список.

6. БСШ – класс set

Задать 2 предложения. Выяснить, в каком предложении разных букв больше.

Практическая часть.

1. Класс Circle (Круг) .

Член-данные: double x, y (координаты центра) и double r (радиус).

Методы: все конструкторы (если не требуется конструктора копирования – объяснить почему); ввод и вывод; нахождение площади; проверка, попадает ли заданная точка в круг.

Перегрузка операторов: операции + и += (умножение круга на число – центр не меняется, радиус умножается на число); сравнение (== и !=; два круга равны, если равны их радиусы); потокового ввода >> и потокового вывода <<.

2. Класс FreeVector (Свободный вектор) .

Член-данные: double x, y, z (координаты вектора).

Методы: все конструкторы (если не требуется конструктора копирования – объяснить почему); ввод и вывод; нахождение длины вектора.

Перегрузка операторов: операции +, += (сумма векторов); операции -, -= (разность векторов); операции * (скалярное произведение векторов); сравнения (== и !=); потокового ввода >> и потокового вывода <<.

3. Класс Complex.

Член-данные: double x, y (действительная и мнимая часть комплексного числа).

Методы: все конструкторы (если не требуется конструктора копирования – объяснить почему); ввод, вывод, модуль; комплексно-сопряженное;

Перегрузка операторов: арифметические операции (+, +=, -, -=, *, *=, /, /=); сравнение (== и !=); потокового ввода >> и потокового вывода <<.

4. Класс массив Array.

Член-данные (2 варианта):

А) указатель на целое, количество взятой памяти (mem), число элементов массива (n);

Б) указатель на целое, число элементов массива (памяти берётся столько же) (n);

Методы:

Для А – Array(int mem_=1, int n_=0) (можно добавить параметр – способ заполнения) , для Б – Array(int n_=0) (можно добавить параметр – способ заполнения);

Array(int*, int) (создание объекта класса Array из целочисленного массива);

Array(const Array&); ~Array(); функции ввода/вывода; формирование случайного массива; функция поиска (на входе – ключ, на выходе – индекс, если ключ присутствует в массиве, -1, если ключа в массиве нет); функция нахождения max/min

элемента (возвращает индекс элемента); добавление ключа на позицию;
удаление позиции (изменяется существующий объект, создается новый объект).

Перегрузка операторов:

= (присвоение); [] (возвращает ссылку); +(int key) – добавление ключа key в конец массива; +=(int key) – добавление ключа key в конец массива; +(Array) – добавление массива в конец массива; +=(Array) – добавление массива в конец массива; -(int key) – удаление ключа key (первое вхождение); –=(int key) – удаление ключа key (первое вхождение); ==, !=(сравнение); потоковый ввод/вывод;

5. Класс String

Член-данные: char *Str (строка), int Len (размер строки Str).

Методы: String(char *); String(int); String(const String&); ~String().

Перегрузка операторов: = (присвоение); [] (возвращает ссылку); >> (потоковый ввод); << (потоковый вывод); +, += (конкатенация); == (сравнение строк).

6. Класс-шаблон Stack на основе массива.

Член-данные: массив, размер массива, реальное число элементов стека.

Методы: Stack(int k=50); Stack(const Stack<T> &); ~ Stack(); проверка на пустоту; проверка на заполненность; добавить в стек элемент; удалить из стека элемент; взять значение с вершины стека;

Перегрузка операторов: = (присвоение); потоковый вывод <

Критерии оценивания. «отлично» – студент правильно выполнил задание. Показал отличные владения навыками применения полученных знаний и умений при решении задач в рамках усвоенного учебного материала. Ответил на все дополнительные вопросы; «хорошо» – студент выполнил задание с небольшими неточностями. Показал хорошие владения навыками применения полученных знаний и умений при решении задач в рамках усвоенного учебного материала. Ответил на большинство дополнительных вопросов; «удовлетворительно» – студент выполнил задание с существенными неточностями. Показал удовлетворительное владение навыками применения полученных знаний и умений при решении задач в рамках усвоенного учебного материала. При ответах на дополнительные вопросы было допущено много неточностей; «неудовлетворительно» – при выполнении задания студент продемонстрировал недостаточный уровень владения умениями и навыками при решении задач в рамках усвоенного учебного материала. При ответах на дополнительные вопросы было допущено множество неточностей

ПРИМЕРНАЯ ТЕМАТИКА РЕФЕРАТОВ

Примерный список тем рефератов:

1. Разработка библиотеки классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований (окружность, треугольник, прямоугольник).
Предусмотреть удобный интерфейс.
2. Разработка библиотеки классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований (шар, пирамида, куб).
Предусмотреть удобный интерфейс.
3. Разработка библиотеки классов, реализующих графические примитивы на плоскости с реализацией операций над множествами.
4. Разработка программы, демонстрирующей движение окружностей, с использованием принципов объектно-ориентированного программирования. По траектории, представляющей собой окружность, происходит движение шаров. Каждый шар характеризуется скоростью, направлением движения и радиусом. Шары имеют синий цвет. Однако, если происходит пересечение шаров, то все шары, попадающие в пересечение, закрашиваются красным цветом на время пересечения. Предусмотреть добавление и удаление шаров, изменение их параметров.
5. Разработка библиотеки классов, реализующих векторы в n-мерном пространстве.
6. Разработка библиотеки классов, реализующих строки и операции над ними, включая

работу с регулярными выражениями.

7. Реализация различных типов графов и операций над ними.

8. Разработка объектной модели и программы, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...).

9. Разработка системы классов, описывающих различные транспортные средства.

10. Моделирование замкнутой биологической системы (корм, травоядное, хищник).

11. Моделирование муравейника (несколько типов муравьёв, источники питания, внешние раздражители,...).

12. Моделирование планетарной системы.

13. Система классов для реализации матричной арифметики.

14. Разработка библиотеки классов для работы с комплексными числами. Написать тестовую программу для работы с библиотекой «Комплексное число».

15. Разработка программы для игры «Шахматы».

16. Моделирование дорожного движения на заданной карте дорог. Программа – эмулятор

гоночной трассы (со случайными препятствиями, с 4 автомобилями). На бесконечной трассе шириной 10 единиц двигаются 4 автомобиля (вернее движется сама трасса сверху вниз). Сверху появляются препятствия (не более 3-х общей шириной не более 6 единиц). Препятствия реализовать объектами, создавать их динамически, при выходе их за нижнюю границу удалять из памяти. Автомобили также реализовать в виде объектов (ширина автомобиля 2 единицы), автомобили могут двигаться и по горизонтали, и по вертикали, не накладываясь друг на друга. Для автомобилей предусмотреть обязательное наличие между препятствиями двух «коридоров» по 2 единицы шириной. При столкновении автомобиля с препятствием автомобиль удаляется с трассы и из памяти. Игра заканчивается, когда на трассе остается один автомобиль (о чем вывести сообщение). Всеми автомобилями управляет компьютер.

17. Программа – эмулятор пианино (или любого другого музыкального инструмента).

Стандартными средствами языка C++ создать объект нота, от которого создать 7 потомков – 7 нот. Вывести на экран клавиатуру (клавиши пианино). Клавиатуру также реализовать с помощью объектов (иерархию объектов клавиатуры выбрать произвольно).

18. Программа – дизайнер комнаты (с заранее заданными элементами мебели).

19. Разработка программы для игры «Жизнь».

20. Разработка объектной модели и программы «Автозаправка», в которой присутствуют объекты, представляющие транспорт, а также объекты, представляющие топливо.

21. Разработка программы для игры «Ханойская башня». На игровом поле находятся 3 стержня, на которые можно нанизывать кольца только в порядке убывания размера кольца. Изначально все кольца находятся в порядке убывания (самое широкое – снизу, самое узкое - наверху) на первом стержне. Цель игры – переместить все кольца на другой столбик. Сложность игры зависит от количества колец (минимальное количество колец - три). Создать объекты для кольца (с переменным размером) и стержня, у которого будет свойство – семейство колец.

22. Разработка объектной модели и программы «Аквариум».

Методические рекомендации по написанию реферата, требования к оформлению

Реферат (от лат. *refereo* – сообщаю) – краткое изложение в письменном виде научно-исследовательской работы студента, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее. Содержание реферата должно быть логичным, а изложение материала должно носить проблемно-тематический характер.

Каждая работа, носящая реферативный характер, должна быть оформлена в соответствии с Государственным Образовательным Стандартом (ГОСТ).

Структура работы:

- титульный лист;
- оглавление;

- введение (указывается актуальность, цель, задачи);
- не менее 3х глав;
- выводы (личное мнение);
- приложения (если есть);
- список используемой литературы.

Каждый новый раздел начинается с новой страницы.

Во введении обязательно должны быть указаны следующие параметры: тема реферата; предмет и объект исследования; цель; задачи; актуальность; методы исследования.

В основной части формулируются ключевые понятия и положения, вытекающие из анализа теоретических источников (точек зрения, моделей, концепций), документальных источников и материалов практики, экспертных оценок по вопросам исследуемой проблемы, а также результатов эмпирических исследований. Реферат носит исследовательский характер, содержит результаты творческого поиска автора.

