

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ПРАКТИКИ**
по профессиональному модулю
ПМ.01 Осуществление интеграции программных модулей

программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование

Форма обучения: *очная*

Находка 2026

Рабочая программа учебной (по профилю специальности) практики по профессиональному модулю *ПМ.01 Осуществление интеграции программных модулей* разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности / профессии *09.02.07 Информационные системы и программирование*, утвержденного приказом Минобрнауки России от «09» декабря 2016 г. № 1547, примерной образовательной программой.

Разработчик(и): *Лихацкий М.Б. преподаватель ОСПО филиала ФГБОУ ВО «ВВГУ» в г. Находке*

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № 10 от «29» июня 2026 г.

Председатель ЦМК _____ *В.В Плахотина*

подпись

СОДЕРЖАНИЕ

1. Общая характеристика программы практики	5
2. Результаты практики	7
3. Структура и содержание практики.....	7
4. Условия реализации программы практики	11
5. Контроль и оценка результатов освоения практики	14

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ ПРАКТИКИ

1.1. Область применения программы практики

Программа практики является составной частью Программы подготовки специалистов среднего звена, обеспечивающей реализацию ФГОС СПО. Практика является частью учебного процесса и направлена на формирование у студентов практических профессиональных умений, приобретение первоначального практического опыта по основным видам профессиональной деятельности для последующего освоения ими общих и профессиональных компетенций по избранной специальности.

1.2. Цели и задачи практики – требования к результатам освоения практики, формы отчетности

Учебная практика направлена на:

- формирование у студентов практических профессиональных умений, приобретение первоначального практического опыта;
- закрепление теоретических знаний, полученными студентами в процессе обучения профессиональных модулей;
- углубление первоначального профессионального опыта студента, развития общих и профессиональных компетенций, проверку его готовности к самостоятельной трудовой деятельности;
- сбор, систематизация и обобщение практического материала (в том числе) для использования в выпускной квалификационной работе.

По окончании практики студент сдаёт отчет в соответствии с содержанием тематического плана практики и по установленной форме.

Итоговая аттестация проводится в форме - дифференцированного зачёта.

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями студент в ходе прохождения учебной и производственной практики должен: иметь практический опыт:

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения. уметь:
- использовать выбранную систему контроля версий;
- использовать методы для получения кода с заданной функциональностью и степенью качества.

знать:

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения; – основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения.

1.3. Количество часов на освоение программы практики

Рабочая программа практики рассчитана на прохождение студентами практики в объеме **108** час.

2. РЕЗУЛЬТАТЫ ПРАКТИКИ

Общие компетенции:

Код	Наименование общих компетенций
ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках
ОК 11	Планировать предпринимательскую деятельность в профессиональной сфере.

Профессиональные компетенции:

Код	Наименование профессиональных компетенций
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
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

3. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИКИ

3.1 Тематический план практики

Наименование профессионального модуля	Коды формируемых компетенций	Объем времени, отводимый на практику
1	2	3
ПМ.01. Осуществление интеграции программных модулей	ПК 2.1	108 часов
	ПК 2.2	
	ПК 2.3	
	ПК 2.4	
	ПК 2.5	
Экзамен по модулю «ПМ.01. Осуществление интеграции программных модулей»		4 часа

3.2. Содержание практики

код ПК	Виды работ, обеспечивающих формирование ПК	Объем часов
		УП
ПК 2.1	– анализ требований к программному обеспечению; – определение характера взаимодействия компонентов программного обеспечения	20
ПК 2.2	– определение этапов разработки программного обеспечения; – демонстрация построения концептуальной, логической и физической моделей программного обеспечения и отдельных модулей; – выбор технологии разработки исходного модуля исходя из его назначения;	22
ПК 2.3	– выявление ошибок в программных модулях; – определение возможности увеличения быстродействия программного продукта; – определение способов и принципов оптимизации;	22
ПК 2.4	– осуществлять разработку тестовых наборов для программного обеспечения; – осуществлять разработку тестовых сценариев для программного обеспечения;	22
ПК 2.5	– выбор методов обеспечения качества и надежности в процессе разработки сложных программных средств; – изложение основных принципов тестирования; – способен производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.	22

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ПРАКТИКИ

4.1 ТРЕБОВАНИЯ К ДОКУМЕНТАЦИИ, НЕОБХОДИМОЙ ДЛЯ ПРОВЕДЕНИЯ ПРАКТИКИ

Для проведения практики разработана следующая документация:

- положение об учебной практике студентов;
- рабочая программа практики;
- календарно-тематический план;
- приказ о назначении руководителя практики от образовательного учреждения;
- договоры с предприятиями по проведению практики;
- приказ о распределении студентов по базам практики;
- план-график консультаций и контроля за выполнением студентами программы практики (при проведении практики на предприятии);
- график защиты отчетов по практике.

