МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ
ГАПОУ СО
«НИЖНЕТАГИЛЬСКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ»

Утверждаю:
Зам. директора по УР
_____ М.Г. Дмитриева
«30» июня 2026 г.

КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ

«Основы алгоритмизации и программирования»

(ОП.04)

по специальности
09.02.11 Разработка и управление программным обеспечением

Нижний Тагил 2026

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

СОДЕРЖАНИЕ

5.1 ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ.....4
5.1.1 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....4
5.1.2 ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....6
5.1.3 ОПИСАНИЕ ПРОЦЕДУРЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....6
5.1.4 КРИТЕРИИ ОЦЕНКИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.....8
5.2 КОМПЛЕКТ «ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ».....11
5.2.1 Вопросы к экзамену.....11
5.2.2 Типовые практические задания.....14
5.2.3 Задания для экзамена.....15
5.3 КОМПЛЕКТ «ТЕКУЩИЙ КОНТРОЛЬ».....32
5.3.1 Задания для срезов знаний.....32
5.3.2 Тестовые задания.....32
5.3.3 Задания для проверочных, самостоятельных и контрольных работ.....39
5.3.4 Лабораторные и практические работы.....51
5.3.5 Компьютерные программы для контроля знаний студентов.....54

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

5.1 ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

5.1.1 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Предметом оценки освоения учебной дисциплины (УД) являются умения и знания. Контроль и оценка этих дидактических единиц осуществляются с использованием следующих форм и методов:

Таблица 1 – Формы и методы контроля и оценки дидактических единиц

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения	Наименование оценочного средства
Уметь:		
– разрабатывать и анализировать алгоритмы для решения поставленных задач	Оценка выполнения лабораторных работ. Контрольная работа. Экзамен.	ЛР №4–6, 15; КР №1; тесты 1–3; билеты
– определять сложность алгоритмов	Оценка выполнения лабораторных работ. Контрольная работа. Экзамен.	ЛР №7, 8, 16–18; КР №1; тесты 4–7, 13; билеты
– реализовывать типовые алгоритмы (поиск, сортировка, обработка стеков, очередей, списков, деревьев и графов) в виде программ на языке PHP	Оценка выполнения лабораторных работ. Контрольная работа. Экзамен.	ЛР №16–22; КР №2; тесты 6–14; билеты
– использовать средства проектирования для создания и графического отображения алгоритмов (блок-схемы по ГОСТ 19.701-90)	Наблюдение и оценка выполнения лабораторных работ. Контрольная работа.	ЛР №4–6; КР №1; билеты
– оформлять код программ в соответствии со стандартом кодирования (PSR-12, принципы чистого кода)	Оценка выполнения лабораторных работ. Контрольная работа. Тестирование.	ЛР №27, 28; КР №3; тесты 20–22
– выполнять проверку и отладку кода программы, разрабатывать модульные тесты	Оценка выполнения лабораторных работ. Контрольная работа.	ЛР №25, 26; КР №3; тест 15
– использовать систему контроля версий Git и платформу GitVerse при разработке программ, оформлять запросы на слияние и выполнять код-ревью	Оценка выполнения лабораторных работ. Код-ревью запросов на слияние.	ЛР №2, 3; тесты 31, 32 и Moodle 0.1–0.5; билеты №3–5
– разрабатывать серверные веб-приложения на PHP и Laravel с применением ООП и Composer, читать техническую документацию на русском и английском	Оценка выполнения лабораторных работ. Контрольная работа. Экзамен.	ЛР №29–34; КР №4; билеты

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения	Наименование оценочного средства
языках		
– использовать российские ИИ-ассистенты (GigaChat, GigaCode, Алиса AI) для генерации, ревью и рефакторинга кода PHP и составления тестов с обязательной проверкой результата и соблюдением требований информационной безопасности	Оценка выполнения лабораторных работ. Тестирование. Экзамен.	ЛР №35, 36; тесты 26–30; билеты №9, 22, 23, 24
Знать:		
– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции	Тестирование. Контрольная работа. Экзамен.	Тесты 1–3; КР №1; билеты
– классификацию языков программирования, понятие системы программирования, жизненный цикл программы	Тестирование. Экзамен.	Тесты Moodle 1.1; билеты №1, 2
– основные элементы языка PHP, структуру программы, операторы и операции, управляющие структуры, классы памяти (область видимости и время жизни переменных)	Тестирование. Контрольная работа. Экзамен.	Тесты 16–19 и Moodle 1.2–1.11; КР №3; билеты
– методы реализации типовых алгоритмов, асимптотическую оценку сложности алгоритмов	Тестирование. Контрольные работы. Экзамен.	Тесты 4–7, 13–14; КР №1, 2; билеты
– структуры данных (массив, множество, запись, стек, очередь, связный список, хеш-таблица, дерево, граф) и области их применения	Тестирование. Контрольная работа. Экзамен.	Тесты 8–12; КР №2; билеты
– понятие подпрограммы, рекурсию, библиотеки подпрограмм, модули, пространства имён, менеджер зависимостей Composer, стандарты PSR	Тестирование. Контрольная работа. Экзамен.	Тесты 15, 19–22; КР №3; билеты
– объектно-ориентированную модель программирования, основные принципы ООП на примере PHP: понятия классов и объектов, их свойств и методов, инкапсуляцию,	Тестирование. Контрольная работа. Экзамен.	Тесты 33, 34 и Moodle 1.7; КР №3; билеты

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения	Наименование оценочного средства
наследование, полиморфизм и переопределение методов; принципы SOLID		
– основы работы с Git и GitVerse; общую архитектуру фреймворка Laravel (маршрутизация, контроллеры, Eloquent ORM, контейнер внедрения зависимостей)	Тестирование. Контрольная работа. Экзамен.	Тесты 23–25, 31, 32 и Moodle 0.1–0.5, 3.1–3.12; КР №4; билеты
– возможности, ограничения и риски применения ИИ-ассистентов при разработке; требования к защите персональных данных (152-ФЗ), конфиденциальной информации и учётных данных при работе с ними	Тестирование. Экзамен.	Тесты 26–30; билеты №9, 22, 23, 24

Оценка освоения УД предусматривает использование пятибалльной системы оценки.

5.1.2 ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Таблица 2 – Запланированные формы промежуточной аттестации

№ семестра	Формы промежуточной аттестации	Форма проведения
6	Экзамен	Устный ответ по билету и выполнение практического задания

5.1.3 ОПИСАНИЕ ПРОЦЕДУРЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Экзамен по дисциплине проводится в 6 семестре и устанавливает уровень сформированности умений и усвоения знаний по всему курсу. К экзамену допускаются обучающиеся, выполнившие и защитившие все лабораторные работы и получившие положительные оценки за контрольные работы. Билет содержит два теоретических вопроса (по темам разных разделов курса) и практическое задание (трассировка алгоритма, оценка сложности, разбор или доработка фрагмента кода на PHP, разбор результата работы ИИ-ассистента). Всего в банке 24 билета. Использование ИИ-ассистентов и посторонних материалов на экзамене не допускается.

В результате освоения дисциплины студент должен уметь:

– разрабатывать и анализировать алгоритмы для решения поставленных задач;

– определять сложность алгоритмов;

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

– реализовывать типовые алгоритмы (поиск, сортировка, обработка стеков, очередей, списков, деревьев и графов) в виде программ на языке PHP;

– использовать средства проектирования для создания и графического отображения алгоритмов (блок-схемы по ГОСТ 19.701-90);

– оформлять код программ в соответствии со стандартом кодирования (PSR-12, принципы чистого кода);

– выполнять проверку и отладку кода программы, разрабатывать модульные тесты;

– использовать систему контроля версий Git и платформу GitVerse при разработке программ, оформлять запросы на слияние и выполнять код-ревью;

– разрабатывать серверные веб-приложения на PHP и Laravel с применением ООП и Composer, читать техническую документацию на русском и английском языках;

– использовать российские ИИ-ассистенты (GigaChat, GigaCode, Алиса AI) для генерации, ревью и рефакторинга кода PHP и составления тестов с обязательной проверкой результата и соблюдением требований информационной безопасности.

В результате освоения дисциплины студент должен знать:

– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;

– классификацию языков программирования, понятие системы программирования, жизненный цикл программы;

– основные элементы языка PHP, структуру программы, операторы и операции, управляющие структуры, классы памяти (область видимости и время жизни переменных);

– методы реализации типовых алгоритмов, асимптотическую оценку сложности алгоритмов;

– структуры данных (массив, множество, запись, стек, очередь, связный список, хеш-таблица, дерево, граф) и области их применения;

– понятие подпрограммы, рекурсию, библиотеки подпрограмм, модули, пространства имён, менеджер зависимостей Composer, стандарты PSR;

– объектно-ориентированную модель программирования, основные принципы ООП на примере PHP: понятия классов и объектов, их свойств и методов, инкапсуляцию, наследование, полиморфизм и переопределение методов; принципы SOLID;

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

– основы работы с Git и GitVerse; общую архитектуру фреймворка Laravel (маршрутизация, контроллеры, Eloquent ORM, контейнер внедрения зависимостей);

– возможности, ограничения и риски применения ИИ-ассистентов при разработке; требования к защите персональных данных (152-ФЗ), конфиденциальной информации и учётных данных при работе с ними.

Количество заданий для студента: 3 (два теоретических вопроса и одно практическое задание)

Время выполнения каждого задания и максимальное время на проведение аттестации:

Вопрос 1 – 10 мин.

Вопрос 2 – 10 мин.

Практическое задание – 15 мин.

Всего на экзамен (с подготовкой) – 45 мин. на одного обучающегося.

Условия выполнения заданий

Помещение: лаборатория «Программного обеспечения и сопровождения компьютерных систем» (кабинет программирования)

Требования охраны труда: инструктаж по технике безопасности при работе с вычислительной техникой и электрооборудованием.

Оборудование:

– компьютеры по числу экзаменуемых с установленными PHP 8.5 (CLI) и VS Code (для практических заданий, требующих запуска кода);

– проектор;

– билеты на бумажном и электронном носителях.

Перечень справочной и нормативной литературы, разрешённой для использования:

1. Официальная документация php.net в оффлайн-режиме – только для практического задания;
2. Памятка по обозначениям блок-схем ГОСТ 19.701-90 и сложности стандартных операций, выданная преподавателем.

5.1.4 КРИТЕРИИ ОЦЕНКИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Таблица 3 – Критерии оценки

Оценка	Показатели оценки
Отлично (90–100 баллов)	Студент умеет увязывать теорию с практикой (решает задачи, формулирует выводы, поясняет полученные результаты), владеет понятийным аппаратом, полно и глубоко овладел материалом, обосновывает свои суждения и даёт правильные ответы на вопросы преподавателя.
Хорошо (75–89	Студент умеет увязывать теорию с практикой, владеет

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Оценка	Показатели оценки
баллов)	понятийным аппаратом, полно овладел материалом, но ответы содержат некоторые неточности и требуют уточнения и комментария со стороны преподавателя.
Удовлетворительно (60–74 балла)	Студент знает и понимает материал, но изложение неполное, непоследовательное, допускаются неточности в определении понятий, студент не может обосновать свои ответы на уточняющие вопросы.
Неудовлетворительно (менее 60 баллов)	Студент допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал, делает ошибки в ответах на уточняющие вопросы.

Критерии оценивания экзамена

Экзамен охватывает весь курс «Основы алгоритмизации и программирования» – разделы 1–6 (алгоритмизация и инструментальные средства, основы PHP, алгоритмы и структуры данных, ООП и качество кода, Laravel, применение ИИ-ассистентов). Банк состоит из 24 экзаменационных билетов; обучающийся вытягивает один билет случайным образом.

Формат билета. Вопрос 1 и вопрос 2 относятся к разным разделам курса; практическое задание – разбор конкретного фрагмента кода, блок-схемы или расчёта с 2–3 подвопросами.

Распределение баллов (максимум за билет – 100 баллов).

Часть	Баллы
Вопрос 1	35
Вопрос 2	35
Практическое задание	30
Итого	100

Теоретические вопросы (35 баллов) оцениваются по трём аспектам: полнота ответа (0–15), корректность формулировок и терминологии (0–15), связность и собственные примеры, в том числе фрагменты кода на PHP (0–5).

Практическое задание (30 баллов): правильный итоговый ответ на каждый подвопрос – 15 баллов суммарно; верное пошаговое обоснование (трассировка, оценка сложности, анализ кода) – 10 баллов; точное использование терминологии и синтаксиса PHP – 5 баллов.

Шкала перевода баллов в оценку.