В заключении (1-2 страницы) подводятся главные итоги авторского исследования в соответствии с целью и задачами реферата делаются выводы или даются практические рекомендации по разрешению исследуемой проблемы.

Требования к оформлению работы:

- объём 15-20 страниц;
- поля – по 2 см сверху и снизу, 3 см – слева, 1,5 см – справа;
- выравнивание текста по ширине;
- номера страниц – внизу страницы, справа; номер на первой странице не указывается;
- шрифт – «Times New Roman»;
- основной текст – 14 кегль, междустрочный интервал – 1,5 (основной текст, между абзацами);
- абзацный отступ – 1,25;
- расстановка переносов не применяется;
- все лишние пробелы убираются, между словами должен быть только один пробел; знаки препинания (за исключением тире) ставятся сразу же за предваряющим его словом без пробела;
- ссылки – затекстовые (вынесенные за текст реферата и оформленные как список использованной литературы в алфавитном порядке), использование автоматических постраничных ссылок не допускается;
- не менее 5 источников;
- ссылки оформляются в соответствии с ГОСТ Р 7.0.100-2018 (приложение 2);
- для связи с текстом порядковый номер библиографической записи из списка литературы указывают в отсылке, которую приводят в квадратных скобках в строку с текстом. Если ссылку приводят на конкретный фрагмент текста, в отсылке указывают порядковый номер и страницы, на которых помещен объект ссылки. Сведения разделяют запятой. Если отсылка содержит сведения о нескольких затекстовых ссылках, группы сведений разделяют знаком точка с запятой.

Критерии оценки реферата. При оценке реферата преподаватель руководствуется следующими критериями: – соответствие содержания текста выбранной теме; – наличие четкой и логичной структуры; – качество аналитической работы, проделанной при написании реферата; 12 – использование адекватных выбранной теме литературных источников; – самостоятельность, невторичность текста; – обоснованность сделанных автором реферата выводов, соответствие их поставленной цели; – отсутствие орфографических, пунктуационных, стилистических, а также фактических ошибок; – соответствие оформления работы предъявляемым требованиям; – сдачи реферата в установленный срок.

5.3 ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы **6 семестр**

Итоговый тест (с ответами) для проверки сформированности компетенций

ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации;

ОПК-3.1 Знает методы разработки продуктов в сфере информационно-коммуникационных технологий с учетом требований информационной безопасности

1. Прочитайте текст и выберите правильный ответ Совокупность пошаговых процедур, определяющих последовательность технологических операций проектирования; критериев и правил, используемых для оценки результатов выполнения технологических операций; графических и текстовых средств, используемых для описания проектируемой системы

- а) технологии проектирования
- б) методологии проектирования
- в) жизненного цикла
- г) стандарта проектирования

Правильный ответ: а)

2. Прочитайте текст и выберите правильный ответ Что является элементом технологического процесса проектирования ИС

- а) технологические карты
- б) стандарты проектирования
- в) технологическое оборудование
- г) технологическая операция

Правильный ответ: г)

3. Прочитайте текст и выберите правильный ответ. Часть реального мира, подлежащая изучению с целью организации управления и автоматизации

- а) физическая область
- б) географическая область
- в) выделенная область
- г) предметная область

Правильный ответ: г)

4. Установите правильную последовательность разработки веб-сайта

- а) Анализ и проектирование
- б) Написание кода
- в) Наполнение контентом
- г) Тестирование
- д) Публикация
- е) Поддержка

Правильный ответ: а) – б) – в) – г) – д) – е)

5. Закончите фразу. Ответ следует записать с маленькой буквы, в именительном падеже. предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей – это

Правильный ответ Алгоритм

6.Прочитайте текст и выберите правильный ответ. Параметр цикла For в Питоне может получать значения...

- 1. 2,3,4,5.
- 2. 2,4,6,8,10
- 3. 1,3,5,7,9
- 4. Нет правильного ответа

Правильный ответ 1

7.Прочитайте текст и выберите правильный ответ.