4.2 ТРЕБОВАНИЯ К УЧЕБНО-МЕТОДИЧЕСКОМУ ОБЕСПЕЧЕНИЮ ПРАКТИКИ

В целях реализации требований к учебно-методическому обеспечению практики разработаны и утверждены:

- Задания на практику;
- Методические рекомендации для студентов по выполнению видов работ на практике;
- Методические рекомендации по формированию отчетов по практике;
- Методические рекомендации по оформлению дневника по практике;
- Критерии оценки прохождения практики и защиты отчетов.

4.3 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ОБУЧЕНИЯ.

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы:

Печатные издания не используются. Дисциплина полностью обеспечена электронными изданиями.

Основные источники:

1. Рудаков А. Технология разработки программных продуктов: учебник. Изд. Academia. Среднее профессиональное образование. 2018 г. 208 стр.
<http://www.academia-moscow.ru/catalogue/4831/401005/> ссылка на книгу 2018

Электронные издания (электронные ресурсы)

1. От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам. http://real.tepkom.ru/Real_OM-CM_A.asp

Дополнительные источники

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб. пособие / Л.Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул; Под ред. Л. Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2018. -400 с. <https://znanium.com/bookread2.php?book=768473>

<https://znanium.com/bookread2.php?book=1011120> ссылка на книгу 2019

4.4 ТРЕБОВАНИЯ К РУКОВОДИТЕЛЮ ПРАКТИКИ

Руководителем практики от образовательного учреждения назначается педагогический работник, имеющий высшее образование, соответствующее профилю проводимой практики.

4.4.1 Руководитель практики от образовательного учреждения:

1. Разрабатывает тематику заданий для студентов;
2. Проводит консультации со студентами перед направлением их на практику с разъяснением целей, задач и содержания практики;
3. Принимает участие в распределении студентов по рабочим местам или перемещении их по видам работ;
4. Осуществляет контроль правильного распределения студентов в период практики; формирует группы в случае применения групповых форм проведения практики;
5. Проводит индивидуальные и групповые консультации в ходе практики;
6. Оказывает методическую помощь студентам при выполнении ими заданий и сборе материалов к отчету по практике;
7. Контролирует выполнение требований охраны труда, безопасности жизнедеятельности и пожарной безопасности;

4.4.2 Руководитель практики от организации:

1. Согласовывает программу практики, планируемые результаты практики, задание на практику;
2. Участвует в организации и проведении зачета по практике и экзамена квалификационного по профессиональному модулю;
3. Участвуют в организации и оценке результатов освоения общих и профессиональных Компетенций, освоенных студентами в период прохождения практики;
4. Проводит инструктаж студентов по ознакомлению с требованиями охраны труда, безопасности жизнедеятельности и пожарной безопасности

4.5 ТРЕБОВАНИЯ К СОБЛЮДЕНИЮ ТЕХНИКИ БЕЗОПАСНОСТИ И ПОЖАРНОЙ БЕЗОПАСНОСТИ

Регламентация требований по пожарной безопасности и техники безопасности осуществляется внутренними локальными актами образовательного учреждения.

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРАКТИКИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

По результатам усвоения программы практики студенты представляют руководителю практики отчет и дневник на студента-практиканта от руководителя базы практики.

По окончании практики студент защищает дневник, отчет с дифференцированной оценкой в присутствии комиссии, назначаемой заместителем директора по производственному обучению. Комиссия по защите дневников и отчетов должна состоять не менее чем из двух членов. В зависимости от места защиты дневника, отчета в состав комиссии входят: руководитель практики от образовательного учреждения, руководитель практики от базы практики, председатель ЦМК спецдисциплин и профессиональных модулей. Руководитель практики входит в состав комиссии и при защите отчетов в организации. Защита дневников и отчетов проводится в организации или в образовательном учреждении (если группа размещена по разным объектам практики). На базах практики защита должна проводиться в последний день практики. В образовательном учреждении председателем комиссии по защите дневников и отчетов по практике является заместитель директора по производственному обучению.

