Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Выборнова Любовь Алексеевна
Должность: Ректор
Дата подписания: 30.10.2026 08:25:20
Уникальный программный ключ:
c3b3b9c625f6c113afa2a2c42baff9e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования«Поволжский государственный университет сервиса» (ФГБОУ ВО «ПВГУС»)

Колледж креативных индустрий и предпринимательства

УТВЕРЖДЕНО
на заседании ПЦК

по образовательной программе 09.02.07 Информационные системы и программирование

Протокол от 24.10.2025 г. № 2

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
(фонд оценочных средств)для проведения промежуточной аттестации

по междисциплинарному курсу

учебному предмету, дисциплине, междисциплинарному курсу, профессиональному модулю

«Инструментальные средства разработки программного обеспечения»

наименование учебного предмета, дисциплины, междисциплинарного курса, профессионального модуля

по образовательной программе среднего профессионального образования –

программе подготовки специалистов среднего звена

программе подготовки специалистов среднего звена / программе подготовки квалифицированных рабочих, служащих

Информационные системы и программирование

наименование образовательной программы

специальность 09.02.07 Информационные системы и программирование

шифр, наименование специальности / профессии

Составитель Ефимов Д.М., преподаватель Колледжа креативных индустрий и предпринимательства

ФИО, должность, структурное подразделение, ученая степень, ученое звание

Тольятти
2025

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ (ДАЛЕЕ – ФОС)

1.1. Планируемые результаты обучения по дисциплине

Код компетенции	Наименование компетенции
ПК 2.1	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств

В результате освоения междисциплинарного курса обучающийся должен:

иметь практический опыт:

-выполнение анализа проектной и технической документации для определения взаимодействия компонентов программной системы;

- разработка требований к программным модулям на основе изученной документации;

- участие в согласовании требований с другими участниками проекта.

-выполнение сборки программного обеспечения из отдельных модулей;

- проверка совместимости программных модулей между собой при интеграции;

-участие в тестировании работоспособности программного обеспечения после интеграции;

-выполнение отладки программного модуля с использованием программных средств разработки;

- поиск и устранение ошибок в программном коде на этапе отладки;

- анализ результатов работы программного модуля для выявления сбоев и некорректной логики;

- интеграции модулей в программное обеспечение.

уметь:

-определять требования к программным модулям на основе анализа проектной и технической документации;

- применять основные методы анализа структуры программных систем при разработке требований;

- оформлять требования в соответствии с установленными правилами;

-выполнять интеграцию программных модулей с учётом логики их взаимодействия;

-использовать стандартные инструменты и средства разработки для сборки программных решений;

- выявлять и устранять ошибки, возникающие при интеграции программных компонентов;

-использовать отладочные инструменты для пошагового анализа и тестирования программного модуля.

-распознавать типовые ошибки в программном коде и устранять их;

-оценивать корректность работы программного модуля по результатам его

выполнения;

- использовать выбранную систему контроля версий;
- использовать методы для получения кода с заданной функциональностью и степенью качества.

знать:

-виды и назначение проектной и технической документации;

-принципы взаимодействия между программными модулями в составе цифровой системы;

-основные подходы и правила описания требований к программному обеспечению;

-основные принципы и этапы интеграции программных модулей;

-способы организации взаимодействия между программными компонентами, средства и инструменты, используемые для сборки, отладки и тестирования программного обеспечения.

-основные методы и приёмы отладки программного обеспечения;
- возможности стандартных отладчиков и инструментов разработки;
- типовые ошибки программирования и способы их устранения;
- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения.

1.2. Содержание учебной дисциплины

№	Тема (раздел дисциплины) (в соответствии с РПД)	Код компетенции
1	Программное обеспечение. Виды ПО. Инструментальное ПО. Язык программирования Java. Области применения. Основные типы данных. Транспортные протоколы.	ПК 2.1, ПК 2.2, ПК 2.3
2	Примитивные и ссылочные типы данных. Классы-обертки. Среды разработки. Системы контроля версий.	ПК 2.1, ПК 2.2, ПК 2.3
3	Общая характеристика ИС. Кроссплатформенность.	ПК 2.1, ПК 2.2, ПК 2.3

1.3. Система оценивания по учебной дисциплине

Курс изучается в течение одного семестра. Форма промежуточной аттестации по дисциплине – контрольная работа.

Шкалы оценки уровня сформированности результатов обучения		Шкала оценки уровня освоения дисциплины	
Уровневая шкала оценки компетенций	100 бальная шкала, %	100 бальная шкала, %	недифференцированная оценка
допороговый	ниже 61	ниже 61	не зачтено
пороговый	61-85,9	61-69,9	зачтено
		70-85,9	
повышенный	86-100	86-100	

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

В ходе проведения промежуточной аттестации осуществляется контроль и оценка результатов освоения профессиональных компетенций и их частей.

