

Министерство образования Ростовской области
государственное бюджетное профессиональное образовательное учреждение
Ростовской области
«Белокалитвинский гуманитарно-индустриальный техникум»
(ГБПОУ РО «БГИТ»)

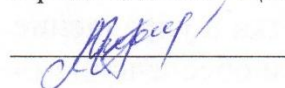
Приложение 4.33
к ОПОП по специальности
09.02.11 «Разработка и управление
программным обеспечением»

КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ
для проведения процедуры оценки уровня освоения обучающимися
ФГОС СПО по ПМ.02 «Разработка и интеграция модулей программного
обеспечения» специальность 09.02.11 Разработка и управление
программным обеспечением

г. Белая Калитва

Рассмотрена
на заседании цикловой комиссии
специальности 09.02.11 Разработка и
управление программным обеспечением
Протокол № 4
от «23» декабря 2025г.

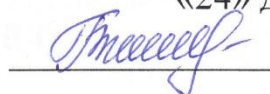
Председатель ЦК

 Комиссарова И.В.

УТВЕРЖДАЮ

Заместитель директора по УМР

«24» декабря 2025 г.

 Г. В. Шматко

Комплект оценочных средств разработан в соответствии с Федеральным государственным образовательным стандартом среднего профессионального образования по специальности 09.02.11 «Разработка и управление программным обеспечением», утверждённым приказом Министерства просвещения Российской Федерации от 24 февраля 2025 г. № 138; Рабочей программой профессионального модуля ПМ.02 «Разработка и интеграция модулей программного обеспечения» и Методическими рекомендациями по формированию комплекта оценочных материалов образовательной программы среднего профессионального образования.

Организация - разработчик: государственное бюджетное профессиональное образовательное учреждение Ростовской области «Белокалитвинский гуманитарно-индустриальный техникум»

Разработчик: Комиссарова Ирина Викторовна – преподаватель ГБПОУ РО «БГИТ»

СОДЕРЖАНИЕ

1	Спецификация комплекта оценочных средств.	4
1.1	Назначение комплекта оценочных средств.	4
1.2	Общее количество тестовых заданий.	4
1.3	Распределение тестовых заданий по компетенциям и дисциплинам ...	5
1.4	Типы, уровни сложности и время выполнения тестовых заданий.....	8
1.5	Система оценивания выполнения тестовых заданий.....	10
1.6	Описание дополнительных материалов и оборудования, необходимых для выполнения тестовых заданий.....	11
1.7	Продолжительность выполнения.....	11
1.8	Инструкция по организации и проведению тестирования.....	11
2	Тестовые задания	13
2.1	Вариант 1	13
2.2	Вариант 2	19
2.3	Вариант 3	25
2.4	Вариант 4	31
3	Ключи к оцениванию тестовых заданий	37

1. Спецификация комплекта оценочных средств.

1.1. Назначение комплекта оценочных средств.

Контрольно-оценочные средства (далее – КОС) подготовлены с целью оценки результатов освоения программы подготовки специалистов среднего звена

- УГС 09.00.00 «Информатика и вычислительная техника»
- специальности 09.02.11 «Разработка и управление программным обеспечением»
- ПМ.02 «Разработка и интеграция модулей программного обеспечения»

1.2. Общее количество тестовых заданий.

Количество заданий в комплекте оценочных материалов

Код компетенции	Наименование компетенции	Количество заданий
ПК 2.1	Проектировать модули программного обеспечения	3
ПК 2.2	Разрабатывать модули программного обеспечения	6
ПК 2.3	Выполнять интеграцию модулей и компонентов программного обеспечения	4
ПК 2.4	Выполнять тестирование и отладку программного обеспечения	4
ПК 2.5	Осуществлять документирование программных модулей программного обеспечения.	3
ОК 1.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	2
ОК 2.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.	3
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях.	3
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде.	2
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.	2
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации	2

	межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения.	
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях.	3
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	1
ОК 9	Пользоваться профессиональной документацией на государственном и иностранных языках.	2

1.3.Распределение тестовых заданий по компетенциям и дисциплинам

Распределение заданий по компетенциям и дисциплинам

Код компетенции	Наименование компетенции	Наименование индикаторов сформированности компетенции	Наименование дисциплины / модуля/ практики	Семестр	Номер задания
ПК 2.1	Проектировать модули программного обеспечения	Умение применять принципы ООП, выбирать архитектурные стили и паттерны проектирования	МДК.02.01	3-4	1, 5, 6
ПК 2.2	Разрабатывать модули программного обеспечения	Умение писать код на C#, использовать системы контроля версий, применять принципы разработки	МДК 02.01, УП.02	3-4	4, 7, 10, 12, 16, 20
ПК 2.3	Выполнять интеграцию модулей и компонентов программного обеспечения	Умение применять методы интеграции, использовать API, выполнять развертывание	МДК.02.01, ПП.02	4	8, 13, 14, 19
ПК 2.4	Выполнять тестирование и отладку программного обеспечения	Умение разрабатывать тесты, проводить различные виды тестирования, анализировать результаты	МДК.02.01, УП.02	3-4	3, 11, 17, 18
ПК 2.5	Осуществлять документирование программных	Умение составлять техническую документацию,	МДК.02.01	3-4	2, 9, 15

	модулей программного обеспечения.	оформлять код, создавать отчеты			
ОК 1.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	Способность анализировать требования, выбирать технологии	МДК.02.01	3-4	3, 6
ОК 2.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.	Умение работать с документацией, использовать IDE, системы контроля версий	МДК.02.01, УП.02	3-4	1, 7, 16
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие	Умение оценивать трудоемкость, планировать этапы разработки	МДК.02.01, ПП.02	4	8, 14, 19
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде.	Умение распределять роли, проводить код-ревью, согласовывать решения	УП.02, ПП.02	3-4	4, 10
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.	Умение составлять ТЗ, документацию, презентовать результаты	МДК.02.01, ПП.02	4	9, 17
ОК 6	Проявлять гражданско-патриотическую позицию,	Знание правовых норм, этики в IT, антикоррупционного поведения	Все виды занятий	3-4	5, 15

	демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей				
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства	Умение оптимизировать код, использовать энергоэффективные технологии	МДК.02.01	3-4	11, 18, 20
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Умение применять знания о здоровьесберегающих технологиях, принципах эргономики	Все виды занятий	3-4	12
ОК 9	Пользоваться профессиональной документацией на государственном и иностранных языках.	Умение читать документацию на английском	МДК.02.01	3-4	2, 13

1.4. Типы, уровни сложности и время выполнения тестовых заданий

Распределение заданий по типу и уровням сложности

Код компетенции	Индикаторы сформированности компетенции	Номер задания	Тип задания	Уровень сложности задания	Время выполнения (мин)
ПК 2.1	Умение применять принципы ООП, выбирать архитектурные стили и паттерны проектирования	1, 5, 6	Соответствие, Выбор одного ответа, Выбор одного ответа	Б, Б, П	4, 5, 6
ПК 2.2	Умение писать код на С#, использовать системы контроля версий, применять принципы разработки	4, 7, 10, 12, 16, 20	Последовательнось, Выбор нескольких ответов, Открытый вопрос, Написание кода, Выбор нескольких ответов, Открытый вопрос	Б, П, В, В, П, П	5, 6, 10, 8, 6, 6
ПК 2.3	Умение применять методы интеграции, использовать API, выполнять развертывание	8, 13, 14, 19	Выбор нескольких ответов, Соответствие, Последовательнось, Написание кода	П, Б, Б, В	7, 4, 5, 8
ПК 2.4	Умение разрабатывать тесты, проводить различные виды тестирования, анализировать результаты	3, 11, 17, 18	Последовательнось, Анализ кода, Открытый вопрос, Анализ кода	Базовый, Высокий, Повышенный, Повышенный	4, 8, 8, 7
ПК 2.5	Умение составлять техническую документацию, оформлять код, создавать отчеты	2, 9, 15	Соответствие, Открытый вопрос, Выбор одного ответа	Базовый, Высокий, Базовый	3, 10, 5
ОК 1.	Способность анализировать требования, выбирать технологии.	3, 6	Последовательнось, Выбор одного ответа	Базовый, Повышенный	4, 6
ОК 2.	Умение работать с документацией,	1, 7, 16	Соответствие, Выбор	Базовый, Повышенный	4, 6, 6