При оценке итогов работы студента на практике учитываются содержание и правильность оформления студентом дневника и отчета по практике, отзывы руководителей практики от организации, качество ответов на вопросы в ходе защиты отчета.

Зарегистрированные и защищенные дневники, отчеты хранятся в образовательном учреждении в течение трех лет в соответствии с номенклатурой дел.

Аттестация студента по итогам прохождения практики проводится только после сдачи документов по практике и фактической защиты отчета. Зачет по результатам практики принимает комиссия, назначенная заведующим практикой и состоящая из преподавателей-руководителей практики. Защита отчета по практике, как правило, представляет собой краткий, 8-10-минутный доклад студента и его ответы на вопросы членов комиссии.

После защиты отчета руководитель практики от образовательного учреждения дает свое заключение о заполнении дневника, отчета, выполнении программы практики и ставит по итогам дифференцированную оценку по пятибалльной шкале (5 «отлично», 4 «хорошо», 3 «удовлетворительно», 2 «неудовлетворительно»). Оценка одновременно проставляется в ведомость, зачетную книжку студента и «Дневник студента по учебной практике».

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки
ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия	Оценка «отлично» - разработан и обоснован вариант интеграционного решения с помощью графических средств среды разработки, указано хотя бы одно альтернативное решение; бизнес-процессы учтены в полном объеме; вариант оформлен в полном соответствии с требованиями стандартов; результаты верно сохранены в системе контроля версий. Оценка «хорошо» - разработана и прокомментирована архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы; вариант оформлен в соответствии с требованиями стандартов; результаты сохранены в системе контроля

компонент	версий. Оценка «удовлетворительно» - разработана и архитектура варианта интеграционного решения с помощью графических средств, учтены основные бизнес-процессы с незначительными упущениями; вариант оформлен в соответствии с требованиями стандартов с некоторыми отклонениями; результат сохранен в системе контроля версий.
ПК 2.2 Выполнять интеграцию модулей в программное обеспечение	Оценка «отлично» - в системе контроля версий выбрана верная версия проекта, проанализирована его архитектура, архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости); протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций в том числе с созданием классов-исключений (при необходимости); определены качественные показатели полученного проекта; результат интеграции сохранен в системе контроля версий. Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций (при необходимости); определены качественные показатели полученного проекта; результат интеграции сохранен в системе контроля версий. Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля (при необходимости); результат интеграции сохранен в системе контроля версий.
ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств	Оценка «отлично» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; проанализирована и сохранена отладочная информация; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в полном объеме; результаты отладки сохранены в системе контроля версий. Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий. Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта; выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий.
ПК 2.4 Осуществлять разработку тестовых наборов и тестовых	Оценка «отлично» - обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием в соответствии с минимальным размером тестового покрытия, выполнено

сценариев для программного обеспечения	тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, выявлены ошибки системных компонент (при наличии), заполнены протоколы тестирования. Оценка «хорошо»- обоснован размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты в соответствии с этим сценарием, выполнено тестирование интеграции и ручное тестирование, выполнено тестирование с применением инструментальных средств, заполнены протоколы тестирования. Оценка «удовлетворительно»- определен размер тестового покрытия, разработан тестовый сценарий и тестовые пакеты, выполнено тестирование интеграции и ручное тестирование, частично выполнено тестирование с применением инструментальных средств, частично заполнены протоколы тестирования.
ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Оценка «отлично» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде. Оценка «хорошо» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде. Оценка «удовлетворительно» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.