Вопросы для подготовки к контрольной работе

№	Содержание вопроса
ПК 2.1Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент	
1.	Что такое техническое задание (ТЗ), и какова его роль в разработке программных модулей?
2.	Что такое информационная система (ИС)?
3.	Области применения языка программирования Java.
4.	Для чего используются комментарии в Java? Как добавить их в программный код?
5.	Назовите основные этапы анализа проектной документации.
6.	Что подразумевается под взаимодействием компонентов в программном обеспечении? Приведите пример.
7.	Какие основные типы требований могут быть сформулированы для программного модуля?
8.	Объясните, что такое интерфейс программного модуля и его значение для взаимодействия компонентов.
9.	В чем разница между функциональными и нефункциональными требованиями? Приведите по одному примеру каждого.
10.	Как можно описать взаимодействие модулей с использованием диаграмм UML?
11.	Какие аспекты взаимодействия компонентов влияют на надежность системы?
12.	Что такое нефункциональные требования, связанные с производительностью, и как они могут быть проверены?
13.	Назовите основные транспортные протоколы. Как можно работать с ними на уровне языка программирования Java?
14.	Почему важно документировать ограничения и допущения при разработке программного модуля?
15.	Какие виды диаграмм UML помогают отобразить процесс взаимодействия компонентов? Приведите примеры.
16.	Как использование стандартов (например, IEEE) помогает улучшить качество требований к программным модулям?
17.	Что такое согласованность требований, и почему она важна при проектировании модулей?
18.	Как изменение требований может повлиять на взаимодействие между модулями?
19.	Какие методы используются для выявления конфликтов между требованиями различных модулей?
ПК 2.2 Выполнять интеграцию модулей в программное обеспечение	
20.	Что такое интеграция программных модулей, и зачем она нужна?
21.	Что такое программно-аппаратные средства ИС?
22.	Виды программного обеспечения. Какие функции выполняет каждый из них?
23.	Какие существуют основные подходы к интеграции модулей в программное обеспечение?
24.	В чем разница между последовательной интеграцией и интеграцией "сверху вниз"?
25.	Что такое инструментальные средства ИС?
26.	Что такое интеграционное тестирование, и какова его роль в процессе интеграции модулей?
27.	Объясните понятие "точка интеграции" и её значение для успешного объединения модулей.
28.	Какие сложности могут возникнуть при интеграции модулей, разработанных разными командами?
29.	Как подключить библиотеку в программный код, написанный на ЯП Java?
30.	Что такое API, и как он используется для интеграции программных компонентов?

№	Содержание вопроса
31.	Какую роль играют стандарты и соглашения при интеграции модулей? Приведите примеры таких стандартов.
32.	Что такое stub (заглушка) и mock (имитация)? В чем их различия, и как они помогают в интеграционном тестировании?
33.	Какие инструменты и технологии используются для автоматизации процесса интеграции? Приведите примеры.
34.	Как можно выявить и устранить конфликты, возникающие при объединении модулей?
35.	Рекурсия на примере использования функций в Java.
36.	В чем преимущества использования систем управления версиями (например, Git) при интеграции модулей?
37.	Объясните концепцию "непрерывной интеграции" (ContinuousIntegration, CI) и её значение в современном ПО.
38.	Почему важно учитывать зависимости между модулями перед их интеграцией? Как можно минимизировать риски?
39.	Какие метрики качества интеграции можно использовать для оценки успешности процесса?
40.	Назовите основные типы данных в Java.
ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств	
41.	Что такое отладка программного модуля, и какие задачи она решает?
42.	Какие существуют основные этапы процесса отладки программного модуля?
43.	Объясните разницу между синтаксическими, логическими и исполнительными ошибками в программе.
44.	Что такое системы контроля версий? Какие системы вы знаете?
45.	Что такое точка останова (breakpoint), и как она используется в процессе отладки?
46.	Какие данные можно получить при пошаговом выполнении программы (step-by-stepdebugging)?
47.	Что такое стек вызовов (callstack), и как его анализ помогает в отладке?
48.	Какие инструменты предоставляют современные IDE для отладки программного модуля? Приведите примеры.
49.	Объясните понятие журналирования (logging). В чем его преимущества и ограничения по сравнению с интерактивной отладкой?
50.	Какие существуют типы точек останова, и в каких ситуациях они применяются?
51.	Как можно использовать переменные среды и параметры запуска для отладки программного модуля?
52.	Что такое кроссплатформенность? Принцип WORA в Java.
53.	Что такое "плавающая ошибка" (Heisenbug), и почему её трудно обнаружить?
54.	Как определить и устранить утечки памяти в программном модуле? Какие инструменты могут помочь?
55.	Какие методы используются для отладки многопоточных приложений? Какие сложности могут возникнуть?
56.	Опишите действия при отладке кода на ЯП Java в среде IntellijIdea.
57.	Что такое профилирование кода, и как оно помогает в процессе отладки?
58.	Почему важно тестировать исправления после отладки, и какие виды тестирования для этого подходят?