	использовать IDE, системы контроля версий.		нескольких ответов, Выбор нескольких ответов	ый, Повышенн ый	
ОК 3	Умение оценивать трудоемкость, планировать этапы разработки.	8, 14, 19	Выбор нескольких ответов, Последовательность, Написание кода	Повышенн ый, Базовый, Высокий	7, 5, 8
ОК 4	Умение распределять роли, проводить код-ревью, согласовывать решения.	4, 10	Последовательность, Открытый вопрос	Базовый, Высокий	5, 10
ОК 5	Умение составлять ТЗ, документацию, презентовать результаты.	9, 17	Открытый вопрос, Открытый вопрос	Высокий, Повышенн ый	10, 8
ОК 6	Умение применять правовые нормы, этические принципы, стандарты антикоррупционного поведения в профессиональной деятельности	5, 15	Выбор одного ответа, Выбор одного ответа	Базовый, Базовый	5, 5
ОК 7	Умение оптимизировать код, использовать энергоэффективные технологии.	11, 18, 20	Анализ кода, Анализ кода, Открытый вопрос	Высокий, Повышенн ый, Повышенн ый	8, 7, 6
ОК 8	Умение применять знания о здоровьесберегающих технологиях, принципах эргономики и безопасной организации рабочего места	12	Написание кода	Базовый	8
ОК 9	Умение читать документацию на английском (технологии, фреймворки).	2, 13	Соответствие, Соответствие	Базовый, Базовый	3, 4

1.5. Система оценивания выполнения тестовых заданий

Система оценивания заданий

Номер задания	Указания по оцениванию	Результат оценивания (баллы, полученные за выполнение задания/ характеристика правильности ответа)
1, 2, 13	Соответствие	1 балл за каждое верное соответствие, всего 4 балла. Лишний вариант не оценивается.
3, 4	Последовательность	1 балл за каждое верное соответствие, всего 4 балла. Лишний вариант не оценивается.
5, 6, 15	Выбор одного ответа + обоснование	1 балл за правильный выбор, 1 балл за верное обоснование. Итого: 2 балла.
7, 8, 16	Выбор нескольких ответов + обоснование	2 балла за все верные ответы (частичное оценивание: 1 балл за 2 из 3 верных), 1 балл за обоснование. Итого: до 3 баллов.
9, 10, 17, 20	Открытый вопрос	Оценивается по критериям: структура (2 балла), содержание (2 балла), полнота (1 балл). Итого: до 5 баллов.
11, 18	Анализ кода	1 балл за выявление ошибки, 2 балла за верное исправление. Итого: до 3 баллов.
12, 19	Написание кода	3 балла за рабочую функцию с учётом всех условий, 2 балла при незначительных недочётах.

Максимальное количество баллов – 54.

Оценка «5» (отлично) соответствует результатам, содержащим 48 – 54 баллов (90-100%).

Оценка «4» (хорошо) — 37 – 47 баллов (70-89%).

Оценка «3» (удовлетворительно) — 27 – 36 баллов (50-69%).

Оценка «2» (неудовлетворительно) — менее 27 баллов (менее 50 %).

1.6. Описание дополнительных материалов и оборудования, необходимых для выполнения тестовых заданий

Код компетенции	Наименование компетенции	Дополнительные материалы и оборудование
ПК 2.1 – 2.5	Проектировать модули программного обеспечения	<i>Аппаратное обеспечение:</i> Компьютер с процессором не ниже Intel Core i5 / AMD Ryzen 5 Оперативная память: 8 ГБ минимум Монитор с разрешением не ниже 1920×1080 Клавиатура, мышь Доступ в интернет (проводной или Wi-Fi) <i>Программное обеспечение:</i> ОС Windows 10/11 Visual Studio 2022 Community/Professional .NET SDK 6.0+ Git for Windows Docker Desktop Браузер (Chrome, Edge) Офисный пакет (Word, Excel) или аналог
ОК 1–ОК 9	Общие компетенции (все)	• Компьютер с доступом в интернет • Браузер для поиска информации, нормативной базы, документации • Электронные библиотеки, справочные системы (КонсультантПлюс, Гарант — для правовых аспектов)

1.7. Продолжительность выполнения

Общее время, отведенное на выполнение задания, не более двух академических часов.

Распределение времени рекомендуется следующим образом:

1. Блок А (задания на соответствие, последовательность, выбор одного ответа (базовые) 1, 2, 3, 4, 5, 6, 13, 14) – 30 минут

2. Блок Б (задания на выбор нескольких ответов с обоснованием, анализ кода, простые открытые вопросы 7, 8, 11, 15, 16, 18) – 25 минут

3. Блок В (задания на написание кода, сложные открытые вопросы, проектирование 9, 10, 12, 17, 19, 20) – 35 минут

При необходимости допускается увеличение времени до 120 минут для обучающихся с особыми образовательными потребностями.

1.8. Инструкция по организации и проведению тестирования

Действия преподавателя во время проведения тестирования:

1) приветствует обучающихся;

- 2) объясняет обучающимся цель предстоящей работы: получение объективной информации о состоянии качества подготовки и освоения обучающимися основной профессиональной образовательной программы; доводит правила проведения тестирования;
- 3) устанавливает соответствие представленного профессиональной образовательной организацией списка группы и списка обучающихся, фактически присутствующих;
- 4) раздает обучающимся бланки заданий и бланки ответов;
- 5) знакомит со структурой заданий, отмечает особенности, объясняет правила заполнения бланков ответов;
- 6) обращается к обучающимся: «Есть ли вопросы?» (не имеет права отвечать на вопросы по содержанию заданий);
- 7) фиксирует время начала и время окончания работы на видном месте (доске);
- 8) отслеживает полную самостоятельность обучающихся при выполнении работы;
- 9) предупреждает обучающихся о том, что до конца процедуры осталось 10 мин;
- 10) по истечении времени выполнения работы собирает материалы работы (заполненные бланки ответов); проверяет соответствие количества работ числу испытуемых обучающихся;
- 11) благодарит обучающихся за участие в работе, желает успехов.

После завершения задания преподавателям выдаются эталоны ответов для каждого варианта заданий, работы проверяются преподавателями, проводящими проверочную работу согласно системе оценивания заданий и работы в целом. По итогу заполняется протокол.

Действия обучающихся при выполнении тестового задания:

- 1) внимательно читайте все задания работы и указания по его выполнению;
- 2) если не можете выполнить очередное задание, не тратьте время, переходите к следующему;
- 3) только выполнив все задания, вернитесь к тем, которые у вас не получились сразу;
- 4) старайтесь работать быстро и аккуратно;
- 5) все задания выполнять на выданных листах;
- 6) если ошибся, зачеркни ошибку и выбери другой ответ;
- 7) после выполнения всех заданий осуществить проверку.
- 8) сдать заполненные бланки ответов преподавателю.

2. Тестовые задания

2.1. Вариант 1

Задание 1 (ПК 2.1, ОК 2). Установите соответствие между этап разработки ПО с его основной задачей. Время: 4 мин.

Инструкция: Соотнесите этап (1–4) с задачей (А–Д). В правом столбце один лишний вариант.