5.1 Экзаменационные билеты

- Вопрос 1:** Понятие требований, их классификация и уровни. **Вопрос 2:** Модели процесса разработки ПО (каскадная, спиральная, Agile). **Задание:** Разработайте фрагмент ТЗ на модуль, анализируя требования заказчика.
- Вопрос 1:** Методологии и стандарты работы с требованиями (ГОСТ 19, IEEE 830). **Вопрос 2:** Организация работы команды разработчиков (Scrum, Kanban). **Задание:** Составьте список функциональных требований к модулю на основе анализа проектной документации.
- Вопрос 1:** Анализ бизнес-процессов с использованием IDEF0 (блоки, стрелки, принципы). **Вопрос 2:** Основные подходы к интегрированию программных модулей. **Задание:** Постройте контекстную диаграмму IDEF0 для процесса «Интеграция модулей».
- Вопрос 1:** Описание требований с помощью унифицированного языка моделирования (UML). **Вопрос 2:** Этапы жизненного цикла программного обеспечения. **Задание:** Опишите сценарий «Вариант использования» для процесса авторизации пользователя в системе.
- Вопрос 1:** Спецификация требований (SRS): состав и структура. **Вопрос 2:** Современные принципы и методы разработки приложений (ООП, модульность). **Задание:** Проведите анализ взаимодействия компонент программного обеспечения на основе схемы архитектуры.
- Вопрос 1:** Определение характера взаимодействия компонентов ПО. **Вопрос 2:** Основные стандарты и регламенты, используемые при анализе предметной области. **Задание:** Составьте матрицу трассировки (требование → компонент → тест).
- Вопрос 1:** Классификация требований: функциональные и нефункциональные. **Вопрос 2:** Методы сбора и анализа данных для разработки требований (интервью, анкетирование). **Задание:** Разработайте перечень критериев для выбора технологии интеграции.
- Вопрос 1:** Цели и уровни интеграции программных модулей. **Вопрос 2:** Назначение и функции систем контроля версий (Git). **Задание:** Опишите процесс создания ветки, внесения изменений и слияния (merge) в Git.
- Вопрос 1:** Виды интеграции (Big Bang, Bottom-Up, Top-Down) и их особенности. **Вопрос 2:** Транспортные протоколы и стандарты форматирования сообщений. **Задание:** Разработайте схему интеграции двух модулей через API.
- Вопрос 1:** Автоматизация бизнес-процессов при интеграции. **Вопрос 2:** Выбор источников и

