

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ефанов Алексей Валерьевич

Должность: Директор Невномысского технологического института (филиал) СКФУ

Дата подписания: 12.10.2022 09:19:08

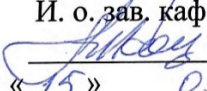
Уникальный программный ключ:

49214306dd433e7a1b0f8632f645f9853e99e5d0

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И. о. зав. кафедрой ИСЭиА

 Колдаев А.И.

«15» 03 2021 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля и промежуточной аттестации
по дисциплине «**Объектно-ориентированное программирование**»

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

Направление подготовки:	09.03.02 Информационные системы и технологии
Направленность (профиль):	Информационные системы и технологии в бизнесе
Квалификация выпускника:	Бакалавр
Форма обучения:	Заочная
Учебный план:	2021 г.
Изучается	на 3 курсе

Предисловие

1. Назначение: данный фонд оценочных средств предназначен для оценивания уровня сформированности компетенций при проведении текущего контроля успеваемости и промежуточной аттестации студентов, обучающихся по направлению подготовки 09.03.02 «Информационные системы и технологии», направленность (профиль) «Информационные системы и технологии в бизнесе».

2. Фонд оценочных средств текущего контроля успеваемости и промежуточной аттестации разработан на основе рабочей программы дисциплины «Объектно-ориентированное программирование» в соответствии с образовательной программой по направлению подготовки 09.03.02 «Информационные системы и технологии», направленность (профиль) «Информационные системы и технологии в бизнесе», утвержденной на заседании Учёного совета НТИ (филиал) СКФУ.

3. Разработчик: доцент кафедры Информационных систем, электропривода и автоматики Д.В. Болдырев,

4. ФОС рассмотрен и утвержден на заседании кафедры Информационных систем, электропривода и автоматики.

5. ФОС согласован с выпускающей кафедрой Информационных систем, электропривода и автоматики.

6. Проведена экспертиза ФОС. Члены экспертной группы, проводившие внутреннюю экспертизу:

Председатель:

Кузьменко В.В. — и.о. директора НТИ (филиал) СКФУ, профессор кафедры гуманитарных и математических дисциплин

Члены экспертной группы

М.В. Должикова — заместитель директора по учебно-воспитательной работе НТИ (филиал) СКФУ;

А.И. Колдаев — доцент кафедры ИСЭиА

Эксперт, проводивший внешнюю экспертизу:

Н.А. Остапенко — кандидат технических наук, ведущий инженер-конструктор КБ модернизации ООО КИЭП «Энергомера» филиал АО «Электротехнические заводы «Энергомера»

7. Экспертное заключение: фонд оценочных средств отвечает основным требованиям федерального государственного образовательного стандарта и способствует формированию требуемых компетенций.

8. Срок действия ФОС: на срок реализации образовательной программы.

Паспорт фонда оценочных средств
 для проведения текущего контроля и промежуточной аттестации
 по дисциплине «Объектно-ориентированное программирование»

Направление подготовки: 09.03.02 Информационные системы и технологии
 Направленность (профиль): Информационные системы и технологии в бизнесе
 Квалификация выпускника: Бакалавр
 Форма обучения: Заочная
 Учебный план: 2021 г.
 Изучается на 3 курсе

Код оцениваемой компетенции	Этап формирования компетенции (№ темы в соответствии с рабочей программой)	Средства и технологии оценки	Вид контроля, аттестации	Тип контроля	Наименование оценочного средства	Количество заданий для каждого уровня	
						Пороговый	Повышенный
ПК-2	Темы: 1-2	Собеседование	Текущий	Устный	Вопросы для собеседования	29	13
ПК-2	Темы: 1-2	Устный экзамен	Промежуточный	Устный	Вопросы к устному экзамену	9	12

Составитель: Д.В. Болдырев

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Вопросы для собеседования
по дисциплине «Объектно-ориентированное программирование»**