Этап		Задача	
1.	Сбор требований	А.	Проверка работоспособности модулей в единой системе
2.	Проектирование архитектуры	Б.	Формализация потребностей заказчика в функциональные спецификации
3.	Интеграционное тестирование	В.	Определение структуры компонентов системы и их взаимодействия
4.	Рефакторинг кода	Г.	Улучшение внутренней структуры кода без изменения внешнего поведения
		Д.	Написание исходного кода на выбранном языке программирования

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 2 (ПК 2.5, ОК 9). Установите соответствие между типом документации и его описанием. Время: 3 мин.

Инструкция: Соотнесите тип (1–4) с описанием (А–Д).

Тип документации		Описание	
1.	Техническое задание (ТЗ)	А.	Описание API, классов и методов для разработчиков
2.	Пользовательская инструкция	Б.	Юридически значимый документ с требованиями заказчика
3.	Техническая документация (кодовая)	В.	Пошаговое руководство для конечного пользователя
4.	Отчет об тестировании	Г.	Документ с перечнем найденных дефектов и результатов тестов
		Д.	Описание структуры базы данных и связей между таблицами

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 3 (ПК 2.4, ОК 1). Расположите этапы модульного тестирования (юнит-тестирования) в правильной последовательности. Время: 4 мин.

Инструкция: Запишите последовательность цифр.

1. Написание тестового метода с использованием NUnit/xUnit
2. Анализ результатов тестирования
3. Выделение тестируемого метода или класса
4. Запуск теста через Test Runner
5. Исправление выявленных ошибок в коде

Ответ:

Задание 4 (ПК 2.2, ОК 4). Расположите действия при разрешении конфликта слияния (merge conflict) в системе Git в правильной последовательности. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. Выполнить git merge feature-branch
2. Открыть файл с конфликтами в редакторе (например, в Visual Studio)
3. Вручную отредактировать файл, оставив нужные изменения
4. Добавить разрешенный файл в индекс (git add)
5. Создать новый коммит (git commit)
6. Получить уведомление о конфликте слияния

Ответ:

Задание 5 (ПК 2.1, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какой принцип ООП наиболее важен для обеспечения легкой замены одного модуля другим без изменения остальной системы?

1. Инкапсуляция
2. Наследование
3. Полиморфизм
4. Абстракция

Ответ:

Обоснование:

Задание 6 (ПК 2.1, ОК 1). Выберите ОДИН правильный ответ и обоснуйте его. Время: 6 мин.

Инструкция: Какой паттерн проектирования следует использовать для создания сложного объекта пошагово, позволяя изменять этапы его создания?

1. Фабричный метод (Factory Method)
2. Строитель (Builder)
3. Одиночка (Singleton)
4. Адаптер (Adapter)

Ответ:

Обоснование:

Задание 7 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Инструкция: Какие инструменты используются для контроля версий кода в разработке на C#? (Выберите 3)

1. Jenkins
2. Git
3. GitHub / GitLab
4. Docker
5. Visual Studio с интеграцией Git

Ответ:

Обоснование:

Задание 8 (ПК 2.3, ОК 3). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 7 мин.

Инструкция: Какие из перечисленных практик относятся к методологии DevOps при интеграции модулей? (Выберите 3)

1. Непрерывная интеграция (CI)
2. Ручное развертывание на продакшн
3. Использование систем мониторинга
4. Отсутствие автоматизации тестирования
5. Инфраструктура как код (IaC)

Ответ:

Обоснование:

Задание 9 (ПК 2.5, ОК 5). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите структуру технического задания на разработку модуля «Управление пользователями» для desktop-приложения на C# (например, с использованием Windows Forms или WPF). Включите не менее 5 основных разделов.

Ответ:

Задание 10 (ПК 2.2, ОК 4). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите процесс взаимодействия двух разработчиков при совместной работе над одной функцией в Git с использованием C#. Включите этапы: ветвление, коммиты, создание пул-реквеста (merge request), код-ревью.

Ответ:

Задание 11 (ПК 2.4, ОК 7). Анализ кода. Время: 8 мин.

Инструкция: В приведенном фрагменте кода на C# найдите потенциальную ошибку и предложите исправление.

```
csharp
public double CalculateAverage(List<int> numbers)
{
    int total = 0;
    foreach (var num in numbers)
    {
        total += num;
    }
    double average = total / numbers.Count;
    return average;
}
```

Ответ (укажите ошибку и исправленный код):

Задание 12 (ПК 2.2, ОК 8). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию IsPalindrome(string text), которая возвращает true, если строка является палиндромом (игнорируя регистр и пробелы), и false — в противном случае.

Ответ (код функции):

Задание 13 (ПК 2.3, ОК 9). Установите соответствие между понятием и его определением в контексте интеграции ПО. Время: 4 мин.

Инструкция: Соотнесите понятие (1–4) с определением (А–Д).

Понятие		Определение	
1.	API	А.	Независимо развертываемый сервис, выполняющий одну бизнес-функцию
2.	Микросервис	Б.	Набор правил для взаимодействия компонентов
3.	DI (Dependency Injection)	В.	Паттерн, при котором зависимости передаются извне
4.	Контейнеризация	Г.	Упаковка приложения со всеми зависимостями в изолированную среду
		Д.	Протокол для передачи данных между клиентом и сервером

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 14 (ПК 2.3, ОК 3). Установите последовательность этапов внедрения обновления в работающую систему. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. Резервное копирование базы данных и файлов
2. Тестирование обновления на staging-окружении
3. Уведомление пользователей о плановых работах
4. Развертывание обновления на продакшн-серверах
5. Откат изменений в случае критических ошибок

Ответ:

Обоснование:

Задание 15 (ПК 2.5, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какое действие является нарушением профессиональной этики разработчика?

1. Использование open-source кода с указанием лицензии
2. Написание собственной реализации алгоритма
3. Копирование чужого кода без указания автора и лицензии
4. Участие в обсуждении стандартов кодирования

Ответ:

Обоснование:

Задание 16 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Инструкция: Какие из перечисленных утверждений характеризуют принцип DRY (Don't Repeat Yourself)? (Выберите 3)

1. Каждая часть знания должна иметь единственное представление в системе.
2. Дублирование кода допустимо, если это упрощает понимание.
3. Логика должна быть вынесена в отдельные методы или классы.
4. Код должен быть максимально компактным, даже в ущерб читаемости.
5. Изменения в одной точке не должны требовать дублирования изменений.

Ответ:

Обоснование:

Задание 17 (ПК 2.4, ОК 5). Открытый вопрос. Время: 8 мин.

Инструкция: Опишите план тестирования для модуля «Авторизация пользователя». Включите не менее 4 типов проверок.

Ответ:

Задание 18 (ПК 2.4, ОК 7). Анализ кода. Время: 7 мин.

Инструкция: В приведенном фрагменте SQL-запроса в C# найдите потенциальную уязвимость и предложите безопасную альтернативу.

csharp

```
string query = $"SELECT * FROM Users WHERE Login = '{login}' AND Password = '{password}'";
```

Ответ (укажите уязвимость и безопасный вариант):

Задание 19 (ПК 2.3, ОК 3). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию, которая принимает список чисел и возвращает список, содержащий только четные числа.

Ответ (код функции):

Задание 20 (ПК 2.2, ОК 7). Открытый вопрос. Время: 6 мин.

Инструкция: Опишите, как принципы энергоэффективности и «зеленого» IT могут быть применены при разработке и эксплуатации программного обеспечения (приведите 2-3 примера).

Ответ:

2.2.Вариант 2

Задание 1 (ПК 2.1, ОК 2). Установите соответствие между методом разработки ПО и его описанием. Время: 4 мин.