Баллы	Оценка
90–100	«отлично»
75–89	«хорошо»
60–74	«удовлетворительно»
менее 60	«неудовлетворительно»

Дополнительные правила.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

– Практическая часть обязательна: при отсутствии верного решения практического задания оценка не может быть выше «удовлетворительно».
– Существенная фактическая ошибка (например, перепутаны LIFO и FIFO, сложность быстрой и пузырьковой сортировок, различие == и ===) снижает оценку за вопрос не менее чем на треть.
– Использование ИИ-ассистентов, мобильных устройств и посторонних материалов на экзамене является основанием для аннулирования результата и повторной сдачи в установленном порядке.
– Если ответ не соответствует вопросу билета, но по существу верен и относится к материалу курса, преподаватель вправе засчитать его на общих основаниях, зафиксировав это в ведомости.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

5.2 КОМПЛЕКТ «ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ»

5.2.1 Вопросы к экзамену

1. Классификация языков программирования. Компиляторы и интерпретаторы. Понятие системы программирования.
2. Жизненный цикл программы. Основные этапы решения задач на компьютере. Характеристики программного продукта.
3. Система контроля версий Git: назначение, три состояния файла, основные команды (init, add, commit, log, diff).
4. Ветвление в Git: ветка, указатель HEAD, слияние перемоткой и трёхстороннее слияние, конфликт слияния и его разрешение.
5. Платформа GitVerse: удалённый репозиторий, аутентификация по SSH и токенам, запрос на слияние и код-ревью. Что нельзя помещать в репозиторий.
6. Понятие алгоритма и алгоритмизации. Свойства алгоритма. Формальные и неформальные алгоритмы.
7. Способы записи алгоритмов. Графические обозначения блок-схем по ГОСТ 19.701-90 и правила построения схем.
8. Базовые алгоритмические конструкции. Структурное программирование и теорема Бёма – Якопини.
9. Нисходящее проектирование алгоритмов. Трассировка алгоритма, тестовые наборы, граничные значения, инвариант цикла.
10. Алгоритм Евклида: описание, блок-схема, реализация на PHP, число шагов.
11. Сложность алгоритмов: время и память, лучший, средний и худший случаи. Асимптотические обозначения O, Ω, Θ.
12. Классы сложности O(1), O(log n), O(n), O(n log n), $O(n^2)$, $O(2^n)$. Правила оценки сложности фрагмента кода с примерами на PHP.
13. Сложность рекурсивных алгоритмов. Экспериментальная оценка времени работы и памяти (hrtime, memory_get_peak_usage).
14. Структура программы на PHP. Переменные, константы, ввод и вывод данных.
15. Типы данных PHP. Приведение типов, режим strict_types. Различие операторов == и ===.
16. Операции и выражения PHP: приоритет, арифметические, логические и строковые операции, оператор <=>, оператор ??.
17. Условный оператор, оператор выбора switch и выражение match.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

18. Циклы while, do…while, for, foreach. Операторы break и continue. Вложенные циклы.
19. Массивы PHP: индексные, ассоциативные, многомерные. Основные функции. Множества и операции над ними. Запись как ассоциативный массив.
20. Строки и стандартные функции для работы со строками. Многобайтовые функции mb_*. Форматирование sprintf и number_format.
21. Регулярные выражения: назначение, функции preg_match, preg_match_all, preg_replace, метасимволы, кванторы, группы.
22. Файлы последовательного и прямого доступа. Форматы CSV и JSON. Безопасная работа с файлами.
23. Подпрограммы: определение и вызов функций, параметры, значения по умолчанию, типы, замыкания. Библиотеки подпрограмм.
24. Область видимости и время жизни переменных. Статические переменные. Передача параметров по значению и по ссылке. Управление памятью в PHP.
25. Рекурсия: базовый случай и рекурсивный шаг, стек вызовов. Сравнение рекурсии и итерации. Мемоизация.
26. Линейный и бинарный поиск: условия применения, алгоритмы, сложность.
27. Сортировки пузырьком, выбором и вставками: алгоритмы, сложность, устойчивость.
28. Сортировка слиянием и быстрая сортировка: идея, сложность, худший случай.
29. Встроенные функции сортировки PHP (sort, usort). Устойчивость сортировки. Выбор алгоритма сортировки под задачу.
30. Стек: операции, реализация на массиве и SplStack, применение (проверка скобок, обратная польская запись, стек вызовов).
31. Очередь, дек, приоритетная очередь. Классы SPL. Применение очередей в веб-приложениях.
32. Связные списки: односвязный и двусвязный. Операции и их сложность. Сравнение списка и массива.
33. Хеш-таблицы: хеш-функция, коллизии, средняя сложность операций. Ассоциативный массив PHP. Множества.
34. Двоичное дерево поиска: вставка, поиск, прямой, симметричный и обратный обходы.
35. Графы: матрица и список смежности. Обход графа в ширину и в глубину.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

36. Выбор структуры данных под задачу. Сводная таблица сложности операций для массива, списка, стека, очереди, хеш-таблицы и дерева.
37. Основные понятия ООП: класс, объект, свойства, методы, конструктор, модификаторы доступа. Инкапсуляция.
38. Наследование, переопределение методов, абстрактные классы, интерфейсы, полиморфизм. Особенности «перегрузки» методов в PHP.
39. Возможности PHP 8.x: конструкторное продвижение свойств, readonly, перечисления (enum), match, именованные аргументы.
40. Исключения: иерархия Throwable, Exception и Error, конструкция try/catch/finally, собственные исключения.
41. Отладка и модульное тестирование: Xdebug, Pest, граничные значения, эталонные результаты.
42. Модульное программирование: пространства имён, Composer, автозагрузка PSR-4, файлы composer.json и composer.lock.
43. Стандарты кодирования PSR-1 и PSR-12. Принципы чистого кода: DRY, KISS, YAGNI.
44. Принципы SOLID с примерами на PHP.
45. Внедрение зависимостей. Контейнер внедрения зависимостей. Библиотека и фреймворк.
46. Протокол HTTP, формы, сессии и cookie. Уязвимость XSS и экранирование вывода.
47. Архитектура Laravel: структура проекта, жизненный цикл запроса, сервис-провайдеры, контейнер.
48. Маршрутизация, контроллеры и посредники (middleware) в Laravel. Route Model Binding.
49. Шаблонизатор Blade: вывод данных, директивы, макеты, компоненты.
50. Доступ к базе данных из PHP: PDO и подготовленные выражения. SQL-инъекции и защита от них.
51. Eloquent ORM: модели, миграции, CRUD, отношения между моделями, проблема N+1 и жадная загрузка.
52. Валидация данных (Form Requests) и аутентификация (Breeze, Sanctum) в Laravel. Защита от CSRF.
53. Сервисный слой, DTO, API Resources и собственные исключения в Laravel.
54. Коллекции Laravel и их связь с типовыми алгоритмами обработки данных. Административная панель Orchid Platform.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

55. Российские ИИ-ассистенты (GigaChat, GigaCode, Алиса AI, YandexGPT): назначение и возможности при разработке на PHP.
56. Принципы постановки задач ИИ-ассистенту при генерации кода и тестов: контекст, ограничения, формат результата.
57. Риски применения ИИ-ассистентов: недостоверный и устаревший код, небезопасные решения, неэффективные алгоритмы. Порядок проверки результата.
58. Требования законодательства и этики при работе с ИИ: защита персональных данных (152-ФЗ), конфиденциальность кода, академическая добросовестность.

5.2.2 Типовые практические задания

1. Постройте блок-схему по ГОСТ 19.701-90 и составьте программу на PHP, которая вводит натуральное число n и выводит сумму его цифр; выполните трассировку для n = 4075.
2. Определите сложность следующих фрагментов в зависимости от n и обоснуйте ответ: цикл с постоянным шагом; два вложенных цикла по n; цикл, в котором счётчик удваивается; рекурсивный вызов с делением задачи пополам.
3. Измерьте функцией hrtime() время линейного и бинарного поиска в массиве из 10 000, 100 000 и 1 000 000 элементов; составьте таблицу и сопоставьте результат с теоретической оценкой.
4. Напишите на PHP функцию, принимающую массив товаров (название, цена, категория) и возвращающую суммарную стоимость по каждой категории; проверьте её на тестовом массиве.
5. Реализуйте на PHP сортировку вставками и сортировку слиянием; подсчитайте число сравнений на массивах [5, 4, 3, 2, 1] и [1, 2, 3, 4, 5] и объясните различие.
6. Реализуйте стек на массиве и с его помощью проверьте правильность расстановки скобок (), [], {} в строке; приведите три теста, включая граничные.
7. Реализуйте односвязный список (узел, добавление в начало и конец, удаление по значению, обход) и сравните трудоёмкость вставки в начало со вставкой в начало массива.
8. Подсчитайте частоту слов в тексте с помощью ассоциативного массива, выведите пять самых частых слов; укажите сложность алгоритма.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

9. Постройте двоичное дерево поиска из последовательности 40, 20, 60, 10, 30, 50, 70, выведите три вида обхода и найдите путь поиска ключа 50.
10. Спроектируйте иерархию классов «Платёжный метод» с интерфейсом и двумя реализациями, исключением для недостаточной суммы и модульными тестами Pest на нормальные и граничные случаи.
11. Создайте Composer-проект с автозагрузкой PSR-4, приведите код к стилю PSR-12 и зафиксируйте результат в репозитории GitVerse отдельными коммитами; оформите запрос на слияние.
12. В проекте Laravel опишите маршрут, контроллер, Form Request и модель Eloquent для сущности «Отзыв о товаре»; устраните проблему N+1 при выводе списка товаров с отзывами.
13. Сформулируйте запрос к ИИ-ассистенту (GigaChat или Алиса AI) на генерацию функции поиска дубликатов в массиве; проверьте результат тестами, оцените сложность, найдите не менее двух недостатков и исправьте их.

5.2.3 Задания для экзамена

Экзаменационный билет №1

Вопрос 1

Понятие алгоритма и алгоритмизации. Свойства алгоритма. Формальные и неформальные алгоритмы.

Вопрос 2

Связные списки: односвязный и двусвязный. Операции и их сложность. Сравнение списка и массива.

Практическое задание

Дан алгоритм, записанный на псевдокоде:

```
s ← 0; i ← 1
пока i ≤ 5:
    если i mod 2 = 1: s ← s + i
    i ← i + 1
вывод s
```

1. Постройте блок-схему алгоритма по ГОСТ 19.701-90.
2. Выполните трассировку: заполните таблицу значений i и s по шагам и укажите результат.
3. Запишите алгоритм на PHP и укажите, что он вычисляет.

Эталон (для преподавателя). Вычисляется сумма нечётных чисел от 1 до 5: после итераций s = 1, 1, 4, 4, 9; при i = 6 цикл завершается; результат 9. Цикл с предусловием (`while`), условие внутри цикла – неполное ветвление.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Экзаменационный билет №2

Вопрос 1

Алгоритм Евклида: описание, блок-схема, реализация на PHP, число шагов.

Вопрос 2

Область видимости и время жизни переменных. Статические переменные. Передача параметров по значению и по ссылке. Управление памятью в PHP.

Практическое задание

Дана функция:

```
function gcd(int $a, int $b): int
{
    while ($b !== 0) {
        [$a, $b] = [$b, $a % $b];
    }
    return $a;
}
echo gcd(84, 36);
```

1. Выполните трассировку для вызова `gcd(84, 36)` и укажите, что будет выведено и сколько раз выполнится тело цикла.
2. Назовите алгоритм и сформулируйте его свойство, гарантирующее завершение работы.
3. Оцените сложность алгоритма и укажите, как изменится результат при `gcd(36, 84)`.

Эталон (для преподавателя). Шаги: (84, 36) → (36, 12) → (12, 0); выведено 12; тело цикла выполняется 2 раза. Алгоритм Евклида; остаток строго убывает и неотрицателен, поэтому алгоритм конечен. Сложность O(log min(a, b)). Для `gcd(36, 84)` первый шаг меняет числа местами (36 % 84 = 36), результат тот же – 12, шагов на один больше (3).

Экзаменационный билет №3

Вопрос 1

Классы сложности O(1), O(log n), O(n), O(n log n), $O(n^2)$, $O(2^n)$. Правила оценки сложности фрагмента кода с примерами на PHP.

Вопрос 2

Наследование, переопределение методов, абстрактные классы, интерфейсы, полиморфизм. Особенности «перегрузки» методов в PHP.

Практическое задание

Определите асимптотическую сложность фрагментов (n – размер входных данных) и число выполнений тела внутреннего цикла при n = 16:

```
// А
for ($i = 0; $i < $n; $i++) { $s += $a[$i]; }


// Б
for ($i = 0; $i < $n; $i++) {
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
    for ($j = 0; $j < $i; $j++) { $c++; }
}

// B
for ($i = 1; $i < $n; $i *= 2) { $c++; }
```

1. Укажите класс сложности каждого фрагмента.
2. Вычислите число итераций при n = 16 для фрагментов Б и В.
3. Укажите, какой из фрагментов масштабируется хуже всего и почему.