Способы описания алгоритмов. Выберите лишний

- 1. циклический.
- 2. словесный
- 3. графический

- 4. алгоритмический
- 5. программный

Правильный ответ 1

8. Установите правильную последовательность разработки веб-сайта на базе CMS

- а) Пользователь запрашивает страницу;
- б) Движок получает информацию из базы данных или файлов на диске.
- в) Обращение пользователя производится к движку сайта (например, index.php)
- г) Веб-сервер передает созданную страницу в браузер пользователя
- д) Движок сайта генерирует html-страницу в соответствии с шаблоном сайта и передает ее

веб-серверу

Правильный ответ: а) – в) – б) – д) – г)

9. Установите соответствие. понятий и определений веб-технологий

- 1) Back Screen
- 2) Screen Glide
- 3) Pop-up

4) Таргетирование (targeting)

а) всплывающее окно, появляющиеся над содержанием собственно сайта

б) блок в отдельном окне, который загружается за контентом сайта и активируется только после закрытия основного окна

в) блок раскрывается, если навести на него курсор, и захлопывается, если увести

г) направленный показ баннеров для определенной аудитории

Правильный ответ: 1) – б) 2) – в) 3) – а) 4) – г)

10. Закончите фразу. Ответ следует записать с маленькой буквы, в именительном падеже. Модель данных, при которой отношения представляются в виде строго упорядоченного набора структур записей, каждая структура имеет вид дерева называется _____ модель

Правильный ответ: иерархическая

ОПК-3.2 Знает основы алгоритмизации и программирования

11. Прочитайте текст и выберите правильный ответ При реализации информационной технологии поддержки принятия решений решающая роль в выборе окончательного варианта решения отводится

- а) информационной системе
- б) компьютерной программе
- в) цифровому процессору
- г) человеку
- д) искусственному интеллекту

Правильный ответ: г)

12. Прочитайте текст и выберите правильный ответ Совокупность работ от формирования исходных требований к информационной системе до ввода ее в действие называется

- а) жизненный цикл
- б) теоретический этап
- в) подготовительный цикл
- г) стадия внедрения
- д) процесс создания

Правильный ответ: а)

13. Укажите прямую последовательность этапов жизненного цикла ИС

- а) удаление
- б) проектирование
- в) внедрение
- г) реализация
- д) эксплуатация
- е) планирование и анализ требований

Правильный ответ: е) - б) - г) - в) - д) - а)

14. Установите соответствие: основных свойств ИС и их определений

- 1) Свойство относительности
- 2) Делимость
- 3) Свойство целостности

а) устанавливает, что состав элементов, взаимосвязей, входов, выходов, целей и ограничений ЭИС зависит от целей применения

б) указывает на согласованность цели функционирования всей ЭИС с целями функционирования ее подсистем и элементов

в) означает, что ЭИС можно представить состоящей из относительно самостоятельных частей - подсистем, каждая из которых может рассматриваться как система

Правильный ответ: 1) – б) 2) – в) 3) – а)

15. Установите соответствие аббревиатур и названий методик графического анализа информационных систем

- 1) SADT 2) DFD 3) ERD 4) STD

а) диаграммы потоков данных

б) технология структурного анализа и проектирования

в) диаграммы переходов состояний

г) диаграммы «сущность-отношение»

Правильный ответ: 1) – б) 2) – а) 3) – г) 4) – в)

16. Установите типовую последовательность операций информационно-технологического процесса в информационной системе

а) реализация обратной связи для коррекции входной информации

б) вывод информации для представления потребителям или передачи в другую систему

в) обработка входной информации и представление ее в удобном виде

г) ввод информации из внешних или внутренних источников

Правильный ответ: а) - в) - б) - г)

ОПК-3.3 Умеет анализировать актуальные угрозы информационной безопасности и учитывать их в процессе создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий

17. Установите правильное соответствие терминов и их определений, касающихся подсистем обеспечения ИС

1) Техническое обеспечение

2) Информационное обеспечение

3) Программное обеспечение

4) Организационное обеспечение

5) Правовое обеспечение

а) совокупность правовых норм, определяющих создание, юридический статус и функционирование ЭИС

б) комплекс аппаратных средств-компьютеры, устройства сбора, обработки и передачи информации, линии связи, оргтехника

в) совокупность системы классификации и кодирования информации, схем информационных потоков и методологии построения баз данных

г) совокупность алгоритмов и программ для реализации целей информационной системы и нормального функционирования технических средств

д) совокупность методов и средств, регламентирующих взаимодействие работников между собой и с техническими средствами в процессе эксплуатации ЭИС

Правильный ответ: 1) – б) 2) – в) 3) – г) 4) – д) 5) – а)

18. Укажите прямую последовательность Расположите в правильной хронологической последовательности возникновения типов (этапов развития) информационной технологии

а) электрическая

б) компьютерная

- в) ручная
- г) электронная
- д) механическая

Правильный ответ: в) - д) - а) - г) - б)

19. Закончите фразу. Ответ следует записать с маленькой буквы, в именительном падеже Автоматизированная информационная среда, предназначенная для организации, хранения, пополнения, поддержки и представления пользователям информации в соответствии с их запросами называется _____