Инструкция: Соотнесите этап (1–4) с задачей (А–Д). В правом столбце один лишний вариант.

Методы		Описания	
1.	Итеративная разработка	А.	Последовательная разработка с циклами, каждый из которых включает планирование, проектирование, реализацию и оценку
2.	Инкрементальная разработка	Б.	Разработка небольшими порциями (инкрементами), каждый из которых добавляет новую функциональность
3.	Спиральная модель	В.	Модель, в которой тестирование планируется параллельно с соответствующими этапами разработки
4.	V-модель	Г.	Циклический процесс с акцентом на анализ рисков
		Д.	Метод, при котором весь функционал разрабатывается и тестируется одновременно

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 2 (ПК 2.5, ОК 9). Установите соответствие между типом диаграммы UML и её назначением. Время: 3 мин.

Инструкция: Соотнесите тип (1–4) с описанием (А–Д).

Тип документации		Описание	
1.	Диаграмма классов	А.	Описание структуры системы с точки зрения классов и их отношений
2.	Диаграмма последовательности	Б.	Описание взаимодействия объектов во времени
3.	Диаграмма развёртывания	В.	Описание физического размещения компонентов системы
4.	Диаграмма вариантов использования	Г.	Описание функциональных требований системы с точки зрения акторов
		Д.	Описание потоков данных в системе

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 3 (ПК 2.4, ОК 1). Расположите этапы регрессионного тестирования в правильной последовательности. Время: 4 мин.

Инструкция: Запишите последовательность цифр.

1. Определение набора тестов для повторного прогона
2. Внесение изменений в код
3. Прогон регрессионных тестов
4. Анализ результатов тестирования
5. Принятие решения о необходимости доработок

Ответ:

Задание 4 (ПК 2.2, ОК 4). Расположите этапы работы с удалённым репозиторием Git в правильной последовательности. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. git push
2. git add .
3. git commit -m "сообщение"
4. git pull
5. Создание файлов и внесение изменений
6. git remote add origin <URL>

Ответ:

Задание 5 (ПК 2.1, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какой принцип ООП обеспечивает скрывание внутренней реализации объекта и предоставление доступа только через публичные методы?

1. Наследование
2. Инкапсуляция
3. Полиморфизм
4. Абстракция

Ответ:

Обоснование:

Задание 6 (ПК 2.1, ОК 1). Выберите ОДИН правильный ответ и обоснуйте его. Время: 6 мин.

Инструкция: Какой паттерн проектирования используется для создания объекта без указания конкретного класса, делегируя создание подклассам?

1. Фабричный метод
2. Строитель
3. Абстрактная фабрика
4. Прототип

Ответ:

Обоснование:

Задание 7 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Инструкция: Какие инструменты используются для статического анализа кода в C#? (Выберите 3)

1. SonarQube
2. StyleCop
3. ReSharper
4. Docker
5. Jenkins

Ответ:

Обоснование:

Задание 8 (ПК 2.3, ОК 3). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 7 мин.

Инструкция: Какие из перечисленных практик относятся к методологии Agile? (Выберите 3)

1. Ежедневные стендапы
2. Спринты длительностью 2-4 недели
3. Подробное документирование всех требований перед началом разработки
4. Ретроспективы по итогам спринта
5. Жёсткое следование первоначальному плану

Ответ:

Обоснование:

Задание 9 (ПК 2.5, ОК 5). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите структуру технического задания на разработку модуля «Управление заказами» для desktop-приложения. Включите не менее 5 основных разделов.

Ответ:

Задание 10 (ПК 2.2, ОК 4). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите процесс проведения код-ревью в команде разработчиков на C#.

Ответ:

Задание 11 (ПК 2.4, ОК 7). Анализ кода. Время: 8 мин.

Инструкция: В приведённом фрагменте кода на C# найдите потенциальную ошибку и предложите исправление.

```
csharp
public void Divide(int a, int b)
{
    int result = a / b;
    Console.WriteLine(result);
}
```

Ответ (укажите ошибку и исправленный код):

Задание 12 (ПК 2.2, ОК 8). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию, которая принимает строку и возвращает количество гласных букв в ней.

Ответ (код функции):

Задание 13 (ПК 2.3, ОК 9). Установите соответствие между понятием и его определением в контексте интеграции ПО. Время: 4 мин.

Инструкция: Соотнесите понятие (1–4) с определением (А–Д).

Понятие		Определение	
1.	REST API	А.	Протокол для обмена структурированными сообщениями с использованием XML

2.	SOAP	Б.	Архитектурный стиль, использующий стандартные HTTP-методы для взаимодействия
3.	WebSocket	В.	Протокол для полноценного двустороннего обмена данными между клиентом и сервером
4.	Message Queue	Г.	Промежуточное ПО для асинхронного обмена сообщениями между компонентами
		Д.	Технология для кэширования статических веб-страниц

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 14 (ПК 2.3, ОК 3). Расположите этапы устранения дефекта в правильной последовательности. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. Воспроизведение дефекта
2. Поиск и исправление причины дефекта в коде
3. Написание теста для дефекта
4. Регрессионное тестирование
5. Заккрытие дефекта в трекере

Ответ:

Обоснование:

Задание 15 (ПК 2.5, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какое из перечисленных действий нарушает лицензионное соглашение open-source ПО?

1. Использование библиотеки в коммерческом проекте с указанием автора
2. Модификация библиотеки и распространение без указания изменений
3. Участие в разработке библиотеки через GitHub
4. Изучение исходного кода для обучения

Ответ:

Обоснование:

Задание 16 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Инструкция: Какие из утверждений характеризуют принцип KISS (Keep It Simple, Stupid)? (Выберите 3)

1. Система должна быть максимально простой
2. Сложные решения предпочтительнее простых
3. Избегать ненужной сложности
4. Код должен быть понятным и очевидным
5. Дублирование кода допустимо для повышения производительности

Ответ:

Обоснование:

Задание 17 (ПК 2.4, ОК 5). Открытый вопрос. Время: 8 мин.

Инструкция: Опишите план тестирования модуля «Расчёт стоимости заказа». Включите не менее 4 типов проверок.

Ответ:

Задание 18 (ПК 2.4, ОК 7). Анализ кода. Время: 7 мин.

Инструкция: В приведённом фрагменте кода на C# найдите потенциальную уязвимость и предложите безопасную альтернативу.

csharp

```
string userInput = Console.ReadLine();  
System.Diagnostics.Process.Start("cmd.exe", "/C " + userInput);
```

Ответ (укажите уязвимость и безопасный вариант):

Задание 19 (ПК 2.3, ОК 3). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию, которая принимает список чисел и возвращает их сумму.

Ответ (код функции):

Задание 20 (ПК 2.2, ОК 7). Открытый вопрос. Время: 6 мин.

Инструкция: Опишите, как можно уменьшить энергопотребление при работе с базами данных в приложении.

Ответ:

Вариант 3

Задание 1 (ПК 2.1, ОК 2). Установите соответствие между архитектурными стилями ПО и их описаниями. Время: 4 мин.

Инструкция: Соотнесите этап (1–4) с задачей (А–Д). В правом столбце один лишний вариант.

Стили		Описания	
1.	Монолитная архитектура	А.	Все компоненты приложения объединены в единую программу
2.	Микросервисная архитектура	Б.	Система состоит из небольших независимых сервисов, взаимодействующих через API
3.	Событийно-ориентированная архитектура	В.	Обмен сообщениями между компонентами происходит через события
4.	Клиент-серверная архитектура	Г.	Приложение разделено на клиентскую и серверную части
		Д.	Архитектура, где каждый модуль отвечает за конкретный слой данных

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 2 (ПК 2.5, ОК 9). Установите соответствие между типами тестовой документации и их описаниями. Время: 3 мин.