Эталон (для преподавателя). А – O(n); Б – n(n − 1)/2 итераций, то есть O(n²); В – счётчик принимает значения 1, 2, 4, 8, поэтому O(log n). При n = 16: Б – 120 итераций, В – 4 итерации. Хуже всего растёт Б: при удвоении n число операций возрастает примерно вчетверо.

Экзаменационный билет №4

Вопрос 1

Линейный и бинарный поиск: условия применения, алгоритмы, сложность.

Вопрос 2

Платформа GitVerse: удалённый репозиторий, аутентификация по SSH и токенам, запрос на слияние и код-ревью. Что нельзя помещать в репозиторий.

Практическое задание

Дан упорядоченный массив `[3, 8, 15, 21, 27, 34, 40, 56, 63, 77]` (индексы 0–9) и функция бинарного поиска:

```
function binarySearch(array $a, int $x): int
{
    $lo = 0;
    $hi = count($a) - 1;
    while ($lo <= $hi) {
        $mid = intdiv($lo + $hi, 2);
        if ($a[$mid] === $x) {
            return $mid;
        }
        if ($a[$mid] < $x) {
            $lo = $mid + 1;
        } else {
            $hi = $mid - 1;
        }
    }
    return -1;
}
```

1. Выполните трассировку вызова `binarySearch($a, 56)` (значения lo, hi, mid на каждой итерации) и укажите результат.
2. Определите число итераций при поиске числа 10 и результат.
3. Оцените, сколько сравнений потребуется в худшем случае при n = 1000.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Эталон (для преподавателя). Поиск 56: итерация 1 – lo = 0, hi = 9, mid = 4 (27 < 56) → lo = 5; итерация 2 – mid = 7 (56), результат 7. Поиск 10: mid = 4 (27) → hi = 3; mid = 1 (8) → lo = 2; mid = 2 (15) → hi = 1; цикл завершён, результат −1, итераций 3. Для n = 1000 не более $\lfloor \log_2 1000 \rfloor + 1 = 10$ итераций.

Экзаменационный билет №5

Вопрос 1

Сортировки пузырьком, выбором и вставками: алгоритмы, сложность, устойчивость.

Вопрос 2

Архитектура Laravel: структура проекта, жизненный цикл запроса, сервис-провайдеры, контейнер.

Практическое задание

Массив `[5, 1, 4, 2, 8]` сортируется по возрастанию пузырьком:

```
$n = count($a);
for ($p = 0; $p < $n - 1; $p++) {
    for ($j = 0; $j < $n - 1 - $p; $j++) {
        if ($a[$j] > $a[$j + 1]) {
            [$a[$j], $a[$j + 1]] = [$a[$j + 1], $a[$j]];
        }
    }
}
```

1. Запишите состояние массива после первого и второго проходов внешнего цикла.
2. Подсчитайте общее число сравнений и обменов.
3. Укажите сложность алгоритма в худшем случае и предложите улучшение, ускоряющее его на почти упорядоченных данных.

Эталон (для преподавателя). После прохода 1: `[1, 4, 2, 5, 8]`; после прохода 2: `[1, 2, 4, 5, 8]`. Сравнений $n(n-1)/2 = 10$, обменов 4 (число инверсий: (5,1), (5,4), (5,2), (4,2)). Сложность $O(n^2)$. Улучшение: флаг «обменов не было» – досрочный выход, на упорядоченных данных $O(n)$.

Экзаменационный билет №6

Вопрос 1

Сортировка слиянием и быстрая сортировка: идея, сложность, худший случай.

Вопрос 2

Исключения: иерархия Throwable, Exception и Error, конструкция try/catch/finally, собственные исключения.

Практическое задание

Выполняется слияние двух упорядоченных массивов `[1, 4, 9]` и `[2, 3, 10]` на заключительном шаге сортировки слиянием:

```
function merge(array $l, array $r): array
{
    $out = [];
    $i = $j = 0;
    while ($i < count($l) && $j < count($r)) {
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
        $out[] = $l[$i] <= $r[$j] ? $l[$i++] : $r[$j++];
    }
    while ($i < count($l)) { $out[] = $l[$i++]; }
    while ($j < count($r)) { $out[] = $r[$j++]; }
    return $out;
}
```

1. Выполните слияние по шагам и запишите результат.
2. Подсчитайте число сравнений элементов.
3. Объясните, почему операция <= (а не <) обеспечивает устойчивость сортировки, и укажите сложность сортировки слиянием по времени и памяти.

Эталон (для преподавателя). Результат `[1, 2, 3, 4, 9, 10]`; сравнения: 1–2, 4–2, 4–3, 4–10, 9–10, то есть 5; затем остаток `[10]` добавляется без сравнений. При равных ключах из левого (исходно более раннего) массива берётся элемент первым. Время O(n log n), память O(n).

Экзаменационный билет №7

Вопрос 1

Стек: операции, реализация на массиве и SplStack, применение (проверка скобок, обратная польская запись, стек вызовов).

Вопрос 2

Регулярные выражения: назначение, функции preg_match, preg_match_all, preg_replace, метасимволы, кванторы, группы.

Практическое задание

Выражение в обратной польской записи вычисляется с помощью стека:

```
function rpn(string $expr): int
{
    $st = [];
    foreach (explode(' ', $expr) as $t) {
        if (is_numeric($t)) {
            $st[] = (int) $t;
            continue;
        }
        $b = array_pop($st);
        $a = array_pop($st);
        $st[] = match ($t) {
            '+' => $a + $b,
            '-' => $a - $b,
            '*' => $a * $b,
        };
    }
    return array_pop($st);
}
echo rpn('5 1 2 + 4 * + 3 -');
```

1. Выполните трассировку: запишите содержимое стека после каждого токена.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

2. Укажите результат и запишите исходное выражение в обычной (инфиксной) записи.
3. Укажите, какие ошибочные входные данные функция не обрабатывает, и как это исправить.

Эталон (для преподавателя). Стек: [5]; [5,1]; [5,1,2]; [5,3]; [5,3,4]; [5,12]; [17]; [17,3]; [14]. Результат 14; выражение 5 + (1 + 2) * 4 − 3. Не обрабатываются: недостаточное число операндов (`array_pop` вернёт null), неизвестный оператор (`match` выбросит `UnhandledMatchError`), деление на ноль при расширении набора операций; нужны проверки размера стека и собственное исключение.

Экзаменационный билет №8

Вопрос 1

Графы: матрица и список смежности. Обход графа в ширину и в глубину.

Вопрос 2

Принципы SOLID с примерами на PHP.

Практическое задание

Граф задан списками смежности: A: [B, C]; B: [D]; C: [D, E]; D: [F]; E: [F]; F: [].

1. Запишите порядок посещения вершин при обходе в ширину из вершины A (соседи просматриваются в указанном порядке).
2. Запишите порядок посещения при рекурсивном обходе в глубину из вершины A.
3. Назовите структуры данных, на которых строятся эти обходы, и сложность обходов при V вершинах и E рёбрах.

Эталон (для преподавателя). Обход в ширину: A, B, C, D, E, F (используется очередь). Обход в глубину: A, B, D, F, C, E (используется стек вызовов или явный стек). Сложность обоих – O(V + E) при представлении списками смежности.

Экзаменационный билет №9

Вопрос 1

Хеш-таблицы: хеш-функция, коллизии, средняя сложность операций. Ассоциативный массив PHP. Множества.

Вопрос 2

Доступ к базе данных из PHP: PDO и подготовленные выражения. SQL-инъекции и защита от них.

Практическое задание

Студент подсчитывает частоту слов, а ИИ-ассистент предложил фрагмент:

```
$words = ['php', 'git', 'php', 'laravel', 'git', 'php'];
$count = [];
foreach ($words as $w) {
    $count[$w]++;
}
arsort($count);
print_r($count);
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

1. Укажите, какое предупреждение выдаст PHP 8 и почему.
2. Исправьте фрагмент и запишите результат работы.
3. Оцените сложность алгоритма и объясните, какая структура данных лежит в основе ассоциативного массива.

Эталон (для преподавателя). При первом обращении `$count['php']` отсутствует: предупреждение «Undefined array key». Исправление: `$count[$w] = ($count[$w] ?? 0) + 1;`. Результат: php => 3, git => 2, laravel => 1. Сложность O(n) в среднем (операция с ключом O(1)); ассоциативный массив реализован как хеш-таблица; сортировка `arsort` добавляет O(k log k) для k различных слов.

Экзаменационный билет №10

Вопрос 1

Двоичное дерево поиска: вставка, поиск, прямой, симметричный и обратный обходы.

Вопрос 2

Нисходящее проектирование алгоритмов. Трассировка алгоритма, тестовые наборы, граничные значения, инвариант цикла.

Практическое задание

В пустое двоичное дерево поиска последовательно добавлены ключи 50, 30, 70, 20, 40, 60, 80.

1. Нарисуйте полученное дерево.
2. Запишите результаты прямого, симметричного и обратного обходов.
3. Укажите путь и число сравнений при поиске ключа 60, а также высоту дерева. Что произойдёт со сложностью поиска, если ключи добавлять в порядке возрастания?

Эталон (для преподавателя). Корень 50; слева 30 (20, 40), справа 70 (60, 80). Прямой обход: 50 30 20 40 70 60 80; симметричный: 20 30 40 50 60 70 80 (по возрастанию); обратный: 20 40 30 60 80 70 50. Поиск 60: 50 → 70 → 60, три сравнения; высота 3 уровня. При добавлении по возрастанию дерево вырождается в список, поиск становится O(n); решение – балансировка.

Экзаменационный билет №11

Вопрос 1

Рекурсия: базовый случай и рекурсивный шаг, стек вызовов. Сравнение рекурсии и итерации. Мемоизация.

Вопрос 2

Сложность рекурсивных алгоритмов. Экспериментальная оценка времени работы и памяти (hrtime, memory_get_peak_usage).

Практическое задание

Даны функции:

```
function fact(int $n): int
{
    return $n <= 1 ? 1 : $n * fact($n - 1);
}
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
function fib(int $n): int
{
    return $n <= 1 ? $n : fib($n - 1) + fib($n - 2);
}
```

1. Укажите результат `fact(5)` и максимальную глубину рекурсии.
2. Подсчитайте число вызовов функции при вычислении `fib(5)`.
3. Объясните, почему `fib` неэффективна, и предложите улучшение с оценкой новой сложности.

Эталон (для преподавателя). `fact(5)` = 120, глубина 5 вызовов. Число вызовов `fib(5)` равно 15 (для n = 0..5: 1, 1, 3, 5, 9, 15), `fib(5)` = 5. Подзадачи пересчитываются многократно, сложность экспоненциальная. Улучшение – мемоизация (массив результатов) или итеративный цикл: O(n) по времени, O(n) или O(1) по памяти.

Экзаменационный билет №12

Вопрос 1

Подпрограммы: определение и вызов функций, параметры, значения по умолчанию, типы, замыкания. Библиотеки подпрограмм.

Вопрос 2

Внедрение зависимостей. Контейнер внедрения зависимостей. Библиотека и фреймворк.

Практическое задание

Определите, что будет выведено, и объясните результат:

```
function add(array $a, int $x): void { $a[] = $x; }
function addRef(array &$a, int $x): void { $a[] = $x; }
function counter(): int { static $n = 0; return ++$n; }

$list = [1, 2];
add($list, 3);
addRef($list, 4);
echo count($list), PHP_EOL;
echo counter(), counter(), counter(), PHP_EOL;

$x = 10;
function show(): string { return isset($x) ? 'задана' : 'не задана'; }
echo show();
```

1. Запишите вывод программы.
2. Объясните различие передачи по значению и по ссылке.
3. Объясните поведение переменной `static` и области видимости `$x`.

Эталон (для преподавателя). Вывод: `3`, `123`, `не задана`. В `add` изменяется локальная копия массива, в `addRef` – исходный массив (получается [1, 2, 4]). Статическая

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

переменная сохраняет значение между вызовами. `$x` определена в глобальной области и внутри функции недоступна без `global` или передачи параметром.

Экзаменационный билет №13

Вопрос 1

Строки и стандартные функции для работы со строками. Многобайтовые функции mb_*. Форматирование sprintf и number_format.

Вопрос 2

Очередь, дек, приоритетная очередь. Классы SPL. Применение очередей в веб-приложениях.

Практическое задание

Дан массив товаров и два фрагмента кода:

```
$items = [
    ['name' => 'Мышь', 'price' => 1200],
    ['name' => 'Клавиатура', 'price' => 3500],
    ['name' => 'Коврик', 'price' => 400],
];
usort($items, fn($a, $b) => $a['price'] <=> $b['price']);

var_dump(preg_match('/^\+7\d{10}$/', '+79991234567'));
var_dump(preg_match('/^\+7\d{10}$/', '89991234567'));
```

1. Запишите порядок названий товаров после сортировки и объясните роль оператора <=>.
2. Запишите значения, возвращаемые `preg_match`, и объясните смысл регулярного выражения.
3. Напишите регулярное выражение для промокода формата «три заглавные латинские буквы, дефис, четыре цифры» и проверьте строки `ABC-1234` и `AB-12345`.

Эталон (для преподавателя). Порядок: Коврик, Мышь, Клавиатура (по возрастанию цены). Оператор возвращает −1, 0 или 1. `preg_match` вернёт 1 и 0: российский номер в формате +7 и десять цифр, якоря `^` и `$` требуют полного совпадения. Промокод: `/^[A-Z]{3}-\d{4}$/`; `ABC-1234` – 1, `AB-12345` – 0.

Экзаменационный билет №14

Вопрос 1

Наследование, переопределение методов, абстрактные классы, интерфейсы, полиморфизм. Особенности «перегрузки» методов в PHP.

Вопрос 2

Массивы PHP: индексные, ассоциативные, многомерные. Основные функции. Множества и операции над ними. Запись как ассоциативный массив.

Практическое задание

Дан фрагмент кода:

```
abstract class Shape
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
{
    abstract public function area(): float;

    public function describe(): string
    {
        return sprintf('%s: %.2f', static::class, $this->area());
    }
}

class Circle extends Shape
{
    public function __construct(private float $r) {}
    public function area(): float { return M_PI * $this->r ** 2; }
}

class Rect extends Shape
{
    public function __construct(private float $w, private float $h) {}
    public function area(): float { return $this->w * $this->h; }
}

foreach ([new Circle(2), new Rect(3, 4)] as $s) {
    echo $s->describe(), PHP_EOL;
}
```

1. Запишите вывод программы.
2. Укажите, в чём проявляются полиморфизм, наследование и инкапсуляция в этом коде.
3. Опишите, как добавить новую фигуру без изменения существующих классов, и какой принцип SOLID при этом соблюдается.

Эталон (для преподавателя). Вывод: `Circle: 12.57`, `Rect: 12.00`. Полиморфизм – вызов `area()` и `describe()` через тип `Shape`; наследование – общий метод `describe()` в базовом классе; инкапсуляция – закрытые свойства. Новая фигура – новый класс, наследующий `Shape` (принцип открытости/закрытости).

Экзаменационный билет №15

Вопрос 1

Исключения: иерархия Throwable, Exception и Error, конструкция try/catch/finally, собственные исключения.

Вопрос 2

Файлы последовательного и прямого доступа. Форматы CSV и JSON. Безопасная работа с файлами.

Практическое задание

Дан фрагмент кода:

```
class NegativeException extends InvalidArgumentException {}

function sqrtSafe(float $x): float
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
{
    if ($x < 0) {
        throw new NegativeException("negative: $x");
    }
    return sqrt($x);
}

function run(float $v): float
{
    try {
        echo 'A ';
        $r = sqrtSafe($v);
        echo 'B ';
        return $r;
    } catch (NegativeException $e) {
        echo 'C ';
        return -1.0;
    } finally {
        echo 'D ';
    }
}
echo run(9), PHP_EOL;
echo run(-4), PHP_EOL;
```

1. Запишите вывод программы.
2. Объясните порядок выполнения блоков `try`, `catch`, `finally`.
3. Объясните, почему исключение лучше возврата кода ошибки, и когда следует создавать собственный класс исключения.

Эталон (для преподавателя). Вывод: `A B D 3` и `A C D -1`. Блок `finally` выполняется всегда, в том числе после `return`; значение результата уже вычислено, поэтому `D` печатается раньше результата. Исключение нельзя «забыть» проверить, оно переносит контекст и разделяет нормальный и аварийный поток; собственный класс создают для ошибок предметной области, чтобы обрабатывать их отдельно.

Экзаменационный билет №16

Вопрос 1

Модульное программирование: пространства имён, Composer, автозагрузка PSR-4, файлы composer.json и composer.lock.

Вопрос 2

Ветвление в Git: ветка, указатель HEAD, слияние перемоткой и трёхстороннее слияние, конфликт слияния и его разрешение.

Практическое задание

В файле `composer.json` указано:

```
{
    "autoload": {
        "psr-4": { "App\\": "src/" }
    }
}
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

В проекте есть файл `src/Mailer.php` с классом `App\Service\Mailer` (пространство имён `namespace App\Service;`). При обращении `new \App\Service\Mailer()` возникает ошибка «Class not found».

1. Объясните причину ошибки с точки зрения стандарта PSR-4.
2. Укажите правильный путь файла и необходимые команды Composer.
3. Опишите преимущества автозагрузки по сравнению с ручными `require`.

Эталон (для преподавателя). По PSR-4 префикс `App\` отображается на каталог `src/`, остальная часть имени соответствует подкаталогам: класс `App\Service\Mailer` должен находиться в `src/Service/Mailer.php`. Файл нужно переместить и выполнить `composer dump-autoload`. Автозагрузка подключает файлы по требованию, исключает порядок подключения и дублирование `require`.

Экзаменационный билет №17

Вопрос 1

Принципы SOLID с примерами на PHP.

Вопрос 2

Связные списки: односвязный и двусвязный. Операции и их сложность. Сравнение списка и массива.

Практическое задание

Класс, предложенный ассистентом:

```
class OrderService
{
    public function checkout(array $items, string $email): float
    {
        $total = 0;
        foreach ($items as $i) { $total += $i['price'] * $i['qty']; }
        if ($total > 5000) { $total = $total * 0.9; }
        $db = new PDO('sqlite:shop.db');
        $db->exec("INSERT INTO orders(email,total)
VALUES('$email',$total)");
        mail($email, 'Заказ оформлен', "Сумма: $total");
        return $total;
    }
}
```

1. Найдите не менее четырёх нарушений принципов SOLID, чистого кода и безопасности.
2. Предложите структуру классов и интерфейсов после рефакторинга.
3. Покажите, как внедряются зависимости через конструктор.

Эталон (для преподавателя). Нарушения: несколько ответственностей в одном методе (подсчёт, скидка, запись в БД, почта) – SRP; жёсткие зависимости `new PDO` и `mail()` – DIP; «магические» числа 5000 и 0.9; SQL-инъекция из-за подстановки `$email`; смена правил скидок требует правки класса – OCP. Решение: интерфейсы

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

`DiscountStrategy`, `OrderRepository`, `Notifier`; класс `OrderService` получает их в конструкторе; запрос – с подготовленными выражениями; константы вынесены.

Экзаменационный билет №18

Вопрос 1

Система контроля версий Git: назначение, три состояния файла, основные команды (init, add, commit, log, diff).

Вопрос 2

Eloquent ORM: модели, миграции, CRUD, отношения между моделями, проблема N+1 и жадная загрузка.

Практическое задание

В репозитории две ветки изменили одну и ту же строку файла `README.md`. При слиянии ветки `feature` в `main` Git сообщил о конфликте, и в файле появились маркеры:

```
<<<<<<< HEAD
Версия PHP: 8.4
=======
Версия PHP: 8.5
>>>>>>> feature
```

1. Опишите причину конфликта и назначение маркеров.
2. Перечислите последовательность команд Git для разрешения конфликта и завершения слияния (итоговый текст: «Версия PHP: 8.5»).
3. Укажите, как оформить запрос на слияние в GitVerse и что должен проверить рецензент.

Эталон (для преподавателя). Обе ветки изменили одну строку, Git не может выбрать автоматически; блок между `<<<<<<<` и `=======` – версия текущей ветки, между `=======` и `>>>>>>>` – сливаемой. Команды: `git switch main`, `git merge feature`, правка файла и удаление маркеров, `git add README.md`, `git commit -m "merge: объединение веток"`. В запросе – заголовок, описание («что сделано», «как проверить»); рецензент проверяет отсутствие маркеров, корректность изменений, результат проверок CI.

Экзаменационный билет №19

Вопрос 1

Маршрутизация, контроллеры и посредники (middleware) в Laravel. Route Model Binding.

Вопрос 2

Встроенные функции сортировки PHP (sort, usort). Устойчивость сортировки. Выбор алгоритма сортировки под задачу.

Практическое задание

Контроллер Laravel и маршрут:

```
Route::get('/products', [ProductController::class, 'index']);
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
public function index()
{
    $products = Product::all();
    return view('products.index', ['products' => $products]);
}
```

В представлении для каждого товара выводится `{{ $product->category->name }}`, в таблице 50 товаров.

1. Определите число SQL-запросов к базе и назовите проблему.
2. Исправьте код и укажите новое число запросов.
3. Опишите, как с помощью Route Model Binding получить один товар по маршруту `/products/{product}`.

Эталон (для преподавателя). Проблема N+1: один запрос выбирает товары и по одному на категорию каждого товара – 1 + 50 = 51 запрос. Исправление: `Product::with('category')->get()` – 2 запроса. Route Model Binding: `Route::get('/products/{product}', [ProductController::class, 'show']);` и `public function show(Product $product)` – модель находится по идентификатору автоматически, при отсутствии возвращается 404.

Экзаменационный билет №20

Вопрос 1

Валидация данных (Form Requests) и аутентификация (Breeze, Sanctum) в Laravel. Защита от CSRF.

Вопрос 2

Базовые алгоритмические конструкции. Структурное программирование и теорема Бёма – Якопини.

Практическое задание

В Form Request заданы правила проверки:

```
public function rules(): array
{
    return [
        'name'    => 'required|min:3',
        'rating'  => 'required|integer|between:1,5',
        'comment' => 'nullable|string|max:20',
    ];
}
```

На вход пришли данные: `name = "Ив"`, `rating = "6"`, `comment = "ok"`.

1. Определите, какие поля не пройдут проверку и по каким правилам.
2. Опишите, что произойдёт в Laravel при ошибке валидации в веб-запросе и в API-запросе.
3. Объясните, почему проверка данных на стороне клиента не заменяет проверку на сервере, и как связаны валидация и безопасность.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Эталон (для преподавателя). Не пройдут `name` (длина 2 < 3, правило `min:3`) и `rating` (6 вне диапазона 1–5, правило `between`); `comment` проходит. В веб-запросе происходит перенаправление назад с ошибками и старыми значениями, в API – ответ 422 с JSON-описанием ошибок. Клиентскую проверку можно обойти; серверная валидация – обязательный барьер против некорректных и вредоносных данных.

Экзаменационный билет №21

Вопрос 1

Коллекции Laravel и их связь с типовыми алгоритмами обработки данных. Административная панель Orchid Platform.

Вопрос 2

Сложность алгоритмов: время и память, лучший, средний и худший случаи. Асимптотические обозначения O, Ω, Θ.

Практическое задание

Дана коллекция заказов:

```
$orders = collect([
    ['city' => 'Тагил', 'sum' => 100],
    ['city' => 'Салда', 'sum' => 50],
    ['city' => 'Тагил', 'sum' => 70],
]);
$report = $orders->groupBy('city')->map(fn($g) => $g->sum('sum'));
```

1. Запишите значение `$report`.
2. Реализуйте то же вычисление на чистом PHP одним проходом по массиву и оцените сложность.
3. Объясните, какой структурой данных удобно пользоваться при группировке и почему.

Эталон (для преподавателя). `$report` = [Тагил => 170, Салда => 50]. На чистом PHP: `foreach ($orders as $o) { $r[$o['city']] = ($r[$o['city']] ?? 0) + $o['sum']; }`, сложность O(n). Группировка опирается на хеш-таблицу (ассоциативный массив), доступ по ключу O(1) в среднем.

Экзаменационный билет №22

Вопрос 1

Принципы постановки задач ИИ-ассистенту при генерации кода и тестов: контекст, ограничения, формат результата.

Вопрос 2

Выбор структуры данных под задачу. Сводная таблица сложности операций для массива, списка, стека, очереди, хеш-таблицы и дерева.

Практическое задание

ИИ-ассистент по запросу «найти повторяющиеся элементы массива» выдал функцию:

```
function findDuplicates($arr)
{
    $result = [];
    for ($i = 0; $i < count($arr); $i++) {
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
        for ($j = $i + 1; $j < count($arr); $j++) {
            if ($arr[$i] == $arr[$j] && !in_array($arr[$i], $result)) {
                $result[] = $arr[$i];
            }
        }
    }
    return $result;
}
```

1. Найдите не менее трёх недостатков (эффективность, типизация, корректность сравнения).
2. Напишите улучшенную версию со сложностью O(n).
3. Опишите, как вы проверите исправленную функцию и что напишете в отчёте об использовании ИИ.