Пороговый уровень

Тема 1 Теоретические основы объектно-ориентированного программирования

1. Программы с глобальными и локальными данными.
2. Структурное программирование: основные принципы, пошаговая детализация, процедурная декомпозиция; достоинства и недостатки.
3. Модульное программирование: интерфейс и реализация; достоинства и недостатки.
4. Объектно-ориентированное программирование: объектная декомпозиция; достоинства и недостатки.
5. Абстрагирование.
6. Инкапсуляция (ограничение доступа).
7. Наследование (иерархичность).
8. Полиморфизм (типизация).
9. Анализ задачи.
10. Объектная декомпозиция.
11. Логическое проектирование.
12. Физическое проектирование.
13. Эволюция системы.
14. Модификация проекта.

Тема 2 Техника объектно-ориентированного программирования

1. Статические и экземплярные ресурсы.
2. Поля и методы класса.
3. Конструкторы и деструкторы.
4. Композиция, контейнерные классы.
5. Наследование: родители и потомки.
6. Полиморфизм: раннее и позднее связывание.
7. Полиморфизм на основе абстрактных классов.
8. Полиморфизм на основе виртуальных методов.
9. Перегрузка методов.
10. Перегрузка операций.
11. Делегирование методов, статическое и динамическое делегирование.
12. Параметризованные классы.
13. Интерфейсы.
14. Генерация исключений.
15. Порядок обработки исключений.

Повышенный уровень

Тема 1 Теоретические основы объектно-ориентированного программирования

1. Модульность.
2. Параллелизм.
3. Устойчивость.
4. Структура класса.
5. Статические и динамические ресурсы.

Тема 2 Техника объектно-ориентированного программирования

1. Модификаторы доступа.

2. Свойства и индексы, стратегии доступа.
3. Простое и множественное наследование.
4. Таблица виртуальных методов.
5. Делегирование методов, статическое и динамическое делегирование.
6. Интерфейсы.
7. Обработывающая конструкция `try ... catch`.
8. Финализирующая конструкция `try ... finally`.

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Вопросы к экзамену
по дисциплине «Объектно-ориентированное программирование»**

Пороговый уровень

1. Эволюция технологии разработки программных продуктов
2. Основные принципы объектно-ориентированного программирования
3. Классы и объекты
4. Организация класса. Модификаторы доступа. Статические и экземплярные ресурсы.
5. Поля и методы класса.
6. Конструкторы и деструкторы.
7. Генерация исключений. Порядок обработки исключений.
8. Обработка исключений. Конструкция try ... catch.
9. Финализирующая конструкция try ... finally.

Повышенный уровень

1. Объектная декомпозиция
2. Разработка программ с использованием объектно-ориентированной технологии
3. Композиция. Контейнерные классы.
4. Наследование: родители и потомки.
5. Простое и множественное наследование.
6. Полиморфизм: раннее и позднее связывание.
7. Полиморфизм на основе абстрактных классов.
8. Полиморфизм на основе виртуальных методов.
9. Статическое делегирование методов.
10. Динамическое делегирование методов.
11. Параметризованные классы.
12. Интерфейсы.

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Типовые задания и (или) иные материалы, необходимые
для оценки знаний, умений, навыков и (или) опыта
деятельности обучающихся по дисциплине**

Компетентностно-ориентированные задания и задачи

ЗАДАЧА 1

Цель:

Ознакомление с концепциями инкапсуляции и модульности. Изучение приемов работы с классами, конструкторами и деструкторами, разработка интерфейса методов класса, создание и работа с экземплярами класса. Освоение принципа «класс-элемент — класс-набор».

Задание:

Дан объект согласно вариантам, который является элементом для набора. Элемент состоит из компонент, которые хранятся в нем. Над элементом определены операции:

- получение значения компоненты элемента;
- установка и инициализация значения компоненты элемента;
- контроль значения компоненты элемента (на допустимый диапазон);
- копирование элемента.

Из элементов строится набор. Элементы в наборе проиндексированы от стартового значения. Размер набора задается при создании. Над набором определены операции:

- установка стартового индекса, получение диапазона индексов;
- заполнение набора случайными значениями;
- получение и изменение элемента набора по индексу;
- сортировка элементов по возрастанию и по убыванию;
- дополнительные операции согласно вариантам.

