Министерство образования Иркутской области
Государственное бюджетное профессиональное образовательное учреждение Иркутской области
«Иркутский авиационный техникум»

УТВЕРЖДАЮ
Директор
ГБПОУИО «ИАТ»
_____ Якубовский А.Н.
«29» мая 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ОП.06 Основы алгоритмизации и программирования

специальности

09.02.11 Разработка и управление программным обеспечением

Иркутск, 2026

Рассмотрена
цикловой комиссией

№	Разработчик ФИО
1	Денисова Анна Степановна
2	Тимофеев Кирилл Сергеевич

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Область применения фонда оценочных средств (ФОС)

ФОС по дисциплине является частью программы подготовки специалистов среднего звена по специальности 09.02.11 Разработка и управление программным обеспечением

1.2. Место дисциплины в структуре ППССЗ:

ОП.00 Общепрофессиональный цикл.

1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины

Результаты освоения дисциплины	№ результата	Формируемый результат
Знать	1.1	Порядок выполнения инструкций программного кода
	1.2	Назначение операторов ветвления
	1.3	Назначение циклических операторов
	1.4	Перечень операций с массивами
	1.5	Понятия в программировании: функция, параметр функции
	1.6	Области видимости данных
	1.7	Понятие объекта и класса
	1.8	Принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, абстракция, наследование
	1.9	Правила отладки программ
	1.10	Правила объявления и взаимодействие с переменными
	1.11	Классификацию типов данных
	1.12	Понятие линейных конструкций
	1.13	Шаблоны проектирования в объектно-ориентированном программировании
Уметь	2.1	анализировать условие задачи для выявления входных и выходных данных
	2.2	представлять алгоритм в виде блок-схемы

	2.3	работать с одномерными массивами
	2.4	реализовывать программные функции
	2.5	реализовывать классы и объекты с учётом правил объектно-ориентированного программирования
	2.6	выполнять отладку программного кода
	2.7	использовать операторы ветвления
	2.8	использовать операторы цикла
	2.9	работать с двумерными массивами

1.4. Формируемые компетенции:

ОК.1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК.2 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК.3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях
ОК.5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК.6 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК.7 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК.8 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК.9 Пользоваться профессиональной документацией на государственном и иностранном языках
ПК.2.2 Разрабатывать модули программного обеспечения
ПК.2.4 Выполнять тестирование и отладку программного обеспечения

2. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

2.1 Текущий контроль (ТК) № 1 (45 минут)

Тема занятия: 1.2.4.Работа с переменными и типами данных в PHP.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с использованием ИКТ
Дидактическая единица: 1.1 Порядок выполнения инструкций программного кода
Занятие(-я):
1.2.2.Базовые принципы построения алгоритмов.
Задание №1 (5 минут)
Ответьте на следующие вопросы:
1. Что такое «поток выполнения» (execution flow) программы?
2. В каком порядке выполняются инструкции в линейном алгоритме?

3. Может ли порядок выполнения строк кода измениться без использования циклов и условий?

Оценка	*Показатели оценки*
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица: 1.5 Понятия в программировании: функция, параметр функции
Занятие(-я):
1.1.2.Методы анализа постановки задачи.
Задание №1 (5 минут)
Дайте определения понятиям «функция», «параметр» и «аргумент».В чем разница между параметром и аргументом?

Оценка	*Показатели оценки*
5	Студент дает **точные определения** всех трех понятий без существенных ошибок, объяснив разницу между параметром и аргументом.
4	Студент дает **в целом верные определения**, но с небольшими неточностями в формулировках.
3	П**риблизительные определения**: студент путает функцию с процедурой/методом, не может четко сформулировать, что такое параметр.

Дидактическая единица: 1.10 Правила объявления и взаимодействие с переменными

Занятие(-я):

1.2.1.Работа с переменными и хранением данных.

Задание №1 (5 минут)

Ответьте на следующие вопросы: Какие существуют базовые правила именования переменных? Что происходит с переменной в оперативной памяти при повторном присваивании ей нового значения?

Оценка	***Показатели оценки***
5	Даны подробные ответы на оба вопроса: перечислены все базовые правила именования переменных (начальный символ, допустимые символы, запрет на ключевые слова, регистрозависимость) и точно описан механизм перезаписи значения в оперативной памяти с упоминанием освобождения памяти сборщиком мусора.
4	Представлены ответы на оба вопроса, но с небольшими неточностями или неполно: перечислены основные правила именования переменных и описан процесс изменения значения переменной, однако без упоминания деталей работы с памятью или сбора мусора.
3	Дан развернутый ответ только на один из вопросов ИЛИ ответы на оба вопроса даны поверхностно без конкретизации правил именования и понимания механизма перезаписи значения в памяти.