Эталон (для преподавателя). Недостатки: вложенные циклы и `in_array` дают $O(n^2)$ и хуже; `count()` вызывается в условии на каждой итерации; нестрогое `==` считает равными, например, `'1e1'` и `'10'`; нет типов параметров и результата. Улучшение: `$seen = []; $dups = []; foreach ($items as $it) { if (isset($seen[$it])) { $dups[$it] = true; } else { $seen[$it] = true; } } return array_keys($dups);`. Проверка: тесты (пустой массив, без дубликатов, несколько дубликатов, нестрогие совпадения), сравнение результата; в отчёте – запрос, найденные недостатки, итоговый код, факт использования ИИ.

Экзаменационный билет №23

Вопрос 1

Требования законодательства и этики при работе с ИИ: защита персональных данных (152-ФЗ), конфиденциальность кода, академическая добросовестность.

Вопрос 2

Протокол HTTP, формы, сессии и cookie. Уязвимость XSS и экранирование вывода.

Практическое задание

Ассистент предложил код получения пользователя:

```
$email = $_GET['email'];
$pdo = new PDO('sqlite:app.db');
$rows = $pdo->query("SELECT * FROM users WHERE email = '$email'")-
>fetchAll();
```

Студент собирается вставить в запрос к ассистенту реальный пароль администратора БД «для точности».

1. Укажите уязвимость и пример вредоносного значения параметра `email`.
2. Перепишите код безопасно.
3. Объясните, почему нельзя передавать ассистенту пароли и персональные данные, и как поступить вместо этого.

Эталон (для преподавателя). SQL-инъекция: значение `' OR '1'='1` превращает условие в истинное для всех строк. Безопасный вариант: `$st = $pdo->prepare('SELECT * FROM users WHERE email = :email'); $st-`

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

`>execute(['email' => $email]); $rows = $st->fetchAll();`. Пароли, ключи и персональные данные нельзя передавать внешним сервисам (152-ФЗ, риск утечки); используют обезличенные примеры и переменные окружения (`.env`, не хранится в репозитории).

Экзаменационный билет №24

Вопрос 1

Российские ИИ-ассистенты (GigaChat, GigaCode, Алиса AI, YandexGPT): назначение и возможности при разработке на PHP.

Вопрос 2

Отладка и модульное тестирование: Xdebug, Pest, граничные значения, эталонные результаты.

Практическое задание

Ассистент сгенерировал функцию и единственный тест:

```
function maxOf(array $a): int
{
    if ($a === []) {
        throw new InvalidArgumentException('Пустой массив');
    }
    return max($a);
}


test('maxOf возвращает максимум', function () {
    expect(maxOf([1, 5, 3]))->toBe(5);
});
```

1. Назовите не менее четырёх недостающих тестовых случаев.
2. Напишите дополнительные тесты Pest для двух из них.
3. Объясните, почему прохождение единственного теста не доказывает корректность функции.

Эталон (для преподавателя). Случаи: пустой массив (ожидается исключение), один элемент, все элементы равны, отрицательные числа, максимум в начале или конце, повторяющийся максимум. Примеры: `expect(fn() => maxOf([]))->toThrow(InvalidArgumentException::class);` и `expect(maxOf([-3, -7]))->toBe(-3);`. Один тест проверяет лишь один путь выполнения; ошибки на границах и в исключительных ситуациях остаются незамеченными.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

5.3 КОМПЛЕКТ «ТЕКУЩИЙ КОНТРОЛЬ»

5.3.1 Задания для срезов знаний

1. Назовите свойства алгоритма. Какое из них нарушает последовательность «шаг 1: прибавить 1 к x; шаг 2: повторять шаг 1»? Обоснуйте.
2. Постройте таблицу трассировки алгоритма: `s = 0; для i от 2 до 10 шаг 2: s = s + i * i` и определите итоговое значение s (ответ: 220).
3. Определите класс сложности фрагмента: внешний цикл по i от 1 до n, внутренний – по j от 1 до i. Сколько раз выполнится тело цикла при n = 6?
4. Сколько сравнений в худшем случае потребуется бинарному поиску в упорядоченном массиве из 1000 элементов? Чем линейный поиск лучше бинарного?
5. Объясните различие операторов `==` и `===` в PHP. Каким будет результат выражений `"abc" == 0` и `"1" === 1` в PHP 8?
6. В чём различие передачи аргумента по значению и по ссылке? Приведите пример функции, которой нужна ссылка, и пример, где ссылка опасна.
7. Запишите порядок извлечения элементов из стека и очереди после добавления 1, 2, 3. Какие операции этих структур выполняются за O(1)?
8. Что такое коллизия в хеш-таблице? Как ассоциативный массив PHP позволяет найти дубликаты в массиве за O(n)?
9. Чем интерфейс отличается от абстрактного класса? Какое принцип SOLID нарушает класс, одновременно читающий файл, проверяющий данные и формирующий отчёт?
10. Назовите три риска применения ИИ-ассистента при написании кода PHP и по одному способу контроля каждого риска.

5.3.2 Тестовые задания

Текущий контроль знаний по темам программирования проводится в системе дистанционного обучения Moodle. Для каждой лекции курса создан банк тестовых вопросов (типы: один правильный ответ, множественный выбор, верно/неверно, краткий ответ); на занятии каждому обучающемуся выдаётся тест из случайно выбранных вопросов банка. Попыток – 2, время – 15 минут, оценка – по доле правильных ответов (90–100 % – «5», 75–89 % – «4», 60–74 % – «3»).

Банк вопросов	Число вопросов
0.1. Введение в Git	6
0.2. Ветвление в Git	6
0.3. GitVerse — знакомство и удалённые репозитории	6
0.4. Pull Request и код-ревью	6
0.5. Автоматизация проверки — GitVerse Actions	6
1.1. Введение в PHP	6

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Банк вопросов	Число вопросов
1.2. Основы синтаксиса	6
1.3. Условные операторы и циклы	8
1.4. Функции	6
1.5. Формы и обработка данных	8
1.6. Сессии и куки	6
1.7. ООП в PHP	8
1.8. Работа с базами данных	8
1.9. Обработка исключений	6
1.10. Регулярные выражения	6
1.11. Работа с внешними API	6
2.1. Composer — управление зависимостями	8
2.2. Пространства имён и автозагрузка PSR-4	6
2.3. Стандарты PSR	8
2.4. Чистый код	6
2.5. SOLID	8
2.6. Фреймворки — зачем они нужны	6
2.7. Dependency Injection	8
3.1. Введение в Laravel	8
3.2. Структура проекта Laravel	6
3.3. Маршрутизация	8
3.4. Контроллеры	6
3.5. Blade — директивы и вывод данных	6
3.6. Blade — компоненты и макеты	6
3.7. Eloquent ORM — модели и CRUD	8
3.8. Eloquent ORM — отношения между моделями	8
3.9. Form Requests — валидация данных	6
3.10. Аутентификация — сессии и Laravel Breeze	6
3.11. Аутентификация API — Laravel Sanctum	8
3.12. Middleware — посредники запроса	6
4.1. Современный PHP 8.x в Laravel	8
4.2. Сервис-слой	6
4.3. DTO — зачем он нужен	6
4.4. DTO на практике	8
4.5. API Resources	6
4.6. Кастомные исключения и обработка ошибок	8
4.7. Коллекции	6
4.8. Orchid Platform — введение и первый Screen	6
4.9. Orchid Platform — Table, Layout, Form	8
4.10. Orchid Platform — навигация и меню	6

Всего в банках Moodle – 304 вопросов.

Тестовые задания по алгоритмизации, структурам данных, PHP и применению ИИ-ассистентов

Банки Moodle не содержат заданий по алгоритмизации и безопасному применению ИИ, поэтому ниже приведён дополнительный банк из 34 заданий с ключами (тесты 1–

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

15 – алгоритмы и структуры данных, 16–25, 31–34 – PHP, Git, Laravel и качество кода, 26–30 – ИИ-ассистенты).

1. Какое свойство алгоритма означает, что процесс решения разбивается на отдельные элементарные шаги?
   а) результативность
   б) дискретность
   в) массовость
   г) детерминированность
   Правильный ответ: б

2. Какой фигурой в блок-схеме по ГОСТ 19.701-90 обозначается проверка условия с ветвлением?
   а) прямоугольник
   б) параллелограмм
   в) ромб
   г) овал с надписью «Начало»
   Правильный ответ: в

3. Чему равно значение s после выполнения алгоритма: s = 0; для i от 1 до 10 с шагом 3: s = s + i?
   а) 18
   б) 22
   в) 25
   г) 55
   Правильный ответ: б

4. Какова сложность фрагмента: два вложенных цикла по n, в теле – операция присваивания?
   а) O(n)
   б) O(n log n)
   в) $O(n^2)$
   г) $O(2^n)$
   Правильный ответ: в

5. Какова сложность цикла `for ($i = 1; $i < $n; $i *= 2)`?
   а) O(log n)
   б) O(n)
   в) O(1)
   г) $O(n^2)$
   Правильный ответ: а

6. Сколько сравнений в худшем случае требуется бинарному поиску в упорядоченном массиве из 1 000 000 элементов?
   а) около 1000
   б) не более 20
   в) не более 100

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

г) 1 000 000

Правильный ответ: б

7. Какое условие необходимо для применения бинарного поиска?

а) элементы целые числа

б) массив содержит не более 100 элементов

в) массив упорядочен

г) в массиве нет повторяющихся значений

Правильный ответ: в

8. Какой порядок извлечения элементов реализует стек?

а) FIFO – первым вошёл, первым вышел

б) LIFO – последним вошёл, первым вышел

в) по возрастанию ключа

г) случайный

Правильный ответ: б

9. Какая структура данных используется при обходе графа в ширину?

а) стек

б) очередь

в) дерево поиска

г) хеш-таблица

Правильный ответ: б

10. Какова средняя сложность поиска элемента по ключу в хеш-таблице (ассоциативном массиве PHP)?

а) O(n)

б) O(log n)

в) O(1)

г) $O(n^2)$

Правильный ответ: в

11. В каком порядке выдаёт ключи симметричный (in-order) обход двоичного дерева поиска?

а) по убыванию

б) по возрастанию

в) в порядке вставки

г) по уровням

Правильный ответ: б

12. Какова сложность вставки элемента в начало односвязного списка?

а) O(1)

б) O(log n)

в) O(n)

г) $O(n^2)$

Правильный ответ: а

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

13. В каком случае быстрая сортировка с выбором первого элемента в качестве опорного работает за $O(n^2)$?
    а) массив состоит из случайных чисел
    б) массив уже упорядочен
    в) в массиве менее 10 элементов
    г) массив состоит из строк
    Правильный ответ: б
14. Какая из сортировок является устойчивой?
    а) сортировка выбором (классическая реализация)
    б) быстрая сортировка (классическая реализация)
    в) сортировка слиянием
    г) ни одна из перечисленных
    Правильный ответ: в
15. Для чего обязателен базовый случай рекурсивной функции?
    а) для ускорения работы
    б) для завершения рекурсии
    в) для передачи параметров по ссылке
    г) для проверки типов
    Правильный ответ: б
16. Каков результат выражения `0 == "a"` в PHP 8?
    а) true
    б) false
    в) ошибка выполнения
    г) null
    Правильный ответ: б
17. Каков результат выражения `"10" + 5` в PHP при отсутствии `declare(strict_types=1)`?
    а) строка "105"
    б) ошибка типов
    в) целое число 15
    г) строка "15"
    Правильный ответ: в
18. Что выведет `echo intdiv(17, 5) . " " . 17 % 5;`?
    а) 3 2
    б) 3.4 2
    в) 2 3
    г) 3 3
    Правильный ответ: а
19. Как ведёт себя переменная, объявленная в функции с ключевым словом `static`?
    а) становится глобальной

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

б) сохраняет значение между вызовами функции
в) доступна только для чтения
г) удаляется после первого вызова
Правильный ответ: б

20. Что обеспечивает стандарт автозагрузки PSR-4?
а) форматирование кода
б) соответствие пространства имён и пути к файлу класса
в) шифрование паролей
г) миграции базы данных
Правильный ответ: б

21. Какой стандарт описывает стиль оформления кода PHP?
а) PSR-4
б) PSR-7
в) PSR-12
г) PSR-15
Правильный ответ: в

22. Что утверждает принцип единственной ответственности (S в SOLID)?
а) класс должен иметь одну причину для изменения
б) класс может иметь только один метод
в) класс не может наследоваться
г) в программе допускается один класс на файл
Правильный ответ: а

23. Как устранить проблему N+1 при выводе списка товаров с отзывами в Eloquent?
а) использовать `Product::with('reviews')->get()`
б) вызывать `count()` в цикле
в) заменить `get()` на `first()`
г) отключить миграции
Правильный ответ: а

24. Что лучше всего защищает от SQL-инъекций?
а) удаление кавычек из строки
б) подготовленные выражения с параметрами
в) использование `addslashes()`
г) скрытие сообщений об ошибках
Правильный ответ: б

25. Чем опасна директива Blade `{!! $comment !!}` для данных, введённых пользователем?
а) не выводит данные
б) выводит данные без экранирования, возможна XSS-уязвимость
в) замедляет базу данных

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

г) удаляет HTML-теги из данных

Правильный ответ: б

26. Какие сведения нельзя передавать в запросе к ИИ-ассистенту?