Необходимо разработать:

- класс для описания элемента и его свойств;
- класс для описания набора и его свойств;
- методы работы с элементом и с набором для перечисленных операций;
- дефолтный, копирующий, параметрический конструкторы для создания экземпляров набора и экземпляров элемента;
- интерфейс для редактирования элемента и интерфейс для редактирования набора, отображения и изменения их свойств.

Требования:

Интерфейс и все классы реализуются в одном модуле. Для редактирования элемента разработать функцию `ModifyElement()`, которая должна получать ссылку на экземпляр элемента и предоставлять интерактивный консольный интерфейс для работы с ним. Для редактирования набора разработать функцию `ModifyPalette()`, которая должна получать ссылку на экземпляр набора и предоставлять интерактивный консольный интерфейс для работы с ним. Функция `Main()` запрашивает количество элементов, создает экземпляр набора параметрическим конструктором и вызывает функцию `ModifyPalette()`, которая использует `ModifyElement()`.

Комментарии:

Набор хранит элементы как динамический массив, определяя его размер при создании. При копировании набора, копируются все его элементы.

Варианты заданий:

0. Элемент: цвет в формате CYMK. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.

1. Элемент: положение солнца в координатах α -азимут, z -зенит, θ -горизонт. Дополнительно: сортировка по компонентам и целиком, пересчет в другую систему координат (на выбор).
2. Элемент: цвет в формате YUV. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
3. Элемент: положение солнца в экваториальных координатах θ - склонение, p — полярное расстояние, t — часовой угол. Дополнительно: сортировка по компонентам и целиком, пересчет в другую систему координат (на выбор).
4. Элемент: цвет в формате AHSL. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
5. Элемент: цвет в формате RYB. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
6. Элемент: декартовы координаты в пространстве (x, y) — координаты. Дополнительно: сортировка по компонентам и целиком, пересчет в цилиндрическую систему координат.
7. Элемент: цвет в формате YIQ. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.
8. Элемент: время в формате h — часы, m — минуты, s — секунды. Дополнительно: сортировка по компонентам и целиком, пересчет в 12-ти часовой формат времени (AM и PM).
9. Элемент: цвет в формате HSV. Дополнительно: сортировка по компонентам и целиком, пересчет в RGB.

ЗАДАЧА 2

Цель:

Ознакомление с концепциями полиморфизма и типизации. Изучение перегрузки операторов. Разработка классов с перегруженными операторами и программирование выражений с их помощью.

Задание:

Дан объект данных, над которым определены операции согласно вариантам. Реализовать набор операций для работы с объектом так, чтобы его можно было использовать в выражениях, не прибегая к вызову функций. Необходимо разработать:

- класс объекта и определить правила выполнения операций над ним;
- набор перегруженных операторов, реализующих операции с объектом;
- интерфейс для редактирования объекта с помощью операторов;
- интерфейс для тестирования использования объекта в выражениях.

Требования:

Интерфейс и класс объекта реализуются в одном модуле. Для редактирования объекта разработать функцию `ModifyObject()`, которая должна получать ссылку на экземпляр объекта и предоставлять интерактивный консольный интерфейс для работы с ним. Для тестирования использования объекта в выражениях разработать функцию `Main()`, которая должна предоставлять интерактивный консольный интерфейс, демонстрирующий применение объекта в выражениях.

Обязательной реализации подлежат следующие операции: сложение (+ и +=), вычитание (- и -=), умножение (* и *=), сравнение на равенство (== и !=), унарные (+, -), инверсия (~), присваивание (=), проверка на ноль (!), преобразование к типу (type), ввод из потока (cin >>) и вывод в поток (cout <<). Разработать набор тестовых примеров, демонстрирующих использование перегруженных операторов в арифметических и логических выражениях.