Правильный ответ: информационная система

20. Закончите фразу. Ответ следует записать с маленькой буквы, в именительном падеже Процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, называется _____

Правильный ответ: информационная технология

ОПК-7.3. Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности

21. Установите правильное соответствие Установите соответствие понятий

1) MySQL 2) MS SQL 3) PostgreSQL 4) SQL

- а) Реляционная система управления базами данных с открытым исходным кодом
- б) объектно-реляционная система управления базами данных с открытым исходным кодом
- в) Язык структурированных запросов, предназначенный для работы с реляционными базами

данных

г) реляционная система управления базами данных, разработанная компанией Microsoft

Правильный ответ: 1) – а) 2) – г) 3) – б) 4) – в)

22. Прочитайте текст и выберите правильный ответ. Графическое задание алгоритма - это?

- а) способ представления алгоритма с помощью геометрических фигур;
- б) представление алгоритма в форме таблиц и расчетных формул;
- в) система обозначений и правил для единообразной и точной записи алгоритмов и их

исполнения.

Правильный ответ а

23. Прочитайте текст и выберите правильный ответ. Алгоритм - это?

а) некоторые истинные высказывания, которые должны быть направлены на достижение поставленной цели;

б) отражение предметного мира с помощью знаков и сигналов, предназначенное для конкретного исполнителя;

в) понятное и точное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи или цели.

Правильный ответ в

24. Прочитайте текст и выберите правильный ответ В расчете на кого должен строиться алгоритм?

- а) в расчете на ЭВМ;
- б) в расчете на умственные способности друга;
- в) в расчете на конкретного исполнителя.

Правильный ответ в

25. Установите соответствие для типовых операций обработки данных в БД

- 1) Чтение
- 2) Добавление
- 3) Изменение
- 4) Удаление

а) Обновление существующих данных в базе для корректировки информации

б) Уничтожение данных из базы для очистки от ненужной информации

- в) Извлечение данных из базы для получения информации
- г) Запись новых данных в базу для внесения новой информации

Правильный ответ: 1) – в) 2) – г) 3) – а) 4) – б)

26. Установите соответствие свойств транзакции в БД

- 1) Атомарность
- 2) Изоляция
- 3) Согласованность
- 4) Долговечность

а) Транзакция переводит базу данных из одного согласованного состояния в другое.

б) Транзакция выполняется полностью или не выполняется вовсе. Если происходит сбой, все изменения отменяются.

в) После успешного завершения транзакции её результаты сохраняются даже при сбоях системы.

г) Результаты транзакции невидимы для других транзакций до её завершения.

Предотвращает взаимное влияние параллельных транзакций

Правильный ответ: 1) – б) 2) – г) 3) – а) 4) – в)

ОПК-3.4 Имеет базовые навыки разработки алгоритмов и создания программ на различных языках программирования

27. Прочитайте текст и выберите правильный ответ Какую смысловую нагрузку несет блок?

- а) логический блок;
- б) блок автоматического ввода/вывода информации;
- в) блок вывода информации на печатающее устройство

Правильный ответ б

28. Установите соответствие операторов SQL и действий

- 1) CREATE VIEW
- 2) DROP ASSERTION
- 3) ALTER TABLE
- 4) CREATE TABLE

а) создание таблицы

б) добавления, изменения или удаления столбцов в таблице

в) создание виртуальной таблицы

г) удаление существующего ограничения из базы данных

Правильный ответ: 1) – в) 2) – г) 3) – б) 4) – а)

29. Установите правильное соответствие для моделей данных в БД

- 1) Иерархическая
- 2) Сетевая
- 3) Реляционная
- 4) Объектно-ориентированная

а) Данные организуются в таблицы, взаимосвязь между которыми осуществляется через ключи

б) Объединяет данные и методы их обработки в единую структуру. Используется в приложениях, которые требуют высокой степени соответствия между программными объектами и хранящейся информацией

в) Представляет данные в виде древовидной структуры, где каждый уровень подчиняется верхнему

г) Позволяет каждому элементу иметь несколько родительских связей

Правильный ответ: 1) – в) 2) – г) 3) – а) 4) – б)

30. Прочитайте текст и выберите правильный ответ Оператор CREATE TABLE в БД

- а) удаляет отношение
- б) создает отношение
- в) умножает отношения

г) вычитает отношения

Правильный ответ: б)

Инструкция по выполнению. При ответе на вопросы:

Выбрать один правильный ответ. Дополнить фразу, сопоставить, установить последовательность

Вопросы для самостоятельной подготовки к зачету с оценкой

1. Основы структурного и объектного подходов к программированию. Преимущества и недостатки объектного подхода.