а) общие вопросы о синтаксисе PHP

б) пароли, ключи доступа, персональные данные и конфиденциальный код

в) тексты ошибок без реквизитов доступа

г) публичные фрагменты документации

Правильный ответ: б

27. Что необходимо сделать с кодом, полученным от ИИ-ассистента, перед включением в проект?

а) сразу зафиксировать в основной ветке

б) прочитать, проверить тестами и сверить с официальной документацией

в) отправить другому ИИ без чтения

г) удалить комментарии

Правильный ответ: б

28. Что такое «галлюцинация» языковой модели при генерации кода?

а) сбой компилятора

б) правдоподобный, но недостоверный ответ (несуществующая функция, неверный алгоритм)

в) ошибка соединения с сетью

г) замедление работы IDE

Правильный ответ: б

29. Какой ИИ-ассистент для разработки интегрирован с платформой GitVerse?

а) GigaCode

б) YandexGPT Maps

в) GigaMusic

г) Алиса Станция

Правильный ответ: а

30. Верно ли, что на контрольных работах и экзамене по дисциплине допускается использование ИИ-ассистентов?

а) да, если указать это в ответе

б) нет, использование не допускается

в) да, только для проверки синтаксиса

г) да, по желанию обучающегося

Правильный ответ: б

31. Какая команда создаёт новую ветку `feature/cart` и переключается на неё?

а) git branch feature/cart

б) git switch -c feature/cart

в) git merge feature/cart

г) git commit -b feature/cart

Правильный ответ: б

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

32. Что делает директива Blade `@csrf` внутри формы?
    а) подключает стили
    б) добавляет скрытое поле с токеном защиты от подделки межсайтовых запросов
    в) проверяет синтаксис формы
    г) включает отладку
    Правильный ответ: б
33. Какой модификатор доступа делает свойство класса недоступным извне класса и его наследников?
    а) public
    б) protected
    в) private
    г) static
    Правильный ответ: в
34. Что даёт объявление свойства `public readonly string $title` в конструкторе класса PHP 8?
    а) свойство нельзя изменить после инициализации
    б) свойство удаляется после вызова конструктора
    в) свойство становится статическим
    г) свойство хранится только в базе данных
    Правильный ответ: а

5.3.3 Задания для проверочных, самостоятельных и контрольных работ

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Контрольная работа №1. Алгоритмы и сложность

Охватывает: темы 1.3, 1.4, 2.3, 3.1; лабораторные работы №4–8, 15, 16.

Формат: письменная работа с использованием PHP CLI, ИИ-ассистенты не используются.

Длительность: 90 минут.

Задание (вариант 1; в других вариантах изменяются числовые данные и фрагменты)

Часть 1. Алгоритмы (12 баллов)

1. Постройте блок-схему по ГОСТ 19.701-90 и запишите на PHP функцию `digitSum(int $n): int`, возвращающую сумму цифр числа. Выполните трассировку для n = 4075. (4 балла)
2. Алгоритмом Евклида найдите НОД(84, 36). Запишите последовательность остатков и число шагов. (3 балла)
3. Дан массив `[3, 8, 12, 17, 23, 31, 40, 52]`. Выполните бинарный поиск ключей 31 и 10, указав индексы середины и число сравнений на каждом шаге. (5 баллов)

Часть 2. Сложность (6 баллов)

4. Определите асимптотическую сложность фрагментов и обоснуйте ответ (по 1 баллу):
   а) два вложенных цикла `for` по n с постоянным телом;
   б) цикл `for ($i = 1; $i < $n; $i *= 2);`
   в) внешний цикл по i от 1 до n, внутренний – по j от 1 до i;
   г) рекурсивная функция, вызывающая себя один раз с аргументом n − 1 и выполняющая O(1) операций;
   д) рекурсивная функция, вызывающая себя дважды с n/2 и выполняющая O(n) операций вне вызовов;
   е) линейный поиск в массиве из n элементов в худшем случае.

Часть 3. Теория (4 балла, по 1 баллу)

5. Назовите и поясните свойства алгоритма.
6. Чем различаются сложность в худшем, среднем и лучшем случаях? Приведите пример для линейного поиска.
7. Почему бинарный поиск неприменим к неупорядоченному массиву?
8. Что такое инвариант цикла и для чего он используется при доказательстве корректности?

Эталон ответов (для преподавателя)

1. Блок-схема: «Начало», «Ввод n», «n = |n|; s = 0», ромб «n > 0?», при «да» – блоки «s = s + n mod 10», «n = ⌊n / 10⌋» с возвратом к ромбу, при «нет» – «Вывод s», «Конец». Код:

```
function digitSum(int $n): int
{
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
    $n = abs($n);
    $s = 0;
    while ($n > 0) {
        $s += $n % 10;
        $n = intdiv($n, 10);
    }
    return $s;
}
```

Трассировка для n = 4075: (n, s) = (4075, 0) → (407, 5) → (40, 12) → (4, 12) → (0, 16). Результат: 16.

2. 84 mod 36 = 12; 36 mod 12 = 0. НОД = 12, шагов 2.
3. Ключ 31: lo = 0, hi = 7, mid = 3 (17 < 31) → lo = 4; mid = 5 (31) – найден, индекс 5, 2 сравнения. Ключ 10: mid = 3 (17 > 10) → hi = 2; mid = 1 (8 < 10) → lo = 2; mid = 2 (12 > 10) → hi = 1; lo > hi – не найден, 3 сравнения.
4. а) $O(n^2)$; б) $O(\log n)$; в) $n(n + 1)/2$ операций – $O(n^2)$; г) $T(n) = T(n - 1) + O(1)$ – $O(n)$; д) $T(n) = 2T(n/2) + O(n)$ – $O(n \log n)$; е) $O(n)$.
5. Дискретность, детерминированность, понятность, конечность (результативность), массовость, корректность.
6. Лучший случай линейного поиска – ключ в первой позиции (1 сравнение), худший – ключ в конце или отсутствует (n сравнений), средний – около n/2.
7. Решение «искомое слева или справа от середины» верно только если элементы упорядочены.
8. Инвариант – условие, истинное перед каждой итерацией и после неё; из инварианта и условия выхода выводится правильность результата.

Критерии оценки

Максимум – 22 баллов (часть 1 – 12, часть 2 – 6, часть 3 – 4).

Баллы	Оценка
20–22	«5»
16–19	«4»
12–15	«3»
менее 12	«2»

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Контрольная работа №2. Структуры данных и сортировки

Охватывает: темы 3.1–3.3; лабораторные работы №16–22.

Формат: письменная работа с использованием PHP CLI, ИИ-ассистенты не используются.

Длительность: 90 минут.

Задание (вариант 1)

Часть 1. Сортировки и поиск (6 баллов)

1. Выполните сортировку пузырьком массива `[5, 2, 4, 1, 3]` (без досрочного выхода): запишите состояние массива после каждого прохода, число сравнений и число обменов. (4 балла)
2. Какая из сортировок – пузырьком, выбором, вставками, слиянием – устойчива и почему? Какова худшая сложность быстрой сортировки? (2 балла)

Часть 2. Линейные структуры (8 баллов)

3. Напишите функцию `rpn(string $expr): int|float`, вычисляющую выражение в обратной польской записи с помощью стека (`SplStack`) и операций `+ - * /`. Вычислите `2 3 + 4 *` и `5 1 2 + 4 * + 3 -`. (5 баллов)
4. Определите, правильно ли расставлены скобки: `{[()]}`, `([)]`, `(([`. Опишите алгоритм проверки через стек. (3 балла)

Часть 3. Нелинейные структуры (8 баллов)

5. Постройте двоичное дерево поиска из последовательности 50, 30, 70, 20, 40, 60, 80, 35. Запишите прямой, симметричный и обратный обходы; путь поиска ключа 35. (4 балла)
6. Дан граф: A–B, A–C, B–D, C–D, D–E (соседи перебираются в алфавитном порядке). Запишите порядок обхода из вершины A в ширину и в глубину. (3 балла)
7. Опишите алгоритм подсчёта частот слов в тексте с помощью ассоциативного массива и укажите его сложность. (1 балл)

Эталон ответов (для преподавателя)

1. Проходы: [2, 4, 1, 3, 5] (4 сравнения, 4 обмена); [2, 1, 3, 4, 5] (3 сравнения, 2 обмена); [1, 2, 3, 4, 5] (2 сравнения, 1 обмен); [1, 2, 3, 4, 5] (1 сравнение, 0 обменов). Всего 10 сравнений и 7 обменов (7 – число инверсий массива).
2. Устойчивы сортировки пузырьком, вставками и слиянием (равные элементы не меняются местами); сортировка выбором в классической реализации неустойчива. Худшая сложность быстрой сортировки – $O(n^2)$ (например, упорядоченный массив при выборе первого элемента опорным).
3. Эталонная реализация:

```
function rpn(string $expr): int|float
{
    $stack = new SplStack();
    foreach (explode(' ', $expr) as $token) {
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
        if (is_numeric($token)) {
            $stack->push($token + 0);
            continue;
        }
        $b = $stack->pop();
        $a = $stack->pop();
        $stack->push(match ($token) {
            '+' => $a + $b,
            '-' => $a - $b,
            '*' => $a * $b,
            '/' => $a / $b,
        });
    }
    return $stack->pop();
}
```

Результаты: 20 и 14.

4. `{[()]}` – верно; `([)]` – неверно; `(()` – неверно. Открывающую скобку помещают в стек; при закрывающей извлекают верхний элемент и сравнивают с парной; в конце стек должен быть пуст.
5. Прямой: 50, 30, 20, 40, 35, 70, 60, 80; симметричный: 20, 30, 35, 40, 50, 60, 70, 80; обратный: 20, 35, 40, 30, 60, 80, 70, 50; путь к ключу 35: 50 → 30 → 40 → 35.
6. В ширину: A, B, C, D, E; в глубину: A, B, D, C, E.
7. Разбить текст на слова (`preg_split('/\W+/u', mb_strtolower($text), -1, PREG_SPLIT_NO_EMPTY)`), для каждого слова увеличить счётчик `$freq[$w] = ($freq[$w] ?? 0) + 1`; при необходимости упорядочить результат `arsort()`. Сложность подсчёта O(n) в среднем, с сортировкой – O(n + k log k), где n – число слов, k – число различных слов.

Критерии оценки

Максимум – 22 балла; шкала: 20–22 – «5», 16–19 – «4», 12–15 – «3», менее 12 – «2».

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Контрольная работа №3. PHP: валидация, исключения и ООП

Охватывает: лекции 1-11 раздела «1. PHP» и лабораторные ЛР 1.1-1.10

Формат: автономная практическая задача (не связана с проектом `online-shop`) + теория

Длительность: 90 минут

Часть 1. Практическое задание

Реализовать на чистом PHP (без Composer, без фреймворков) консольный или веб-скрипт **«Валидатор анкеты участника»**, объединяющий темы всего раздела. Требования:

1. Скрипт принимает данные анкеты (имя, email, возраст, телефон, промокод-приглашение) через форму (`$_POST`) или через жёстко заданный тестовый массив, если преподаватель выдаёт консольный вариант задания.
2. Валидация: - имя — не пустое, минимум 2 символа; - email — через `filter_var(..., FILTER_VALIDATE_EMAIL)`; - возраст — целое число от 14 до 120; - телефон — через регулярное выражение в формате `+7XXXXXXXXXX`; - промокод-приглашение (необязательное поле) — если указан, должен соответствовать формату `XXX-0000` (3 заглавные буквы, дефис, 4 цифры).
3. При ошибке валидации хотя бы одного поля — выбрасывается собственное исключение `ValidationException`, несущее имя поля и сообщение об ошибке; все ошибки собираются и выводятся студенту одним списком (не только первая найденная).
4. При успешной валидации — данные сохраняются в объект `Applicant` (класс с приватными свойствами, конструкторным продвижением, геттерами) и добавляются в статичный список через метод `ApplicantRegistry::add(Applicant $applicant): void`.
5. Реализовать метод `ApplicantRegistry::countByAgeGroup(): array`, возвращающий массив вида `['14-17' => 2, '18-25' => 5, ...]`, посчитанный через `foreach`/`switch` без использования функций массивов раздела 4 (реализация «руками», закрепление циклов).
6. Весь пользовательский вывод должен проходить через `htmlspecialchars`, если задание выполняется в виде веб-формы.