Комментарии:

При перегрузке операторов необходимо учитывать:

- нельзя определить новый лексический символ для оператора;
- нельзя изменить приоритет операторов;
- нельзя изменить арифметичность операторов;
- нельзя перегрузить оператор для стандартных типов данных;
- нельзя перегружать не перегружаемые операторы (`.`), (`::`), (`?_:_`), (`sizeof`), (`typeid`), (`,`);
- некоторые операторы можно перегружать только как методы класса (`=`), (`[]`), (`()`), (`->`);
- если оператор можно использовать и как унарный, и как бинарный, например, (`&`), (`*`), (`+`), (`-`), то каждый способ применения перегружается отдельно;
- перегруженные операторы не могут иметь аргументов по умолчанию;
- перегруженные операторы должны учитывать смысловую эквивалентность: `var = var + 1`;

var += 1; var++; ++var; var -= (-1);

- поведение перегруженных операторов должно соответствовать их смысловому содержанию для определяемых типов данных.

Варианты заданий:

0. Объект: комплексное число (вещественная и мнимая части). Принять: (!) — проверка на ноль, (+ и +=) — сложение, (- и -=) — вычитание, (* и *=) — умножение, (== и !=) — сравнение, (double) — вычисление модуля, (float) — вычисление аргумента, (~) — сопряженное число.
1. Объект: интервал времени (часы, минуты, секунды). Реализовать операции с учетом ограничений на часы (0 до 23), минуты и секунды (0 до 59), т.е. результат всегда от 0:0:0 до 23:59:59. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (* и *=) — удлинение или сокращение, (!) — проверка на ноль, (== и !=) — сравнение, (long) — преобразование в секунды, (float) — преобразование в часы (3600 сек), (~) — дополнение до конца суток.
2. Объект: денежная сумма (признак валюты [p., \$], сумма в номинале [рубли, доллары], сумма в размене [копейки, центы]). Реализовать операции с учетом конвертации, если валюты не совпадают. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (* и *=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (float) — в номинал, (int) — в размен, (~) — изменение признака валюты с конвертацией, (%) — процент от суммы.
3. Объект: интервал даты (часов, дней, лет). Реализовать операции с учетом столетия (0 до 99) и ограничений на дни (0 до 364) и часы (0 до 23), т.е. результат всегда от 0-0-0 до 23-364-99. Принять: (* и *=) — удлинение или сокращение, (+ и +=) — сложение, (- и -=) — вычитание, (== и !=) — сравнение, (!) — проверка на ноль, (long) — преобразование в часы, (float) — преобразование в года (365 дней), (~) — дополнение до конца столетия.
4. Объект: расстояние (сажень, аршин, вершок). 1 сажень = 3 аршинам, 1 аршин = 16 вершкам, 1 вершок = 44,5 мм. Результат всегда от 0 до 500 саженей (1 верста). Принять: (+ и +=) — сложение, (- и -=) — разность, (* и *=) — удлинение или сокращение, (== и !=) — сравнение, (!) — проверка на ноль, (double) — преобразование в миллиметры, (int) — преобразование в вершки, (~) — дополнение до версты (500 саженей).
5. Объект: строка символов (0 до 128). Принять: (+ и +=) — соединение строк, повторение символа, (- и -=) — отсечение строки, (*) — поиск подстроки, (*=) — заполнение подстрокой или символом, (== и !=) — сравнение, (!) — проверка на пусто, (~) — переворот наоборот, (int) — длина строки.
6. Объект: натуральная дробь (целое, числитель, знаменатель). Реализовать операции с учетом приведения к общему знаменателю. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (* и *=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (double) — преобразование в рациональную дробь, (~) — взаимобратная натуральная дробь.
7. Объект: угол (градусы, минуты, секунды). Реализовать операции с учетом целых оборотов и ограничений на градусы (0 до 359), минуты и секунды (0 до 59), т.е. результат всегда от 0°0'0" до 359°59'59". Принять: (+ и +=) — сложение, (- и -=) — вычитание, (* и *=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (double) — преобразование в радианы, (int) — преобразование в секунды, (~) — обратный угол до 360°.
8. Объект: квадратная матрица [3x3]. Реализовать операции над матрицами. Принять: (+ и +=) — сложение, (- и -=) — вычитание, (* и *=) — умножение, (!) — проверка на ноль, (== и !=) — сравнение, (double) — вычисление детерминанта, (int) — количество ячеек, (~) — транспонирование.
9. Объект: алфавит (только прописные от А до Z). Реализовать операции над алфавитами как над множествами. Принять: (+ и +=) — объединение, (!) — проверка на пусто, (- и -=) — разность, (* и *=) — пересечение, (== и !=) — сравнение, (int) — количество букв, (~) — отрицание как замена на буквы, которых нет.