- приемников данных, сопоставление объектов. **Задание:** Составьте план интеграции модуля с внешней системой (например, платежным шлюзом).
11. **Вопрос 1:** Методы и способы идентификации сбоев и ошибок при интеграции приложений. **Вопрос 2:** Структура репозитория проекта (артефакты, протоколы). **Задание:** Опишите порядок настройки репозитория для командной работы (.gitignore, README).
 12. **Вопрос 1:** Организация постобработки данных при интеграции. **Вопрос 2:** Отладочные классы (Debug и Trace) и их использование. **Задание:** Разработайте план по устранению ошибок, возникающих при интеграции модулей (сбор логов).
 13. **Вопрос 1:** Методы отладки ПО (пошаговое выполнение, контрольные точки). **Вопрос 2:** Классификация ошибок (синтаксические, логические, ошибки доступа). **Задание:** Найдите и исправьте ошибку в предоставленном фрагменте кода (например, деление на ноль).
 14. **Вопрос 1:** Инструменты отладки в современных средах разработки (IDE). **Вопрос 2:** Организация обработки исключительных ситуаций (try-catch). **Задание:** Опишите алгоритм отладки модуля, который выдает ошибку «NullReferenceException».
 15. **Вопрос 1:** Принципы работы отладчиков: точки останова, окна Locals и Watch. **Вопрос 2:** Методы и схемы обработки исключений. **Задание:** Составьте план по исправлению ошибки, возникающей при обработке некорректных входных данных.
 16. **Вопрос 1:** Основные понятия модульного программирования (сплочённость, сцепление). **Вопрос 2:** Инструментарий тестирования и анализа качества. **Задание:** Проанализируйте код на предмет соответствия стандартам кодирования (названия переменных, отступы).
 17. **Вопрос 1:** Отладочные классы и их применение в проекте. **Вопрос 2:** Приемы работы с инструментальными средствами тестирования. **Задание:** Разработайте чек-лист по проверке исправленной ошибки перед сдачей заказчику.
 18. **Вопрос 1:** Методы и способы идентификации сбоев системных компонентов. **Вопрос 2:** Технологии решения задач планирования и контроля развития проекта. **Задание:** Выполните отладку модуля с использованием условной компиляции (Debug).
 19. **Вопрос 1:** Цели, задачи и виды тестирования (функциональное, регрессионное). **Вопрос 2:** Стандарты качества программной документации (ГОСТ 19). **Задание:** Разработайте тест-кейс для проверки функции авторизации.
 20. **Вопрос 1:** Разработка тестовых наборов и тестовых сценариев. **Вопрос 2:** Тестовое покрытие и метрики качества (плотность дефектов). **Задание:** Оцените количество необходимых тестов для модуля с 5 ветками условий.
 21. **Вопрос 1:** Методы тестирования «белого ящика» (покрытие веток, путей). **Вопрос 2:** Методы тестирования «черного ящика» (классы эквивалентности). **Задание:** Разработайте тестовый пакет для модуля, обрабатывающего диапазон чисел.
 22. **Вопрос 1:** Тестирование интерфейса пользователя средствами инструментальной среды. **Вопрос 2:** Верификация и аттестация программного обеспечения. **Задание:** Составьте план интеграционного тестирования для двух взаимодействующих модулей.
 23. **Вопрос 1:** Анализ спецификаций и выявление ошибок в требованиях. **Вопрос 2:** Назначение и процедура инспектирования кода (Code Review). **Задание:** Проведите инспекцию предоставленного кода на предмет соблюдения стандартов кодирования.
 24. **Вопрос 1:** Роль и методы функционального тестирования. **Вопрос 2:** Документирование результатов тестирования (отчеты, протоколы). **Задание:** Создайте отчет о тестировании модуля с указанием найденных ошибок и их статусов.
 25. **Вопрос 1:** Диаграмма Вариантов использования и диаграмма Последовательности (UML). **Вопрос 2:** Роль архитектуры программного средства в интеграции. **Задание:** Постройте диаграмму последовательности для процесса добавления товара в корзину.
 26. **Вопрос 1:** Диаграмма Кооперации и диаграмма Развертывания (UML). **Вопрос 2:** Методы хранения и обработки данных при интеграции. **Задание:** Разработайте схему развертывания системы с сервером и клиентскими рабочими местами.
 27. **Вопрос 1:** Диаграмма Деятельности и диаграмма Состояний (UML). **Вопрос 2:** Схемы взаимодействия модулей и компонентов. **Задание:** Постройте диаграмму классов для предметной области «Интернет-магазин».
 28. **Вопрос 1:** Диаграмма Компонентов (UML): назначение и элементы. **Вопрос 2:** Основы построения диаграмм потоков данных (DFD). **Задание:** Создайте диаграмму компонентов для

системы, состоящей из интерфейса и серверной части.

29. **Вопрос 1:** Влияние архитектуры на процесс интеграции модулей. **Вопрос 2:** Технологии разработки программных приложений (MSF, RUP). **Задание:** Сравните две архитектуры интеграции (монолитная vs микросервисная) и выберите оптимальную по заданным критериям.
30. **Вопрос 1:** Построение концептуальной, логической и физической моделей ПО. **Вопрос 2:** Принципы построения диаграмм (нотация, связь элементов). **Задание:** Разработайте полный пакет UML-диаграмм (Use Case, Class, Activity) для модуля отчетности.

5.2 Ключи к экзаменационным билетам

Билет 1

- 1. *Требования* — это описание того, что система должна делать. Уровни: *Бизнес-требования* (цели организации), *Пользовательские требования* (задачи пользователей), *Функциональные* (что делает система), *Нефункциональные* (как делает: скорость, безопасность).
- 2. *Модели процесса разработки*: **Каскадная** (строгая последовательность), **Спиральная** (итерации с анализом рисков), **Agile** (гибкая разработка короткими спринтами).
- Практическое задание: Алгоритм:** 1) Выделить цели. 2) Описать функции. 3) Указать ограничения (сроки, технологии).

Билет 2

- 1: Стандарты: **ГОСТ 19** (ЕСПД), **IEEE 830** (SRS).
- 2. Методологии работы в команде: **Scrum** (спринты, роли) и **Kanban** (визуальные доски).
- Практическое задание: Алгоритм:** Собрать перечень функций из описания заказчика и классифицировать их по важности.