Критерии оценки практической части

Критерий	Баллы
Корректная валидация всех 5 полей, включая regex-поля	4
Собственное исключение `ValidationException`, собирающее все ошибки	3
Класс `Applicant` с типизированными свойствами и конструкторным продвижением	2
Метод `countByAgeGroup()` реализован без функций массивов (`array_filter`/`array_map`)	2
Защита вывода через `htmlspecialchars`, отсутствие PHP-предупреждений	1

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Автоматическая проверка практической части (GitVerse Smart Class)

Эта контрольная выполняется в отдельном, самостоятельном репозитории (по аналогии с `control-work-0` из раздела 0) — предполагается структура `src/Applicant.php`, `src/ApplicantRegistry.php`, `src/ValidationException.php` и точка входа в корне репозитория; при иной раскладке файлов поправьте пути в скрипте под свою структуру. Проверка сознательно не завязывается на конкретный порядок параметров конструктора `Applicant` (он не зафиксирован в задании дословно) — только на их количество и приватность свойств.

Добавьте `tests/smartclass/kr-1.php`:

```
<?php
// tests/smartclass/kr-1.php — вспомогательный скрипт автопроверки КР 1

require __DIR__ . '/../../src/ValidationException.php';
require __DIR__ . '/../../src/Applicant.php';
require __DIR__ . '/../../src/ApplicantRegistry.php';

if (!is_subclass_of('ValidationException', Exception::class)) {
    fwrite(STDERR, "ValidationException должен наследоваться от
Exception\n");
    exit(1);
}

$applicantReflection = new ReflectionClass('Applicant');
$constructor = $applicantReflection->getConstructor();

if ($constructor === null || count($constructor->getParameters()) < 4) {
    fwrite(STDERR, "Конструктор Applicant должен принимать минимум 4
параметра (имя, email, возраст, телефон)\n");
    exit(1);
}

foreach ($applicantReflection->getProperties(ReflectionProperty::IS_PUBLIC)
as $property) {
    fwrite(STDERR, "Свойство Applicant::\${$property->getName()} должно
быть private\n");
    exit(1);
}

if (!method_exists('ApplicantRegistry', 'add') ||
!method_exists('ApplicantRegistry', 'countByAgeGroup')) {
    fwrite(STDERR, "ApplicantRegistry должен содержать методы add() и
countByAgeGroup()\n");
    exit(1);
}

if (!(new ReflectionMethod('ApplicantRegistry', 'countByAgeGroup'))-
>isStatic()) {
    fwrite(STDERR, "countByAgeGroup() должен быть статическим методом\n");
    exit(1);
}
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
echo "OK\n";
```

И job в `.gitverse/workflows/classroom.yml` этого репозитория:

```
  kr-1:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: shivammathur/setup-php@v2
        with:
          php-version: '8.5'
      - name: Автопроверка (SmartClass)
        run: |
          source .gitverse/scripts/grade.sh
          check_exit   "Классы — валидный синтаксис" \
            'php -l src/Applicant.php && php -l src/ApplicantRegistry.php
&& php -l src/ValidationException.php' 2
          check_exit   "Структура классов соответствует заданию
(reflection)" 'php tests/smartclass/kr-1.php' 6
          check_output "Телефон валидируется через regex с +7"    'grep -rq
"preg_match" src/*.php *.php 2>/dev/null && grep -rq "+7" src/*.php *.php
2>/dev/null && echo yes || echo no' exact "yes" 2
          check_output "Промокод: паттерн 3 буквы-дефис-4 цифры"  'grep -
rqF "[A-Z]{3}-\d{4}" src/*.php *.php 2>/dev/null && echo yes || echo no'
exact "yes" 2
          check_output "Вывод защищён через htmlspecialchars"     'grep -rq
"htmlspecialchars" src/*.php *.php 2>/dev/null && echo yes || echo no'
exact "yes" 1
          report
```

Не проверяется автоматически: собирает ли `ValidationException` (или обвязка вокруг него) действительно **все** ошибки валидации разом, а не только первую найденную — это вопрос поведения при нескольких одновременно неверных полях, для надёжной автоматической проверки потребовался бы фиксированный формат тестовых данных, которого задание намеренно не диктует; оценивается вручную по коду и демонстрации.

Часть 2. Теоретические вопросы

1. Почему подготовленные выражения (`:placeholder`) защищают от SQL-инъекций лучше, чем ручное экранирование строк через `addslashes`?
2. Объясните разницу между `Exception` и `Error` в иерархии `Throwable` и приведите по одному примеру ситуации для каждого.
3. Чем `preg_match` отличается от `preg_match_all`, и в каком из заданий выше (валидация промокода) было бы бессмысленно использовать `preg_match_all` вместо `preg_match`?
4. Почему `session_start()` обязательно должен быть вызван раньше любого вывода в браузер?

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

5. Что произойдёт при попытке передать в вариадический параметр `function f(int ...$x)` строку вместо числа при включённом `declare(strict_types=1)`?

Критерии оценки теоретической части

По 2 балла за вопрос при полном и корректном ответе, 1 балл — при частично верном. Максимум — 10 баллов.

Итоговая шкала

Баллы (практика 12 + теория 10, максимум 22)	Оценка
20-22	Отлично
15-19	Хорошо
11-14	Удовлетворительно
< 11	Неудовлетворительно

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Контрольная работа №4. Основы Laravel: мини-CRUD «Отзывы к товару»

Охватывает: лекции 1-12 раздела «3. Основы Laravel» и лабораторные ЛР 3.1-3.11

Формат: мини-CRUD с нуля, ограничено по времени + теория

Длительность: 120 минут

Часть 1. Практическое задание

В отдельном новом Laravel-проекте (либо в изолированной ветке `control-work-3` проекта `shop-laravel` — по указанию преподавателя) реализовать модуль **«Отзывы к товару»**:

1. Миграция и модель `Review`: поля `product_id` (внешний ключ), `author_name`, `rating` (целое число 1-5), `comment` (текст), стандартные `timestamps`.
2. Отношение `Product::reviews(): HasMany` и `Review::product(): BelongsTo`.
3. `ReviewController` (resource или api-resource — по указанию преподавателя), тонкий, использующий сервис `ReviewService`.
4. `ReviewService` с методами `getForProduct(Product $product)`, `create(Product $product, array $data)`.
5. `StoreReviewRequest` — Form Request с валидацией: `author_name` обязательно, `rating` — целое от 1 до 5, `comment` — минимум 10 символов.
6. Минимум 2 маршрута: публичный `GET /products/{product}/reviews` (список отзывов) и защищённый `POST /products/{product}/reviews` (добавление отзыва, доступно только аутентифицированным пользователям через `auth` или `auth:sanctum` — по указанию преподавателя).
7. Вывод списка отзывов на странице товара через Blade с использованием `@forelse`, безопасным выводом (`{{ }}`) и хотя бы одним компонентом (`<x-review-card>` или аналог).
8. Использование eager loading (`with('reviews')`) там, где отзывы выводятся в списке товаров, чтобы избежать N+1.

Критерии оценки практической части

Критерий	Баллы
Корректная миграция и отношения `Product/Review`	3
Контроллер тонкий, использует сервис	2
Form Request с точными правилами валидации	3
Публичный и защищённый маршруты работают корректно	3
Blade-вывод безопасен, использует компонент	2
Eager loading применён корректно	2

Автоматическая проверка практической части (GitVerse Smart Class)

Тест намеренно не завязывается на выбор `auth` vs `auth:sanctum` (это на усмотрение преподавателя по формулировке задания): `postJson()` выставляет заголовок `Accept: application/json`, а стандартный обработчик Laravel при таком заголовке

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

возвращает `401` для неаутентифицированного запроса независимо от того, какой из двух guard'ов защищает маршрут.

Добавьте `tests/Feature/SmartClass/Kr3Test.php`:

```
<?php
// tests/Feature/SmartClass/Kr3Test.php — автопроверка КР 3

namespace Tests\Feature\SmartClass;

use App\Models\Product;
use App\Models\User;
use Illuminate\Foundation\Testing\RefreshDatabase;
use Tests\TestCase;

class Kr3Test extends TestCase
{
    use RefreshDatabase;

    private function makeProduct(): Product
    {
        return Product::create(['name' => 'Тест-товар', 'price' => 100,
'stock' => 5]);
    }

    public function test_public_can_list_reviews_for_product(): void
    {
        $product = $this->makeProduct();

        $response = $this->getJson("/products/{$product->id}/reviews");

        $response->assertOk();
    }

    public function test_guest_cannot_post_review(): void
    {
        $product = $this->makeProduct();

        $response = $this->postJson("/products/{$product->id}/reviews", [
            'author_name' => 'Иван',
            'rating' => 5,
            'comment' => 'Отличный товар, всем советую',
        ]);

        $response->assertStatus(401);
    }

    public function test_authenticated_user_can_post_valid_review(): void
    {
        $product = $this->makeProduct();
        $user = User::factory()->create();
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
        $response = $this->actingAs($user)->postJson("/products/{$product->id}/reviews", [
            'author_name' => 'Иван',
            'rating' => 5,
            'comment' => 'Отличный товар, всем советую',
        ]);

        $response->assertSuccessful();
        $this->assertDatabaseHas('reviews', ['product_id' => $product->id, 'author_name' => 'Иван']);
    }

    public function test_invalid_review_is_rejected_with_422(): void
    {
        $product = $this->makeProduct();
        $user = User::factory()->create();

        $response = $this->actingAs($user)->postJson("/products/{$product->id}/reviews", [
            'author_name' => '',
            'rating' => 9,
            'comment' => 'коротко',
        ]);

        $response->assertStatus(422);
    }

    public function test_product_reviews_relation_works(): void
    {
        $product = $this->makeProduct();
        $product->reviews()->create([
            'author_name' => 'Мария',
            'rating' => 4,
            'comment' => 'Хороший товар, но доставка долгая',
        ]);

        $this->assertCount(1, $product->fresh()->reviews);
    }
}
```

И job:

```
  kr-3:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: shivammathur/setup-php@v2
        with:
          php-version: '8.5'
      - name: Автопроверка (SmartClass)
        run: |
          source .gitverse/scripts/grade.sh
```


ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

```
          composer install --no-interaction > /dev/null
          cp .env.example .env && php artisan key:generate > /dev/null
          touch database/database.sqlite
          check_exit "Модуль «Отзывы»: маршруты, валидация, отношения
(реальный HTTP-тест)" \
            'php artisan test --filter=Kr3Test' 15
          report
```

Не проверяется автоматически: использован ли именно Blade-компонент (`<x-review-card>` или аналог) при выводе отзывов и применён ли `with('reviews')` в каталоге товаров — оба пункта касаются конкретной вёрстки и мест использования, которые надёжнее оценить чтением кода, чем HTTP-тестом на уровне API.

Часть 2. Теоретические вопросы

1. Объясните жизненный цикл запроса в Laravel от `public/index.php` до возврата ответа — назовите все ключевые этапы.
2. Почему регистрация middleware в `app/Http/Kernel.php` считается устаревшим подходом в используемой версии Laravel, и где это делается правильно?
3. Как Route Model Binding упрощает получение конкретного товара по ID из URL по сравнению с ручным `Product::findOrFail($id)`?
4. Что произойдёт при попытке отправить POST-запрос на защищённый `auth:sanctum` маршрут без токена — какой код ответа и почему?
5. Почему для отношения `Product::reviews()` используется `hasMany`, а не `belongsToMany`, если каждый отзыв относится только к одному товару?

Критерии оценки теоретической части

По 2 балла за вопрос при полном и корректном ответе, 1 балл — при частично верном. Максимум — 10 баллов.