ЗАДАЧА 3

Цель:

Ознакомление с концепциями наследования и абстракции. Обычное и множественное наследование классов. Статические свойства, методы, классы. Перегрузка методов при наследовании классов. Освоение приемов создания, редактирования, копирования, удаления экземпляров и разработки интерфейсов классов.

Задание:

Дана фигура на плоскости согласно вариантам. Фигура описывается индивидуальными геометрическими свойствами и общими оформительскими свойствами: цвет, видимость. У фигуры имеются характеристики: периметр, площадь, ограничивающая область. Область размещения фигур в плоскости ограничена экстендами, за которые фигура не должна выходить.

Необходимо разработать:

- классы для описания положения «Location» и ограничивающей области «Clip» в плоскости;
- статический класс «Geometry» для хранения общих констант и методов проверки различных ограничений на размещение фигур в плоскости;
- класс геометрического примитива «Primitive» для хранения и редактирования оформительских свойств фигуры как наследника от статического класса «Geometry»;
- класс примитивной фигуры — точки «Point» как наследника от классов «Location» и «Primitive»;
- класс фигуры согласно варианту «Figure» как наследника от класса «Point» с описанием специфических свойств и методов фигуры;
- наборы конструкторов для создания экземпляров каждого класса различными способами (дефолтный, копирующий, параметрический);
- методы для изменения свойств и вычисления характеристик фигуры;
- интерфейс для отображения и изменения всех свойств фигуры.

Требования:

Интерфейс и классы реализуются в одном модуле. Для редактирования фигуры разработать функцию ModifyFigure(), которая должна получать ссылку на экземпляр фигуры и предоставлять интерактивный консольный интерфейс для работы с ним.

Варианты заданий:

0. Фигура: сектор окружности.
1. Фигура: треугольник Рело.
2. Фигура: правильный шестиугольник.
3. Фигура: эллипс.
4. Фигура: параллелограмм.
5. Фигура: сегмент окружности.
6. Фигура: ромб.
7. Фигура: кольцо (бублик).
8. Фигура: правильная трапеция.
9. Фигура: дельтоид.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Тестовые задания

1. При разработке ПО в первую очередь следует заботиться о...
 - +Корректности
 - Функциональности
 - Простоте использования
 - Интерфейсе пользователя

2. Что относится к принципам объектно-ориентированного программирования?
 - +Абстрагирование
 - Абстракция
 - +Инкапсуляция
 - Замыкание
 - +Наследование
 - Передача
 - +Полиморфизм
 - Мультиморфизм

3. Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение, — это...
 - +Объектно-ориентированное программирование
 - Абстрагирование
 - Инкапсуляция
 - Наследование
 - Полиморфизм

4. При создании объектно-ориентированной программы предметная область представляется в виде совокупности _____ (ответ записать одним словом с маленькой буквы в родительном падеже множественного числа).
 - +объектов

5. Под объектами понимают:
 - Всю абстрактную сущность, заданную набором имен атрибутов и имен методов поведения
 - +Некоторую абстрактную сущность, заданную набором имен атрибутов и имен методов поведения
 - Некоторую видимую сущность, заданную набором имен атрибутов и имен методов поведения

6. _____ — это описание множества объектов программирования и выполняемых над ними действий (ответ записать одним словом с маленькой буквы в именительном падеже единственного числа).
 - +класс

7. Классу соответствует тип:
 - Объективный
 - +Объектный тип
 - Видимый
 - Визуальный

8. Класс — это...
 - +Статическая структура

- +Программный текст
- Динамическая структура, создаваемая в момент выполнения
- +Абстрактный тип данных с заданной реализацией (возможно частичной)

9. Из чего состоит класс?