2. Уровни доступа к членам класса.

3. Конструктор с параметрами. Пример программы.

4. Основные принципы ООП: абстракция и инкапсуляция.

5. Объявление объектов. Создание массивов объектов. Пример программы.

6. Конструктор со списком инициализаторов. Пример программы.

7. Понятие класса и объекта в ООП. Состояние и поведение объекта.

8. Указатели на объекты класса. Динамические объекты и массивы объектов. Пример программы.

9. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы.

10. Синтаксис и семантика конструктора. Указатель this. Пример программы.

11. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы.

12. Суть наследования и его виды. Полиморфизм.

13. Свойства конструкторов.

14. Синтаксис и семантика деструктора. Пример программы.

15. Структура класса C++. Пример программы.

16. Конструктор по умолчанию. Пример программы.

17. Свойства деструкторов.

18. Статические переменные и статические функции класса (static). Назначение. Пример программы.

19. Методы класса, дружественные другому классу. Пример программы.

20. Виртуальные базовые классы. Пример программы.

21. Константные переменные и функции класса (const). Пример программы.

22. Дружественные классы. Пример программы.

23. Замещение членов базового класса. Пример программы.

24. Функции, дружественные одному классу. Пример программы.

25. Объявление классов при простом наследовании. Спецификаторы наследуемого доступа. Пример программы.

26. Назначение перегрузки функций. Пример программы.

27. Функции, дружественные нескольким классам. Пример программы.

28. Конструкторы и деструкторы производных классов (очередность вызова). Пример программы.

29. Перегрузка конструкторов. Пример программы.

30. Назначение перегрузки операторов. Правила перегрузки операторов.

31. Механизмы раннего и позднего связывания.

32. Синтаксис шаблонного класса. Пример программы.

33. Ассоциативные контейнеры. Пример.

34. Перегрузка бинарных операторов. Пример программы.

35. Виртуальные функции. Таблица виртуальных функций. Пример программы.

36. Специализация шаблонов функций и классов. Пример программы.

37. Итераторы. Обратные итераторы. Пример.

38. Перегрузка унарных операторов. Пример программы.

39. Чисто виртуальные функции и абстрактные классы. Пример программы.
40. Достоинства и недостатки шаблонов.
41. Алгоритмы библиотеки STL. Пример.
42. Перегрузка оператора доступа к членам класса. Пример программы.
43. Назначение виртуальных деструкторов. Пример программы.
44. Назначение и состав библиотеки STL.
45. Перегрузка и неоднозначность. Пример программы.
46. Назначение шаблонных функций.
47. Последовательные контейнеры. Пример.
48. Виды и реализации полиморфизма.
49. Конкретизация шаблонной функции. Пример программы.
50. Контейнеры – адаптеры. Пример.
51. Иерархия классов VCL-библиотеки.
52. Объекты, компоненты, элементы управления VCL-библиотеки.
53. Классы командных кнопок, флажков и радиокнопок VCL-библиотеки.
54. Классы текстовых компонентов, списков и панелей VCL-библиотеки.
55. Классы стандартных диалогов VCL-библиотеки.
56. Обработка событий средствами VCL-библиотеки.
57. Структура приложения на C# и Java.
58. Создание классов и объектов C# и Java.
59. Методы-члены класса и виртуальные методы C# и Java.
60. Механизмы наследования C# и Java.
61. Классы библиотеки .NET Framework.
62. Приложения – диалоги библиотеки классов .NET Framework.

6. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ

Комплект оценочных средств хранится на кафедре, подлежит обновлению по мере необходимости. Для промежуточной аттестации в виде экзамена каждое ОС по дисциплине обновляется и утверждается за 14 дней до начала сессионного периода и хранится в недоступном месте от несанкционированного доступа. Ответственность несет кафедра.

Порядок проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по ОПОП регламентируются Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся по программам высшего образования.

Текущий контроль успеваемости является формой контроля качества знаний обучающихся, осуществляемого в межсессионный период обучения с целью определения качества освоения ОПОП.

Текущий контроль успеваемости осуществляется: на лекциях, практических (семинарских) занятиях, в рамках контроля самостоятельной работы.

Обучающиеся заранее информируются о критериях и процедуре текущего контроля успеваемости преподавателями по соответствующей учебной дисциплине (модуля).

Успеваемость при текущем контроле характеризует объем и качество выполненной обучающимся работы по дисциплине (модулю).