Инструкция: Соотнесите тип (1–4) с описанием (А–Д).

Тип документации		Описание	
1.	Тест-план	А.	Документ, описывающий цели, объём, подход и расписание тестирования
2.	Тест-кейс	Б.	Пошаговое описание действий для проверки конкретной функции
3.	Чек-лист	В.	Список проверок без подробного описания шагов
4.	Отчёт о дефекте	Г.	Документ, описывающий найденную ошибку с шагами воспроизведения
		Д.	Документ с требованиями к качеству ПО

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 3 (ПК 2.4, ОК 1). Расположите этапы нагрузочного тестирования в правильной последовательности. Время: 4 мин.

Инструкция: Запишите последовательность цифр.

1. Определение целей тестирования
2. Разработка тестовых сценариев
3. Настройка тестового окружения
4. Запуск тестов и сбор метрик
5. Анализ результатов и формирование отчёта

Ответ:

Задание 4 (ПК 2.2, ОК 4). Расположите этапы работы с ветками в Git в правильной последовательности. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. Создание новой ветки: `git branch feature-new`
2. Переключение на ветку: `git checkout feature-new`
3. Внесение изменений и коммиты
4. Возврат на основную ветку: `git checkout main`
5. Слияние веток: `git merge feature-new`
6. Удаление ветки: `git branch -d feature-new`

Ответ:

Задание 5 (ПК 2.1, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какой принцип ООП позволяет создавать новые классы на основе существующих, наследуя их свойства и методы?

1. Инкапсуляция
2. Наследование
3. Полиморфизм
4. Абстракция

Ответ:

Обоснование:

Задание 6 (ПК 2.1, ОК 1). Выберите ОДИН правильный ответ и обоснуйте его. Время: 6 мин.

Инструкция: Какой паттерн проектирования следует использовать, чтобы объект мог уведомлять другие объекты об изменениях своего состояния?

1. Фабричный метод
2. Строитель
3. Наблюдатель (Observer)
4. Адаптер

Ответ:

Обоснование:

Задание 7 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Инструкция: Какие инструменты используются для сборки и развёртывания C#-приложений? (Выберите 3)

1. MSBuild
2. Docker
3. Kubernetes
4. Git
5. Jira

Ответ:

Обоснование:

Задание 8 (ПК 2.3, ОК 3). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 7 мин.

Инструкция: Какие из перечисленных практик относятся к управлению требованиями в Agile? (Выберите 3)

1. User Stories (Пользовательские истории)
2. Product Backlog
3. Use Cases (Варианты использования)
4. Жёсткое фиксирование требований в начале проекта
5. Отсутствие приоритизации требований

Ответ:

Обоснование:

Задание 9 (ПК 2.5, ОК 5). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите структуру технического задания на разработку модуля «Отчётность» для desktop-приложения. Включите не менее 5 основных разделов.

Ответ:

Задание 10 (ПК 2.2, ОК 4). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите процесс работы с техническим долгом в команде разработчиков.

Ответ:

Задание 11 (ПК 2.4, ОК 7). Анализ кода. Время: 8 мин.

Инструкция: В приведённом фрагменте кода на C# найдите потенциальную проблему и предложите исправление.

```
csharp
public void ReadFile(string path)
{
    StreamReader reader = new StreamReader(path);
    string content = reader.ReadToEnd();
    Console.WriteLine(content);
}
```

Ответ (укажите ошибку и исправленный код):

Задание 12 (ПК 2.2, ОК 8). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию, которая принимает строку и возвращает true, если в ней есть повторяющиеся символы, и false — в противном случае.

Ответ (код функции):

Задание 13 (ПК 2.3, ОК 9). Установите соответствие между понятием в области тестирования ПО и его определением. Время: 4 мин.

Инструкция: Соотнесите понятие (1–4) с определением (А–Д).

Понятия		Определение	
1.	Регрессионное тестирование	А.	Быстрая проверка основных функций системы после изменений
2.	Санитарное тестирование (Sanity Testing)	Б.	Тестирование, направленное на проверку того, что исправление дефектов не повлияло на работоспособность других функций
3.	Дымовое тестирование (Smoke Testing)	В.	Проверка производительности системы под нагрузкой
4.	Нагрузочное тестирование	Г.	Углублённое тестирование конкретной функции после её модификации
		Д.	Тестирование интерфейса пользователя на соответствие стандартам

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 14 (ПК 2.3, ОК 3). Расположите этапы миграции базы данных в правильной последовательности. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. Анализ текущей структуры БД
2. Разработка скриптов миграции
3. Резервное копирование данных
4. Тестирование миграции на тестовом окружении
5. Применение миграции на рабочей БД

Ответ:

Обоснование:

Задание 15 (ПК 2.5, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какое из перечисленных действий является обязательным при обработке персональных данных в соответствии с законодательством?

1. Хранение данных в открытом виде для удобства
2. Получение согласия субъекта данных
3. Передача данных третьим лицам без уведомления
4. Отсутствие журналирования доступа к данным

Ответ:

Обоснование:

Задание 16 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Инструкция: Какие из утверждений характеризуют принцип YAGNI (You Ain't Gonna Need It)? (Выберите 3)

1. Не добавлять функциональность, пока она не понадобится
2. Реализовывать все возможные функции на будущее
3. Избегать преждевременной оптимизации
4. Сосредоточиться на текущих требованиях
5. Создавать сложные архитектуры для масштабирования

Ответ:

Обоснование:

Задание 17 (ПК 2.4, ОК 5). Открытый вопрос. Время: 8 мин.

Инструкция: Опишите план тестирования модуля «Импорт данных из CSV». Включите не менее 4 типов проверок.

Ответ:

Задание 18 (ПК 2.4, ОК 7). Анализ кода. Время: 7 мин.

Инструкция: В приведённом фрагменте кода на C# найдите потенциальную уязвимость и предложите безопасную альтернативу.

```
csharp
string fileName = Console.ReadLine();
File.WriteAllText(fileName, "Данные");
```

Ответ (укажите уязвимость и безопасный вариант):

Задание 19 (ПК 2.3, ОК 3). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию для вычисления факториала числа.

Ответ (код функции):

Задание 20 (ПК 2.2, ОК 7). Открытый вопрос. Время: 6 мин.

Инструкция: Опишите, как принципы алгоритмической эффективности могут снизить энергопотребление ПО.

Ответ:

2.3.Вариант 4

Задание 1 (ПК 2.1, ОК 2). Установите соответствие между методологиями управления проектами и их характеристиками. Время: 4 мин.

Инструкция: Соотнесите этап (1–4) с задачей (А–Д). В правом столбце один лишний вариант.

Стили		Описания	
1.	Waterfall	А.	Последовательная модель с чёткими этапами
2.	Scrum	Б.	Итеративная разработка с фиксированными спринтами
3.	Kanban	В.	Визуализация рабочего процесса с использованием канбан-досок
4.	Lean	Г.	Устранение потерь и оптимизация потока создания ценности
		Д.	Модель, основанная на формальных верификации и валидации

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 2 (ПК 2.5, ОК 9). Установите соответствие между форматами данных и их описаниями. Время: 3 мин.

Инструкция: Соотнесите тип (1–4) с описанием (А–Д).

Форматы		Описания	
1.	JSON	А.	Текстовый формат, использующий отступы для структурирования
2.	XML	Б.	Формат на основе тегов для представления иерархических данных
3.	YAML	В.	Формат для хранения табличных данных, разделённых запятыми
4.	CSV	Г.	Легковесный формат обмена данными на основе пар ключ-значение
		Д.	Бинарный формат для эффективной сериализации

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 3 (ПК 2.4, ОК 1). Расположите этапы тестирования безопасности в правильной последовательности. Время: 4 мин.