- Объект
- +Метод
- +Данные
- +Свойства

10. Характеристика объекта, определяющая его состояние — это _____ (ответ записать одним словом с маленькой буквы в именительном падеже единственного числа).

- +свойство

11. Метод класса — это...

- +Компонент класса
- +Процедура или функция
- Поле класса
- Часть структуры данных, представляющей объект

12. Изменение состояния объекта в ответ на какое-либо действие — это _____ (ответ записать одним словом с маленькой буквы в именительном падеже единственного числа).

- +событие

13. Действие, которое может выполнить объект, называется _____ (ответ записать одним словом с маленькой буквы в творительном падеже единственного числа).

- +методом

14. Скрытие деталей реализации объекта — это _____ (ответ записать одним словом с маленькой буквы в именительном падеже единственного числа).

- +инкапсуляция

15. Возможность объектов с одинаковой спецификацией иметь различную реализацию — это _____ (ответ записать одним словом с маленькой буквы в именительном падеже единственного числа).

- +полиморфизм

16. Наследование — это...

- Возможность объектов с одинаковой спецификацией иметь различную реализацию
- +Возможность при описании класса указывать на его происхождение от другого класса
- Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую
- Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое

17. Для вызова статического метода необходимо обратиться к _____ (ответ записать одним словом с маленькой буквы в дательном падеже единственного числа).

- +классу

18. Для вызова экземплярного метода необходимо создать _____ (ответ записать одним словом с маленькой буквы в именительном падеже единственного числа).

- +объект

19. Какая функция не может быть конструктором?

- String (const int a)
- String ()

+void String ()

20. Деструктор класса...

- Принимает в качестве параметра адрес того объекта, который нужно уничтожить
- +Не содержит параметров

21. Отметьте правильное утверждение для абстрактного класса:

- Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам
- +Невозможно создать объект абстрактного класса
- В абстрактном классе вообще не описываются методы

22. Принцип полиморфизма реализуется...

- С помощью множественного наследования
- +С помощью виртуальных методов
- С помощью абстрактных классов

23. Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренним компонентам?

- Шаблонная
- Полиморфная
- +Дружеская
- Статическая

24. Класс Exception...

- +Позволяет классифицировать исключения
- +Позволяет создавать собственные исключения
- +Позволяет организовать разбор случаев при обработке исключения
- Не может иметь наследников

25. Обработка исключений...

- +Обеспечивает устойчивость ПО
- Предназначена для обработки специальных случаев, предусмотренных спецификацией
- Позволяет отключать некоторые модули приложения
- +Является механизмом восстановления в аварийных ситуациях

26. К внутренним факторам, влияющим на качество ПО относятся...

- Расширяемость
- Совместимость
- +Модульность
- +Объектная ориентированность
- Переносимость

27. Корректность программы — это понятие...

-
- Неформальное
- Которое можно определить, используя только термины самой программы
- +Которое можно формализовать триадой Хоара
- +Для формализации которого необходимо задание спецификации

28. Наследник...

- Наследует все компоненты родителя
- +Может определить собственные (непосредственные) компоненты
- Может переопределить атрибуты родителя
- +Может переопределить методы родителя

29. Статическая типизация...