Итоговая шкала

Баллы (практика 15 + теория 10, максимум 25)	Оценка
22-25	Отлично
17-21	Хорошо
12-16	Удовлетворительно
< 12	Неудовлетворительно

5.3.4 Лабораторные и практические работы

Лабораторные работы выполняются в лаборатории программного обеспечения на компьютерах с установленными PHP 8.5 (CLI), Composer, Git, Node.js, СУБД SQLite и MariaDB и средой VS Code; результаты размещаются в личном репозитории на платформе GitVerse. Методические указания, исходные заготовки проектов и автоматические проверки (GitVerse Actions) размещены в курсе Moodle «Основы алгоритмизации и программирования». По итогам каждой работы обучающийся оформляет отчёт (описание хода работы, блок-схемы, фрагменты кода, результаты тестов с пояснениями) и сдаёт запрос на слияние (Pull Request) преподавателю. Работы №1–8 (алгоритмизация, Git) и №15–22 (алгоритмы и структуры данных)

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

разработаны для данной программы; работы №9–14, 23–34 основаны на лабораторных работах курса Moodle.

Перечень лабораторных работ (72 часа, 36 работ по 2 часа)

№	Лабораторная работа	Цель работы	Источник
1	Подготовка среды разработки: PHP CLI, VS Code, запуск и отладка первой программы	Подготовить рабочее место: установить и проверить PHP 8.5 (CLI), настроить VS Code и расширения, запустить первую программу, освоить вывод ошибок и пошаговую отладку.	новая работа
2	Первый репозиторий, коммиты, ветвление и слияние	Создать локальный репозиторий, выполнять атомарные коммиты по формату «тип: описание», работать с ветками, выполнять слияние и разрешать простой конфликт.	Moodle: ЛР 0.1, ЛР 0.2
3	Публикация в GitVerse, запрос на слияние (Pull Request) и код-ревью	Опубликовать проект в GitVerse (SSH-ключ или токен), оформить запрос на слияние по шаблону, выполнить код-ревью чужого запроса; не публиковать секреты.	Moodle: ЛР 0.3, ЛР 0.4
4	Блок-схемы линейных и разветвляющихся алгоритмов по ГОСТ 19.701-90	Научиться строить схемы алгоритмов по ГОСТ 19.701-90 для линейных и разветвляющихся алгоритмов, проверять их на тестовых наборах.	новая работа
5	Циклические алгоритмы: блок-схемы и таблицы трассировки	Составлять блок-схемы циклических алгоритмов (с предусловием, постусловием, параметром), выполнять трассировку по таблице значений переменных.	новая работа
6	Псевдокод, проверка корректности алгоритма и перенос блок-схемы в код PHP	Записывать алгоритм на псевдокоде, обосновывать корректность (инвариант цикла, граничные значения), переносить схему в код PHP и сопоставлять конструкции.	новая работа
7	Оценка сложности алгоритмов: подсчёт операций и O-нотация	Подсчитывать число элементарных операций в циклах и рекурсии, определять класс сложности O(1), O(log n), O(n), O(n log n), O(n^2) для фрагментов кода.	новая работа
8	Экспериментальное измерение времени работы алгоритмов	Измерять время работы алгоритмов функцией hrtime() на входах разного размера, строить таблицу и график зависимости, сопоставлять результат с теоретической оценкой.	новая работа
9	Витрина магазина: типы данных, операции и вывод данных	Освоить типы данных, переменные, операции и вывод PHP на примере витрины магазина; применять приведение типов и строгое сравнение.	Moodle: ЛР 1.1
10	Условные операторы и циклы: фильтрация и сортировка списка товаров	Применять условные операторы и циклы для фильтрации и сортировки списка товаров; использовать match и вложенные циклы.	Moodle: ЛР 1.2
11	Массивы и многомерные массивы: группировка и агрегирование данных	Работать с индексными, ассоциативными и многомерными массивами: группировать, суммировать, искать, сортировать; выполнять операции над множествами.	новая работа
12	Строки и регулярные выражения: валидация промокода и телефона	Проверять строки регулярными выражениями (промокод формата XXX-0000, телефон +7XXXXXXXXXX), использовать многобайтовые строковые функции.	Moodle: ЛР 1.9
13	Файлы: чтение и запись CSV и JSON, журнал заказов	Читать и записывать файлы последовательного доступа (CSV, JSON), вести журнал заказов, обрабатывать ошибки доступа к файлам.	новая работа
14	Рефакторинг кода в функции	Выделить повторяющиеся и самостоятельные фрагменты кода в типизированные функции, оформить их в отдельном файле и подключить.	Moodle: ЛР 1.3

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

№	Лабораторная работа	Цель работы	Источник
15	Рекурсия и обработка вложенных структур	Реализовать рекурсивные алгоритмы (факториал, числа Фибоначчи, ханойские башни, обход дерева категорий), сопоставить с итеративными вариантами и мемоизацией.	новая работа
16	Линейный и бинарный поиск	Реализовать линейный и бинарный поиск, подсчитать число сравнений на массивах разного размера и подтвердить оценки O(n) и O(log n).	новая работа
17	Квадратичные сортировки: пузырьком, выбором, вставками	Реализовать сортировки пузырьком, выбором и вставками, подсчитать числа сравнений и обменов, сравнить на упорядоченных, обратных и случайных данных.	новая работа
18	Быстрая сортировка и сортировка слиянием	Реализовать сортировку слиянием и быструю сортировку, сравнить их с sort() и usort() по времени, проверить устойчивость.	новая работа
19	Стек: проверка скобок и вычисление выражения в обратной польской записи	Реализовать стек на массиве и на SplStack, решить задачи проверки скобочной последовательности и вычисления выражения в обратной польской записи.	новая работа
20	Очередь, дек и односвязный список	Реализовать очередь (SplQueue), дек и односвязный список на объектах; выполнить вставку, удаление и поиск, сопоставить сложность с массивом.	новая работа
21	Хеш-таблицы и множества: подсчёт частот и поиск дубликатов	Использовать ассоциативные массивы как хеш-таблицы: подсчитать частоты слов, найти дубликаты за O(n), выполнить операции над множествами.	новая работа
22	Двоичное дерево поиска и обход графа	Реализовать вставку, поиск и обходы двоичного дерева поиска; представить граф списком смежности и выполнить обход в ширину и в глубину.	новая работа
23	ООП-модель магазина: классы, инкапсуляция, наследование, полиморфизм	Построить ООП-модель магазина (товар, корзина, заказ), применяя инкапсуляцию, наследование, переопределение методов и полиморфизм.	Moodle: ЛР 1.6
24	Интерфейсы, абстрактные классы, перечисления и современный синтаксис PHP 8.x	Использовать интерфейсы, абстрактные классы, перечисления (enum), readonly-свойства и конструкторное продвижение; заменить «магические» значения.	Moodle: ЛР 4.1
25	Исключения и отладка: оформление заказа	Реализовать собственные исключения и их обработку при оформлении заказа; отладить программу с помощью Xdebug и журналов.	Moodle: ЛР 1.8
26	Модульное тестирование алгоритмов (Pest)	Разработать модульные тесты Pest для алгоритмов из лабораторных работ №16–22 (нормальные, граничные и ошибочные случаи), запустить их в GitVerse Actions.	новая работа
27	Composer, автозагрузка PSR-4 и стиль кода PSR-12	Настроить Composer-проект с автозагрузкой PSR-4, подключить пакет, привести код к стилю PSR-12 и проверить его phpcs или Laravel Pint.	Moodle: ЛР 2.1, ЛР 2.2
28	Чистый код, SOLID-рефакторинг и внедрение зависимостей	Устранить дублирование и «магические числа», применить принципы SOLID (стратегия скидок, разделение ответственности), реализовать ручной контейнер внедрения зависимостей.	Moodle: ЛР 2.3, ЛР 2.4, ЛР 2.6
29	Проект Laravel: структура, маршруты и контроллеры	Создать проект Laravel, изучить структуру каталогов и жизненный цикл запроса, описать маршруты магазина и тонкие контроллеры.	Moodle: ЛР 3.1, ЛР 3.2, ЛР 3.3
30	Blade: директивы, макеты и	Построить каталог на директивах Blade, выделить	Moodle: ЛР

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

№	Лабораторная работа	Цель работы	Источник
	компоненты	макет и повторно используемые компоненты, обеспечить экранирование вывода.	3.4, ЛР 3.5
31	Хранение данных: PDO, Eloquent ORM и отношения между моделями	Подключить базу данных (SQLite, MySQL/MariaDB), сравнить PDO с подготовленными выражениями и Eloquent ORM, описать отношения моделей, устранить проблему N+1.	Moodle: ЛР 1.7, ЛР 3.6, ЛР 3.7
32	Валидация данных и аутентификация: Form Requests и Laravel Breeze	Реализовать валидацию через Form Requests, регистрацию и вход покупателей средствами Laravel Breeze; проверить защиту от CSRF.	Moodle: ЛР 3.8, ЛР 3.9
33	Сервис-слой, DTO и API Resources	Вынести бизнес-логику в сервисы, описать DTO заказа (Spatie Laravel Data) и API Resources для витрины.	Moodle: ЛР 4.2, ЛР 4.3, ЛР 4.4
34	Коллекции для отчётов и административная панель Orchid	Построить отчёты на коллекциях Laravel (map, filter, groupBy, sum) и добавить в административную панель Orchid экран с таблицей и формой.	Moodle: ЛР 4.6, ЛР 4.7
35	ИИ-ассистенты: генерация кода PHP и тестов с последующей проверкой	Формулировать корректные запросы к GigaChat, GigaCode и Алиса AI для получения функций, классов и тестов на PHP; проверять результат запуском, тестами и документацией php.net.	новая работа
36	ИИ-ассистенты: ревью, поиск ошибок и рефакторинг кода	Использовать ИИ-ассистент для ревью и рефакторинга кода: найти ошибки, уязвимости и неэффективные алгоритмы, сопоставить предложения с ручным анализом и принять обоснованное решение.	новая работа

Критерии оценки лабораторной работы

Оценка	Показатели
«5»	Все задания выполнены самостоятельно и верно; блок-схемы соответствуют ГОСТ 19.701-90; код соответствует PSR-12 и проходит автоматические проверки и модульные тесты; отчёт и запрос на слияние оформлены полностью; обучающийся объясняет принцип работы алгоритма, его сложность и результат; при использовании ИИ-ассистента представлены запросы, выявленные недостатки ответа и исправленный вариант с указанием использования ИИ.
«4»	Работа выполнена полностью, есть неточности в оформлении, стиле кода или пояснениях; обучающийся устраняет замечания после наводящих вопросов.
«3»	Работа выполнена не полностью (не менее 70 % заданий) или с существенными ошибками; часть тестов не проходит; пояснения неполные.
«2»	Работа не выполнена или выполнена менее чем на 70 %; обучающийся не может объяснить программу и результат; код получен от ИИ-ассистента или другого лица без понимания и проверки.

5.3.5 Компьютерные программы для контроля знаний студентов

Комплект тестов и банки вопросов по дисциплине размещены в системе дистанционного обучения колледжа (Moodle) и импортируются в формате Moodle XML. Контрольные работы выполняются письменно и на компьютере в изолированном репозитории GitVerse; экзамен проводится устно с выполнением практического задания на бумаге или компьютере. Использование ИИ-ассистентов на контрольных работах, тестировании и экзамене не допускается. Автоматические

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

проверки лабораторных и контрольных работ (GitVerse Actions) используются как вспомогательное средство; итоговая оценка выставляется преподавателем после защиты работы.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
1 подпись
Сведения о подписях — на листе подписей. Проверка подлинности — по QR-коду.
ГАПОУ СО «Нижнетагильский торгово-экономический колледж»

Лист подписей

ГАПОУ СО «Нижнетагильский торгово-экономический колледж»
SHA-256 исходного документа: 928378ac0d58d86587e04a8313ea30e970366f99f7dcea4be045a28d3bd838ac
Документ: 54 страницы
Проверка подписей: https://sig.nttek.ru/v/QjTgkexbQ9Scd4QUAaQgk6NbyAqC83LWJgMJj2g3fKhfDKM5
Печатная форма сформирована: 13.10.2026 10:28

№	Роль в подписи	Подписант	Сертификат	Дата и время подписания	Результат проверки
1	Утверждаю	Дмитриева Марина Георгиевна Заместитель директора по учебной работе	№ 7E9C18960643A37EF2C929CD1E513CCD, действителен с 08.10.2026 по 08.10.2027	13.10.2026 10:27:33	**подпись действительна** роль и должность подтверждены подписью

Печатная форма не подписывается и служит для удобства чтения. Юридическую силу имеют исходный PDF-документ и файлы электронных подписей (.sig). Подпись — усиленная неквалифицированная электронная подпись (ч. 3 ст. 5 Федерального закона № 63-ФЗ), выданная удостоверяющим центром колледжа.
Время указано по часовому поясу колледжа (Asia/Yekaterinburg, UTC+5).