Инструкция: Запишите последовательность цифр.

1. Анализ угроз и определение границ тестирования
2. Сканирование уязвимостей автоматизированными инструментами
3. Ручное тестирование на проникновение
4. Анализ результатов и составление отчёта
5. Устранение найденных уязвимостей

Ответ:

Задание 4 (ПК 2.2, ОК 4). Расположите действия при откате изменений в Git в правильной последовательности. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. Определение коммита для отката
2. Выполнение команды git revert <hash>
3. Разрешение возможных конфликтов
4. Создание нового коммита отката
5. Пуш изменений в удалённый репозиторий

Ответ:

Задание 5 (ПК 2.1, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какой принцип ООП позволяет скрыть сложность системы, предоставляя только необходимые детали?

1. Инкапсуляция
2. Наследование
3. Полиморфизм
4. Абстракция

Ответ:

Обоснование:

Задание 6 (ПК 2.1, ОК 1). Выберите ОДИН правильный ответ и обоснуйте его. Время: 6 мин.

Инструкция: Какой паттерн проектирования следует использовать для обеспечения единственной точки доступа к общему ресурсу?

1. Фабричный метод
2. Строитель
3. Одиночка (Singleton)
4. Адаптер

Ответ:

Обоснование:

Задание 7 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Инструкция: Какие инструменты используются для профилирования и отладки C#-приложений? (Выберите 3)

1. Visual Studio Debugger
2. dotMemory
3. dotTrace
4. Git
5. Jenkins

Ответ:

Обоснование:

Задание 8 (ПК 2.3, ОК 3). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 7 мин.

Инструкция: Какие из перечисленных практик относятся к обеспечению качества кода? (Выберите 3)

1. Code Review
2. Pair Programming
3. Static Code Analysis
4. Отсутствие тестирования
5. Игнорирование стандартов кодирования

Ответ:

Обоснование:

Задание 9 (ПК 2.5, ОК 5). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите структуру технического задания на разработку модуля «Настройки приложения». Включите не менее 5 основных разделов.

Ответ:

Задание 10 (ПК 2.2, ОК 4). Открытый вопрос. Время: 10 мин.

Инструкция: Опишите процесс внедрения новых технологий в команде разработчиков.

Ответ:

Задание 11 (ПК 2.4, ОК 7). Анализ кода. Время: 8 мин.

Инструкция: В приведённом фрагменте кода на C# найдите потенциальную проблему производительности и предложите оптимизацию.

```
csharp
public string ConcatenateStrings(List<string> strings)
{
    string result = "";
    foreach (string s in strings)
    {
        result += s;
    }
    return result;
}
```

Ответ (укажите ошибку и исправленный код):

Задание 12 (ПК 2.2, ОК 8). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию для сортировки списка чисел по возрастанию без использования встроенных методов сортировки.

Ответ (код функции):

Задание 13 (ПК 2.3, ОК 9). Установите соответствие между понятием в области паттернов интеграции и его описанием. Время: 4 мин.

Инструкция: Соотнесите понятие (1–4) с определением (А–Д).

Понятия		Определение	
1.	Шина сообщений (Message Bus)	А.	Паттерн, который преобразует интерфейс одного класса в интерфейс, ожидаемый клиентом
2.	Шаблон "Адаптер" (Adapter)	Б.	Паттерн, который предоставляет унифицированный интерфейс к набору интерфейсов в подсистеме
3.	Шаблон "Фасад" (Facade)	В.	Паттерн, который разделяет модели для обновления (команды) и чтения (запросы) данных
4.	Командный запрос (CQRS — Command	Г.	Паттерн, который обеспечивает взаимодействие между различными

	Query Responsibility Segregation)		компонентами системы через общий канал обмена сообщениями
		Д.	Паттерн, который создаёт клон объекта для независимого изменения

Ответ:

Запишите выбранные буквы под соответствующими цифрами

1	2	3	4

Задание 14 (ПК 2.3, ОК 3). Расположите этапы резервного копирования и восстановления данных в правильной последовательности. Время: 5 мин.

Инструкция: Запишите последовательность цифр.

1. Определение критических данных и частоты бэкапов
2. Выбор инструментов и настройка расписания
3. Выполнение резервного копирования
4. Проверка целостности бэкапов
5. Восстановление данных при необходимости

Ответ:

Обоснование:

Задание 15 (ПК 2.5, ОК 6). Выберите ОДИН правильный ответ и обоснуйте его. Время: 5 мин.

Инструкция: Какое действие является корректным при использовании сторонней библиотеки с лицензией MIT?

1. Изменение кода библиотеки без указания авторства
2. Использование в коммерческом проекте с указанием лицензии
3. Распространение изменённой версии без лицензии
4. Соккрытие факта использования библиотеки

Ответ:

Обоснование:

Задание 16 (ПК 2.2, ОК 2). Выберите ВСЕ правильные ответы и обоснуйте их. Время: 6 мин.

Какие из утверждений характеризуют принцип SOC (Separation of Concerns)? (Выберите 3)

1. Разделение системы на отдельные модули
2. Каждый модуль решает свою задачу
3. Изменение одного модуля не должно влиять на другие
4. Объединение всей логики в одном классе
5. Отказ от использования интерфейсов

Ответ:

Обоснование:

Задание 17 (ПК 2.4, ОК 5). Открытый вопрос. Время: 8 мин.

Инструкция: Опишите план тестирования модуля «Экспорт данных в Excel». Включите не менее 4 типов проверок.

Ответ:

Задание 18 (ПК 2.4, ОК 7). Анализ кода. Время: 7 мин.

Инструкция: В приведённом фрагменте кода на C# найдите потенциальную проблему многопоточности и предложите решение.

```
csharp
private int _counter = 0;

public void Increment()
{
    _counter++;
}
```

Ответ (укажите уязвимость и безопасный вариант):

Задание 19 (ПК 2.3, ОК 3). Написание кода. Время: 8 мин.

Инструкция: Напишите на C# функцию для проверки, является ли число простым.

Ответ (код функции):

Задание 20 (ПК 2.2, ОК 7). Открытый вопрос. Время: 6 мин.

Инструкция: Опишите, как использование виртуализации может снизить энергопотребление серверов.

Ответ:

3. Ключи к оцениванию тестовых заданий

ВАРИАНТ 1

Задание 1.

Ответ:

1	2	3	4
Б	В	А	Г

Задание 2.

Ответ:

1	2	3	4
Б	В	А	Г

Задание 3.

Ответ: 3-1-4-2-5

Задание 4.

Ответ: 6-1-2-3-4-5

Задание 5.

Ответ: 3

Обоснование: Полиморфизм позволяет использовать общий интерфейс для разных реализаций, что обеспечивает взаимозаменяемость модулей.

Задание 6.

Ответ: 2

Обоснование: Паттерн Builder позволяет конструировать сложные объекты шаг за шагом, изолируя процесс создания от представления объекта.

Задание 7.

Ответ: 2, 3, 5

Обоснование: Git — система контроля версий, GitHub/GitLab — платформы для хранения репозитория, Visual Studio — IDE с поддержкой Git.

Задание 8.

Ответ: 1, 3, 5

Обоснование: CI, мониторинг и IaC являются ключевыми практиками DevOps, направленными на автоматизацию и контроль.

Задание 9.

Пример ответа (структура ТЗ):

1. Введение (Общие сведения)

- Наименование модуля: «Управление пользователями».
- Назначение модуля: обеспечение регистрации, аутентификации, авторизации и управления учетными записями пользователей в рамках desktop-приложения.
- Краткое описание: модуль должен предоставлять интерфейс для администратора и функции самообслуживания для пользователей.