- +Позволяет обнаруживать многие ошибки еще на этапе компиляции
- Возможна только для ОО-языков
- Позволяет установить динамический тип сущности
- Анализирует состояние объектов в период выполнения

30. Функциональная декомпозиция имеет следующие достоинства:

- +Система строится на основе хорошо понятных последовательных уточнений
- +Уровень абстракции на каждом шаге уточнения уменьшается
- +Появляется возможность справиться со сложностью исходной задачи
- Появляется возможность выделить главную функцию системы

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Критерии и шкалы оценивания для проведения
промежуточной аттестации обучающихся**

Повышенный уровень	Базовый уровень	Компетенции не сформированы
<p><i>Критерий 1.</i> Знает и понимает термины, определения, основные закономерности, может самостоятельно их интерпретировать и использовать; дает полный, развернутый ответ</p>	<p><i>Критерий 1.</i> Знает термины и определения, но допускает неточности; знает основные закономерности, способен их интерпретировать, но не способен использовать; дает часть ответа на вопрос.</p>	<p><i>Критерий 1.</i> Не знает термины и определения, основные закономерности, не способен их интерпретировать и использовать; ответ не дает.</p>
<p><i>Критерий 2.</i> Самостоятельно анализирует теоретический материал, умеет применять теоретическую базу при выполнении практических заданий; выполняет задания повышенной сложности, предлагает собственный метод решения, грамотно обосновывает его ход; самостоятельно анализирует решение и делает выводы.</p>	<p><i>Критерий 2.</i> Умеет выполнять практические задания, но не всех типов; способен решать задачи по заданному алгоритму; испытывает затруднения при анализе теоретического материала в применении теории при решении задач и обосновании решения; допускает ошибки при выполнении заданий, нарушение логики решения; испытывает затруднения с выводами.</p>	<p><i>Критерий 2.</i> Не умеет выполнять поставленные практические задания, выбирать типовой алгоритм решения; не может установить взаимосвязь теории с практикой, не способен ответить на простые вопросы по выполнению заданий, не может проанализировать теоретический материал и обосновать выбор метода решения задач; не делает выводы.</p>
<p><i>Критерий 3.</i> Владеет методикой решения стандартных задач и заданий, использует полученные навыки при решении нестандартных задач; выполняет трудовые действия быстро, качественно, самостоятельно без посторонней помощи, производит оценку.</p>	<p><i>Критерий 3.</i> Не владеет методикой решения стандартных задач и заданий, испытывает трудности при выполнении поставленных задач; выполняет трудовые действия медленно, с недостаточным качеством; оценивает факты и собственные трудовые действия только с помощью наставника.</p>	<p><i>Критерий 3.</i> Не обладает навыками выполнения поставленных задач; не способен выполнять трудовые действия или выполняет очень медленно, некачественно, не достигая поставленных задач; не видит различий между фактами и оценочными суждениями; не может самостоятельно планировать и выполнять собственные трудовые действия, не способен к рефлексии</p>

Критерии оценки ответов при собеседовании:

«5» (отлично): студент демонстрирует системные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью и способность быстро реагировать на уточняющие вопросы.

«4» (хорошо): студент демонстрирует прочные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью, но при этом делает несущественные ошибки, которые быстро исправляет самостоятельно или при незначительной коррекции преподавателем.

«3» (удовлетворительно): студент демонстрирует неглубокие теоретические знания, проявляет слабо сформированные навыки анализа явлений и процессов, недостаточное умение делать аргументированные выводы и приводить примеры, показывает недостаточно свободное владение монологической речью, терминологией, логичностью и последовательностью изложения, делает ошибки, которые может исправить только при коррекции преподавателем.

«2» (неудовлетворительно): студент демонстрирует незнание теоретических основ предмета, не умеет делать аргументированные выводы и приводить примеры, показывает слабое владение монологической речью, не владеет терминологией, проявляет отсутствие логичности и последовательностью изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на занятии.

Критерии оценки практических работ

«5» (отлично): выполнены все задания практической работы, студент четко и без ошибок ответил на все контрольные вопросы.

«4» (хорошо): выполнены все задания практической работы; студент ответил на все контрольные вопросы с замечаниями.

«3» (удовлетворительно): выполнены все задания практической работы с замечаниями; студент ответил на все контрольные вопросы с замечаниями.

«2» (неудовлетворительно): студент не выполнил или выполнил неправильно задания практической работы; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.