Педагогические виды и формы, используемые в процессе текущего контроля успеваемости обучающихся, определяются методической комиссией кафедры. Выбираемый вид текущего контроля обеспечивает наиболее полный и объективный контроль (измерение и фиксирование) уровня освоения результатов обучения по дисциплине.

Преподаватели предоставляют сведения о текущей успеваемости обучающихся в рамках проведения текущей аттестации в семестре в деканаты/ учебный отдел института в сроки, определенные внутренними распорядительными документами института.

В целях обеспечения текущего контроля успеваемости преподаватель проводит консультации.

Преподаватель, ведущий занятия семинарского типа, проводит аттестацию обучающихся за прошедший период. Аттестация проводится, если проведено не менее 3 практических (семинарских) или лабораторных занятий, в установленные деканатом сроки, не реже 1 раза за

учебный семестр. Обучающиеся аттестуются путем выставления в соответствующую групповую ведомость записей по системе: «аттестован» или «не аттестован».

Преподаватель, проставляя итоги текущей аттестации, доводит результаты аттестации до сведения студенческой группы и объясняет причины отрицательной аттестации по запросу обучающегося.

При аттестации обучающихся учитываются следующие факторы:

- результаты работы на занятиях, показанные при этом знания по дисциплине (модулю), усвоение навыков практического применения теоретических знаний, степень активности на практических (семинарских) занятиях;

- результаты и активность участия в семинарах и коллоквиумах;
- результаты выполнения контрольных работ;
- результаты и объем выполненных заданий в рамках самостоятельной работы обучающихся;

- результаты личных бесед со студентами по материалу учебной дисциплины (модуля);
- посещение студентами, семинарских и практических занятий, лабораторных работ;
- своевременная ликвидация задолженностей по пройденному материалу, возникших вследствие пропуска занятий либо неудовлетворительных оценок по результатам работы на занятиях.

- результаты прохождения контрольных точек по дисциплине.

Промежуточная аттестация обучающихся института является формой контроля результатов обучения по дисциплине с целью комплексного определения соответствия уровня и качества знаний, умений и навыков обучающихся требованиям, установленным образовательной программой.

Формирование оценки текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины осуществляется с использованием пятибалльной системы оценки знаний обучающихся.

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций. Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 5 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации. Промежуточная аттестация проводится в форме экзамена. Экзамен проводится по расписанию экзаменационной сессии в письменном виде. Количество вопросов в экзаменационном задании – 30. Проверка ответов и объявление результатов производится в день экзамена. Результаты аттестации заносятся в экзаменационную ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.

7. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Адаптированные оценочные материалы содержатся в адаптированной ОПОП. Обучение обучающихся с ограниченными возможностями здоровья при необходимости осуществляется на основе адаптированной рабочей программы с использованием специальных методов обучения и дидактических материалов, составленных с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся (обучающегося).

Самостоятельная работа обучающихся с ограниченными возможностями здоровья и инвалидов позволяет своевременно выявить затруднения и отставание и внести коррективы в учебную деятельность. Конкретные формы и виды самостоятельной работы обучающихся лиц с ограниченными возможностями здоровья и инвалидов устанавливаются преподавателем. Выбор форм и видов самостоятельной работы, обучающихся с ограниченными возможностями здоровья и инвалидов осуществляется с учетом их способностей, особенностей восприятия и

готовности к освоению учебного материала. Формы самостоятельной работы устанавливаются с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге или на компьютере, в форме тестирования, электронных тренажеров и т.п.).

Основные формы представления оценочных средств – в печатной форме или в форме электронного документа. Для обучающихся с нарушениями зрения предусматривается возможность проведения текущего и промежуточного контроля в устной форме. Для обучающихся с нарушениями слуха предусматривается возможность проведения текущего и промежуточного контроля в письменной форме.

Таблица 7.1. Категории обучающихся с ОВЗ, способы восприятия ими информации и методы их обучения.