2. Функциональные требования

- **Регистрация нового пользователя:** Ввод логина, пароля (с подтверждением), ФИО, электронной почты. Валидация данных.
- **Аутентификация (Вход в систему):** Проверка логина и пароля. Обработка неверных попыток входа.
- **Управление ролями:** Система ролей (например, «Администратор», «Пользователь», «Модератор»). Назначение и изменение ролей пользователям через административный интерфейс.
- **Просмотр и редактирование профиля:** Возможность для пользователя изменять свои данные (пароль, email).
- **Блокировка/активация учетной записи:** Функция для администратора.
- **Выход из системы.**

3. Технические требования

- **Язык и платформа:** C#, .NET Framework 4.8 / .NET 6+.
- **Технология UI:** Windows Presentation Foundation (WPF) с паттерном MVVM (или Windows Forms).
- **Логика и данные:**
- Логика модуля должна быть выделена в отдельный уровень (класс службы — UserService).
- Для хранения данных использовать локальную базу данных (например, SQLite) или файловое хранилище (JSON, XML). Указать структуру таблицы Users (Id, Login, PasswordHash, Email, FullName, RoleId, IsActive).

- **Безопасность:** Пароли должны храниться в виде хеша (например, с помощью BCrypt.Net или PBKDF2).

4. Требования к пользовательскому интерфейсу (UI)

Основные формы:

- Форма входа (LoginView): поля для логина и пароля, кнопки «Войти» и «Регистрация».
- Форма регистрации (RegisterView).
- Главная форма приложения с меню, меняющимся в зависимости от роли пользователя.
- Административная панель (AdminUserManagerView): таблица (DataGrid) со списком пользователей, кнопки для редактирования, блокировки, смены роли.

Стиль и оформление: Соответствие общему стилю приложения (цветовая схема, шрифты). Интерфейс должен быть интуитивно понятным.

5. Требования к документации и тестированию

Кодовая документация: XML-комментарии для публичных методов классов служб и репозиториев.

Руководство пользователя: Краткое описание процесса регистрации и входа.

Тестирование: Необходимо покрыть модуль unit-тестами (например, с использованием NUnit или xUnit) для проверки логики UserService (регистрация, вход, валидация).

Задание 10.

Ответ:

1. Создание ветки feature/auth от main
2. Реализация функции в C# с коммитами
3. Пуш ветки в удаленный репозиторий
4. Создание Pull Request (Merge Request)
5. Проведение код-ревью, обсуждение изменений
6. Исправление замечаний, повторный пуш
7. Слияние через merge после аппрува
8. Удаление ветки

Задание 11.

Ошибка: целочисленное деление, потеря дробной части.

Исправление: привести total к double:

```
csharp
double average = (double)total / numbers.Count;
```

Задание 12.

Ответ:

csharp

```
public bool IsPalindrome(string text)
{
    string cleaned = new string(text.ToLower().Where(c => !char.IsWhiteSpace(c)).ToArray());
    return cleaned.SequenceEqual(cleaned.Reverse());
}
```

Задание 13.

Ответ:

1	2	3	4
Б	А	В	Г

Задание 14.

Ответ: 1-3-2-4-5

Задание 15.

Ответ: 3

Обоснование: Копирование кода без соблюдения лицензии и указания авторства нарушает интеллектуальные права и этические нормы.

Задание 16.

Ответ: 1, 3, 5

Обоснование: DRY направлен на устранение дублирования и централизацию логики.

Задание 17.

Ответ:

1. Позитивные тесты (корректные логин/пароль)
2. Негативные тесты (неверные данные)
3. Тесты безопасности (SQL-инъекция, XSS)
4. Тесты производительности (время ответа)
5. Интеграционные тесты (работа с БД, сессиями)

Задание 18.

Ответ:

Уязвимость: SQL-инъекция.

Безопасный вариант: использовать параметризованные запросы:

csharp

```
string query = "SELECT * FROM Users WHERE Login = @Login AND Password = @Password";
using (var cmd = new SqlCommand(query, connection))
{
    cmd.Parameters.AddWithValue("@Login", login);
    cmd.Parameters.AddWithValue("@Password", password);
}
```

Задание 19.

Ответ:

csharp

```
public List<int> GetEvenNumbers(List<int> numbers)
{
    return numbers.Where(n => n % 2 == 0).ToList();
}
```

Задание 20.

Ответ:

1. Оптимизация запросов к БД для снижения нагрузки на сервер.
2. Использование асинхронного программирования для эффективного использования ресурсов.
3. Минимизация использования памяти и процессора за счет алгоритмических оптимизаций.

ВАРИАНТ 2**Задание 1.**

Ответ:

1	2	3	4
А	Б	Г	В

Задание 2.

Ответ:

1	2	3	4
А	Б	В	Г

Задание 3.

Ответ: 2-1-3-4-5

Задание 4.

Ответ: 6-5-2-3-4-1

Задание 5.

Ответ: 2

Обоснование: Инкапсуляция позволяет скрыть детали реализации и защитить данные объекта от несанкционированного доступа.

Задание 6.

Ответ: 1

Обоснование: Фабричный метод определяет интерфейс для создания объекта, но оставляет выбор конкретного класса создаваемого объекта за подклассами.

Задание 7.

Ответ: 1, 2, 3

Обоснование: SonarQube, StyleCop и ReSharper являются инструментами для анализа качества кода, проверки стиля и рефакторинга.

Задание 8.

Ответ: 1, 2, 4

Обоснование: Ежедневные стендапы, спринты и ретроспективы — ключевые практики Agile.

Задание 9.

Ответ:

1. Общие сведения (назначение модуля)

2. Функциональные требования (создание заказа, изменение статуса, просмотр истории)
3. Технические требования (C#, WPF, SQLite)
4. Требования к интерфейсу (форма списка заказов, форма создания заказа)
5. Требования к безопасности (разграничение прав доступа)
6. План тестирования

Задание 10.

Ответ:

1. Создание пул-реквеста
2. Назначение ревьюеров
3. Проверка кода на соответствие стандартам, наличие тестов, ошибки
4. Комментарии и предложения по улучшению
5. Исправление кода разработчиком
6. Повторное ревью
7. Слияние после утверждения

Задание 11.

Ошибка: возможное деление на ноль.

Исправление: добавить проверку делителя:

```
csharp
public void Divide(int a, int b)
{
    if (b == 0)
        throw new ArgumentException("Делитель не может быть равен нулю");
    int result = a / b;
    Console.WriteLine(result);
}
```

Задание 12.

Ответ:

```
csharp
public int CountVowels(string text)
{
    char[] vowels = { 'a', 'e', 'i', 'o', 'u', 'a', 'e', 'ë', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я' };
    return text.ToLower().Count(c => vowels.Contains(c));
}
```

Задание 13.

Ответ:

1	2	3	4
Б	А	В	Г

Задание 14.

Ответ: 1-3-2-4-5

Задание 15.

Ответ: 2

Обоснование: Лицензии open-source часто требуют сохранения информации об авторстве и открытого распространения изменений.

Задание 16.

Ответ: 1, 3, 4

Обоснование: KISS направлен на упрощение системы и избегание ненужной сложности.

Задание 17.

Ответ:

1. Функциональные тесты (правильность расчёта)
2. Тесты граничных значений (нулевое количество, отрицательные числа)
3. Интеграционные тесты (работа с модулем скидок)
4. Тесты производительности (скорость расчёта при большом количестве позиций)

Задание 18.

Ответ:

Уязвимость: выполнение произвольных команд.

Безопасный вариант: валидация ввода или использование безопасных API.

Задание 19.

Ответ:

```
csharp
public int Sum(List<int> numbers)
{
    return numbers.Sum();
}
```

Задание 20.