Билет 3

- 1: **IDEF0** описывает процессы в виде блоков и стрелок (вход, выход, управление, механизм).
- 2. *Подходы к интеграции*: **Big Bang** (все сразу), **Bottom-Up** (снизу вверх), **Top-Down** (сверху вниз), **Sandwich** (комбинированный).
- Практическое задание: Построение:** Контекстная диаграмма для процесса "Интеграция модулей": блок, слева — исходный код, сверху — стандарты, снизу — инструменты.

Билет 4

- 1. **UML** — язык графического моделирования для описания требований.
- 2. *Этапы ЖЦ*: Анализ, Проектирование, Разработка, Тестирование, Внедрение, Сопровождение.
- Практическое задание: Сценарий:** Актёр «Пользователь» → действие «Ввести логин/пароль» → Система → «Выдать ошибку или открыть кабинет».

Билет 5

- 1: **SRS** — спецификация требований. Состав: введение, общее описание, конкретные требования.
- 2. *Методы разработки*: **ООП** (объекты, классы), **Модульное программирование** (деление на блоки).
- Практическое задание: Матрица:** Таблица, где столбцы — требования, компоненты, тесты.

Билет 6

- 1: Взаимодействие определяется схемами архитектуры (клиент-сервер, микросервисы).
- 2. *Стандарты анализа*: **ГОСТ 34**, **РД 50-34.698**.
- Практическое задание: Матрица трассировки:** Связывает каждое требование ТЗ с конкретным тест-кейсом для проверки.

Билет 7

- 1: **Функциональные** (что делает), **Нефункциональные** (производительность, удобство).
- 2. *Методы сбора*: интервью, анкетирование, наблюдение.
- Практическое задание: Критерии:** Например, скорость обмена (с), стоимость лицензии, простота поддержки.

Билет 8

- 1: **Интеграция** — объединение компонент в единую систему.
- 2. **Git** — система контроля версий для совместной работы.
- Задание:** Создать ветку → внести изменения → git add → git commit → git merge.

Билет 9

- 1: *Big Bang* — все модули в конце. *Bottom-Up* — нижние сначала, драйверы. *Top-Down* — верхние сначала, заглушки.
- 2. *Транспортные протоколы*: TCP/IP, HTTP/HTTPS.
- **Практическое задание**: *Схема API*: Опишите, как один модуль вызывает метод другого через REST API (JSON).

Билет 10

- 1: *Автоматизация* — устранение ручного ввода.
- 2. *Сопоставление данных*: Схема «Источник → Трансформация → Приемник».
- **Практическое задание**: *План*: Создание платежного шлюза: 1) Авторизация, 2) Передача суммы, 3) Получение статуса.

Билет 11

- 1: *Идентификация сбоев*: Логирование (операторы протоколирования).
- 2. *Структура репозитория*: Папки src, tests, docs, configs.
- **Практическое задание**: *Алгоритм*: 1) Создать .gitignore (исключить bin, obj). 2) Описать проект в README.md.

Билет 12

- 1: *Постобработка*: Фильтрация, валидация.
- 2. *Отладочные классы*: Debug (для отладки) и Trace (для логирования).
- **Практическое задание**: *План*: Сбор логов (log.txt), анализ стек-трейса, исправление, повторная проверка.

Билет 13

- 1: *Методы отладки*: Точки останова (F9), пошаговое выполнение (F10/F11), окна Locals/Watch.
- 2. *Ошибки*: Синтаксические (опечатки), логические (неверный алгоритм), ошибки доступа.
- **Практическое задание**: *Проверка*: Проверить знаменатель на 0 (If b = 0 Then).

Билет 14

- 1: *Инструменты*: Отладчик IDE (Visual Studio, PyCharm).
- 2. *Обработка исключений*: Блок Try...Catch...Finally для перехвата ошибок и вывода сообщений.
- **Практическое задание**: *Алгоритм*: Проверить объект на null, использовать безопасный вызов ?..

Билет 15

- 1. *Отладчики*: Step Over, Step Into.
- 2. *Обработка исключений*: Throw, Custom Exceptions.
- **Практическое задание**: *План*: Проверка типов, ограничение диапазона, сообщение об ошибке.

Билет 16

- 1: *Сплоченность* (связность внутри модуля), *Сцепление* (зависимости между модулями).
- 2. *Стандарты кодирования*: Единый стиль именования, комментирование.
- **Практическое задание**: *Анализ*: Проверить, что переменные названы осмысленно (camelCase), есть отступы.