Категории обучающихся по нозологиям		Методы обучения
с нарушениями и зрения	Слепые. Способ восприятия информации: осязательно-слуховой	Аудиально-кинестетические, предусматривающие поступление учебной информации посредством слуха и осязания. Могут использоваться при условии, что визуальная информация будет адаптирована для лиц с нарушениями зрения: визуально-кинестетические, предполагающие передачу и восприятие учебной информации при помощи зрения и осязания; аудио-визуальные, основанные на представлении учебной информации, при которых задействовано зрительное и слуховое восприятие; аудио-визуально-кинестетические, базирующиеся на представлении информации, которая поступает по зрительному, слуховому и осязательному каналам восприятие.
	Слабовидящие. Способ восприятия информации: зрительно-осязательно-слуховой	
С нарушениями и слуха	Глухие. Способ восприятия информации: зрительно-осязательный	визуально-кинестетические, предполагающие передачу и восприятие учебной информации при помощи зрения и осязания. Могут использоваться при условии, что аудиальная информация будет адаптирована для лиц с нарушениями слуха: аудио-визуальные, основанные на представлении учебной информации, при которых задействовано зрительное и слуховое восприятие; аудио-кинестетические, предусматривающие поступление учебной информации посредством слуха и осязания; аудио-визуально-кинестетические, базирующиеся на представлении информации, которая поступает по зрительному, слуховому и осязательному каналам восприятие.
	Слабослышащие. Способ восприятия информации: Зрительно-осязательно-слуховой	
С нарушениями и опорно-двигательного аппарата	Способ восприятия информации: зрительно-осязательно-слуховой	– визуально-кинестетические; – аудио-визуальные; – аудио-кинестетические; – аудио-визуально-кинестетические.

Таблица 7.2. – Способы адаптации образовательных ресурсов.

Условные обозначения:

«+» —образовательный ресурс, не требующий адаптации;

«АФ» — адаптированный формат к особенностям приема-передачи информации обучающихся инвалидов и лиц с ОВЗ формат образовательного ресурса, в том числе с использованием специальных технических средств;

«АЭ»— альтернативный эквивалент используемого ресурса

Категории обучающихся по нозологиям		Образовательные ресурсы				
		Электронные				Печатные
		мультимедиа	графические	аудио	текстовые, электронные аналоги печатных изданий	
С нарушениями зрения	Слепые	АФ	АЭ (например, создание материальной модели графического объекта (3Dмодели)	+	АЭ (например, аудио описание)	АЭ (например, печатный материал, выполненный рельефно-точечным шрифтом Л. Брайля)
	Слабовидящие	АФ	АФ	+	АФ	АФ
С нарушениями слуха	Глухие	АФ	+	АЭ (например, текстовое описание, гиперссылки)	+	+
	Слабослышащие	АФ	+	АФ	+	+
С нарушениями опорно-двигательного аппарата		+	+	+	+	+

Таблица 7.3. - Формы контроля и оценки результатов обучения инвалидов и лиц с ОВЗ

Категории обучающихся по нозологиям	Форма контроля и оценки результатов обучения
С нарушениями зрения	– устная проверка: дискуссии, тренинги, круглые столы, собеседования, устные коллоквиумы и др.; – с использованием компьютера и специального ПО: работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, дистанционные формы, если позволяет острота зрения - графические работы и др.

Категории обучающихся по нозологиям	Форма контроля и оценки результатов обучения
С нарушениями слуха	– письменная проверка: контрольные, графические работы, тестирование, домашние задания, эссе, письменные коллоквиумы, отчеты и др.; – с использованием компьютера и специального ПО: работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, графические работы, дистанционные формы и др.
С нарушениями опорно-двигательного аппарата	– письменная проверка, с использованием специальных технических средств (альтернативных средства ввода, управления компьютером и др.): контрольные, графические работы, тестирование, домашние задания, эссе, письменные коллоквиумы, отчеты и др.; – устная проверка, с использованием специальных технических средств (средств коммуникаций): дискуссии, тренинги, круглые столы, собеседования, устные коллоквиумы и др.; – с использованием компьютера и специального ПО (альтернативных средств ввода и управления компьютером и др.): работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, Графические работы, дистанционные формы - предпочтительнее обучающимся, ограниченным в передвижении и др.

7.1. ЗАДАНИЯ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Текущий контроль и промежуточная аттестация обучающихся инвалидов и лиц с ОВЗ осуществляется с использованием оценочных средств, адаптированных к ограничениям их здоровья и восприятия информации, в том числе с использованием специальных технических средств.

Текущий контроль успеваемости для обучающихся инвалидов и лиц с ОВЗ направлен на своевременное выявление затруднений и отставания в обучении и внесения коррективов в учебную деятельность. Возможно осуществление входного контроля для определения его способностей, особенностей восприятия и готовности к освоению учебного материала.

7.2. ЗАДАНИЯ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Форма промежуточной аттестации устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При необходимости обучающимся предоставляется дополнительное время для подготовки ответа. Промежуточная аттестация, при необходимости, может проводиться в несколько этапов. Для этого рекомендуется использовать рубежный контроль, который является контрольной точкой по завершению изучения раздела или темы дисциплины, междисциплинарного курса, практик и ее разделов с целью оценивания уровня освоения программного материала. Формы и срок проведения рубежного контроля определяются