Ответ:

1. Оптимизация запросов для уменьшения времени выполнения
2. Использование кэширования результатов запросов
3. Пакетная обработка операций вместо множества мелких запросов

ВАРИАНТ 3**Задание 1.**

Ответ:

1	2	3	4
А	Б	В	Г

Задание 2.

Ответ:

1	2	3	4
А	Б	В	Г

Задание 3.

Ответ: 1-2-3-4-5

Задание 4.

Ответ: 1-2-3-4-5-6

Задание 5.

Ответ: 2

Обоснование: Наследование позволяет классу-потомку использовать поля и методы класса-родителя, что способствует повторному использованию кода.

Задание 6.

Ответ: 3

Обоснование: Паттерн «Наблюдатель» определяет зависимость «один-ко-многим» между объектами так, что при изменении состояния одного объекта все зависящие от него объекты получают уведомление.

Задание 7.

Ответ: 1, 2, 3

Обоснование: MSBuild — система сборки, Docker — контейнеризация, Kubernetes — оркестрация контейнеров для развёртывания.

Задание 8.

Ответ: 1, 2, 3

Обоснование: User Stories, Backlog и Use Cases — основные инструменты работы с требованиями в Agile.

Задание 9.

Ответ:

1. Введение (назначение модуля, виды отчётов)

2. Функциональные требования (формирование отчётов, фильтрация, экспорт)
3. Технические требования (C#, WPF, библиотеки для генерации PDF/Excel)
4. Требования к интерфейсу (макет формы отчётов, настройки параметров)
5. Требования к производительности (время генерации отчётов)
6. План тестирования (проверка корректности данных, форматов экспорта)

Задание 10.

Ответ:

1. Выявление и документирование технического долга
2. Оценка влияния и приоритизация
3. Включение задач по устранению долга в бэклог спринта
4. Реализация рефакторинга и улучшений
5. Проверка результатов и обновление документации

Задание 11.

Ошибка: ресурс StreamReader не освобождается.

Исправление: использование блока using для автоматического освобождения.

```
public void ReadFile(string path)
{
    using (StreamReader reader = new StreamReader(path))
    {
        string content = reader.ReadToEnd();
        Console.WriteLine(content);
    }
}
```

Задание 12.

Ответ:

```
csharp
public bool HasDuplicateChars(string text)
{
    return text.Length != text.Distinct().Count();
}
```

Задание 13.

Ответ:

1	2	3	4
Б	Г	А	В

Задание 14.

Ответ: 1-2-3-4-5

Задание 15.

Ответ: 2

Обоснование: Законодательство (например, 152-ФЗ) требует получения явного согласия на обработку персональных данных.

Задание 16.

Ответ: 1, 3, 4

Обоснование: YAGNI направлен на отказ от избыточной функциональности, которая может никогда не понадобиться.

Задание 17.

Ответ:

1. Проверка корректности чтения данных
2. Проверка обработки ошибок (неверный формат, отсутствие файла)
3. Проверка валидации данных (типы, диапазоны)
4. Проверка производительности при большом объеме данных
5. Интеграционные тесты (сохранение в БД)

Задание 18.

Ответ:

Уязвимость: возможность записи в произвольный файл (Path Traversal).

Безопасный вариант: валидация имени файла:

csharp

```
string fileName = Console.ReadLine();
if (!Path.GetFileName(fileName).Equals(fileName))
    throw new ArgumentException("Недопустимое имя файла");
File.WriteAllText(fileName, "Данные");
```

Задание 19.

Ответ:

csharp

```
public int Factorial(int n)
{
    if (n <= 1) return 1;
    return n * Factorial(n - 1);
}
```

Задание 20.

Ответ:

1. Использование оптимальных алгоритмов для уменьшения времени выполнения
2. Минимизация операций ввода-вывода
3. Снижение сложности алгоритмов для уменьшения нагрузки на процессор

ВАРИАНТ 4

Задание 1.

Ответ:

1	2	3	4
А	Б	В	Г

Задание 2.

Ответ:

1	2	3	4
Г	Б	А	В

Задание 3.

Ответ: 1-2-3-4-5

Задание 4.

Ответ: 1-2-3-4-5-

Задание 5.

Ответ: 4

Обоснование: Абстракция позволяет выделять только существенные характеристики объекта, скрывая детали реализации.

Задание 6.

Ответ: 3

Обоснование: Паттерн «Одиночка» гарантирует, что у класса есть только один экземпляр, и предоставляет глобальную точку доступа к нему.

Задание 7.

Ответ: 1, 2, 3

Обоснование: Visual Studio Debugger — отладка, dotMemory — анализ памяти, dotTrace — профилирование производительности.

Задание 8.

Ответ: 1, 2, 3

Обоснование: Code Review, Pair Programming и Static Analysis — практики, направленные на повышение качества кода.

Задание 9.

Ответ:

1. Введение (назначение модуля, типы настроек)
2. Функциональные требования (хранение настроек, интерфейс изменения)
3. Технические требования (C#, WPF, сохранение в JSON/реестр)
4. Требования к интерфейсу (форма настроек, категории, валидация)
5. Требования к безопасности (шифрование чувствительных настроек)
6. План тестирования (проверка сохранения, загрузки, сброса настроек)

Задание 10.

Ответ:

1. Анализ потребностей и выбор технологии
2. Оценка рисков и затрат
3. Обучение команды (воркшопы, документация)
4. Пилотный проект на новой технологии
5. Анализ результатов и принятие решения о внедрении
6. Постепенный переход и поддержка

Задание 11.

Проблема: многократное создание строк (immutable).

Оптимизация: использование StringBuilder.

```
csharp
public string ConcatenateStrings(List<string> strings)
{
    StringBuilder sb = new StringBuilder();
    foreach (string s in strings)
    {
        sb.Append(s);
    }
    return sb.ToString();
}
```

Задание 12.

Ответ:

```
csharp
public List<int> BubbleSort(List<int> numbers)
{
    // ...
}
```

```
var sorted = new List<int>(numbers);
for (int i = 0; i < sorted.Count - 1; i++)
{
    for (int j = 0; j < sorted.Count - i - 1; j++)
    {
        if (sorted[j] > sorted[j + 1])
        {
            int temp = sorted[j];
            sorted[j] = sorted[j + 1];
            sorted[j + 1] = temp;
        }
    }
}
return sorted;
}
```

Задание 13.

Ответ:

1	2	3	4
Г	А	Б	В

Задание 14.

Ответ: 1-2-3-4-5

Задание 15.

Ответ: 2

Обоснование: Лицензия MIT разрешает коммерческое использование при условии сохранения уведомления о лицензии.

Задание 16.

Ответ: 1, 2, 3

Обоснование: SOC предполагает разделение системы на независимые модули, каждый из которых отвечает за свою задачу.

Задание 17.

Ответ:

1. Проверка корректности данных в экспортированном файле
2. Проверка форматов ячеек (даты, числа, валюты)
3. Проверка работы с большими объёмами данных
4. Проверка обработки ошибок (отсутствие прав на запись, нехватка памяти)
5. Интеграционные тесты (связь с модулем отчётности)

Задание 18.

Ответ:

Проблема: состояние гонки (race condition) при многопоточном доступе.

Решение: использование Interlocked или блокировки.

```
csharp
private int _counter = 0;
private object _lock = new object();

public void Increment()
{
    lock (_lock)
    {
        _counter++;
    }
}
```

Задание 19.

Ответ:

```
csharp
public bool IsPrime(int n)
{
    if (n <= 1) return false;
    for (int i = 2; i <= Math.Sqrt(n); i++)
    {
        if (n % i == 0) return false;
    }
    return true;
}
```

Задание 20.

Ответ:

1. Консолидация нескольких серверов на одном физическом оборудовании
2. Динамическое распределение ресурсов в зависимости от нагрузки
3. Возможность отключения неиспользуемых виртуальных машин