Билет 17

- 1: *Отладочные классы*: Debug.Assert.
- 2. *Инструментальные средства*: Линтеры (ESLint), статические анализаторы.
- **Практическое задание**: *Чек-лист*: Код соответствует ТЗ, тесты пройдены, логи чистые.

Билет 18

- 1: *Методы идентификации сбоев*: Мониторинг (Zabbix), анализ журналов.
- 2. *Технологии планирования*: Microsoft Project, Jira.
- **Практическое задание**: *Отладка*: Условная компиляция #If DEBUG Then для вывода отладочной информации.

Билет 19

- 1: *Виды*: Модульное (изоляция), интеграционное, системное.
- 2. *Стандарты*: ГОСТ 19.301 (Программа и методика испытаний).
- **Практическое задание**: *Тест-кейс*: Шаги, ожидаемый результат, предусловия, фактический результат.

Билет 20

- 1: *Тестовые наборы и сценарии* — структурированные документы для проверки.

- 2. *Покрытие*: Процент исполненного кода тестами. *Метрики*: Плотность дефектов.
- **Практическое задание**: *Оценка*: Для 5 веток нужно минимум 5 тест-кейсов + тест на базовый случай.

Билет 21

- 1. *Белый ящик*: Знание кода, покрытие веток/путей (цикломатическая сложность).
- 2. *Черный ящик*: Классы эквивалентности, граничные значения.
- **Практическое задание**: *Пакет*: Валидный диапазон (1), невалидный (0), граничный (1).

Билет 22

- **Ответ на вопрос 1**: *Тестирование интерфейса*: Проверка кнопок, полей (UI).
- 2. *Верификация* (правильно ли), *Аттестация* (тот ли продукт).
- **Практическое задание**: *План*: 1) Связь модулей, 2) Передача данных, 3) Ошибки.

Билет 23

- **Ответ на вопрос 1**: *Анализ спецификаций* — поиск ошибок в ТЗ.
- 2. *Инспекция кода (Code Review)* — проверка кода коллегами.
- **Практическое задание**: *Инспекция*: Проверка соответствия стандартам и стилю.

Билет 24

- **Ответ на вопрос 1**: *Функциональное тестирование*: Проверка функций.
- 2. *Документирование*: Отчеты, протоколы, баг-репорты.
- **Практическое задание**: *Отчет*: ID, название, статус (Passed/Failed).

Билет 25

- **Ответ на вопрос 1**: *Диаграмма Вариантов использования* (кто, что делает),
- *Диаграмма Последовательности* (обмен сообщениями).
- 2 *Роль архитектуры* — каркас интеграции.
- **Практическое задание**: *Построение*: Пользователь → Система → База данных (стрелки сверху вниз).

Билет 26

- **Ответ на вопрос 1**: *Диаграмма Кооперации* (связи), *Диаграмма Развертывания* (серверы, ПК). *Методы хранения*: Реляционные БД.
- **Практическое задание**: *Схема*: Клиенты ↔ Сервер приложений ↔ Сервер БД.

Билет 27

- **Ответ на вопрос 1**: *Деятельности* (поток действий), *Состояний* (переходы). *Схемы взаимодействия*: Шина данных, брокеры сообщений.
- **Практическое задание**: *Классы*: Класс Товар (Название, Цена), класс Заказ (Номер, Дата).

Билет 28

- **Ответ на вопрос 1**: *Диаграмма Компонентов* (логические модули, зависимости). *DFD* — потоки данных.
- **Практическое задание**: *Диаграмма*: Интерфейс UI.dll → Сервер Backend.dll → БД DataBase.mdf.

Билет 29

- **Ответ на вопрос 1**: *Влияние архитектуры*: Микросервисы упрощают независимую интеграцию. *Технологии*: MSF, RUP.
- **Практическое задание**: *Сравнение*: Микросервисы (масштабируемость) против Монолита (простота).

Билет 30

- **Ответ на вопрос 1**: *Модели*: Концептуальная (что), Логическая (как), Физическая (где). *Принципы*: Читаемость, стандартизация.
- **Практическое задание**: *Пакет*: Use Case, Class, Activity для модуля отчетности.