Дидактическая единица: 2.1 анализировать условие задачи для выявления входных и выходных данных

Занятие(-я):

1.1.3.Методы анализа постановки задачи.

Задание №1 (15 минут)

Задание: Заполнить таблицу анализа задачи. Для каждой задачи выписать: 1) Что дано (Вход), 2) Что нужно получить (Выход), 3) Ограничения/допущения

Параметры задач:

1. *Математическая:* «Дан массив оценок студента. Нужно вычислить его средний балл и узнать, есть ли у него двойки».
2. *Бытовая/Алгоритмическая:* «Система умного дома должна включать отопление, если температура на улице падает ниже заданного порога, и выключать, если превышает».

3. *Экономическая:* «Рассчитать итоговую стоимость корзины товаров в интернет-магазине с учетом скидки, если сумма покупок превышает N рублей».

Оценка	***Показатели оценки***
5	Таблица анализа задачи заполнена полностью и верно для всех трех задач: точно выделены все входные и выходные данные, а также корректно сформулированы ограничения и допущения без логических ошибок.
4	Таблица заполнена для всех трех задач, но допущены незначительные неточности: упущены некоторые входные данные или ограничения, либо некорректно сформулированы выходные данные для одной из задач.
3	Таблица заполнена только для одной-двух задач ИЛИ анализ выполнен поверхностно: полностью отсутствуют ограничения и допущения, либо грубо перепутаны входные и выходные данные.

Дидактическая единица: 2.2 представлять алгоритм в виде блок-схемы
Занятие(-я):
1.2.3.Графическое описание алгоритмов.
Задание №1 (15 минут)
Задание: Построить блок-схему алгоритма на бумаге или в графическом редакторе.

Параметры задания: Алгоритм «Вычисление значения кусочно-заданной функции (если x < 0, то y = x^2, иначе y = 2x)». Блок-схема должна содержать не менее 4-5 блоков, включая блок условия (ромб) с двумя ветками (Да/Нет).

Оценка	***Показатели оценки***
5	Блок-схема построена полностью корректно, содержит не менее 4-5 стандартных блоков (включая ромб с двумя ветками «Да/Нет») и точно отражает алгоритм вычисления заданной кусочно-заданной функции.
4	Блок-схема отражает верную логику алгоритма и содержит блок условия с двумя ветками, но допущены незначительные недочеты: отсутствуют блоки начала/конца, использованы нестандартные символы или есть мелкие ошибки в направлениях связей.
3	Блок-схема содержит грубые логические ошибки (неверные условия или формулы), отсутствует одна из веток условия, либо не соблюдены минимальные требования к количеству и типу используемых графических блоков.

2.2 Текущий контроль (ТК) № 2 (45 минут)

Тема занятия: 2.1.5.Решение задач с использованием условных конструкций.
Метод и форма контроля: Практическая работа (Информационно-аналитический)
Вид контроля: Практическая работа с использованием ИКТ
Дидактическая единица: 1.2 Назначение операторов ветвления
Занятие(-я):
2.1.1.Организация выбора действий в программе.

Задание №1 (10 минут)

Ответьте на следующие вопросы:

1. Что такое операторы ветвления и для чего они используются в программировании?
2. Какие виды операторов ветвления существуют в PHP (if, if-else, if-elseif-else, switch)?
3. В чем разница между простой и полной формой условного оператора?

Оценка	*Показатели оценки*
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица: 2.7 использовать операторы ветвления
Занятие(-я):
2.1.2.Построение программ с условными операторами.
2.1.3.Построение программ с условными операторами.
2.1.4.Реализация условных операторов if-else и switch.

Задание №1 (35 минут)

Создайте программу, которая определяет, является ли число четным или нечетным используя тернарный оператор. Выведите соответствующее сообщение.

Оценка	*Показатели оценки*
5	Программа работает корректно, тернарный оператор использован правильно, вывод сообщения соответствует результату проверки.
4	Программа работает, тернарный оператор использован, но допущены незначительные недочеты в коде или выводе сообщений.
3	Тернарный оператор использован неверно или заменен на условный оператор if, программа работает некорректно или содержит синтаксические ошибки.

2.3 Текущий контроль (ТК) № 3 (45 минут)

Тема занятия: 3.1.4.Создание пользовательских программных модулей.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 1.6 Области видимости данных
Занятие(-я):
3.1.2.Использование локальных и глобальных данных.

Задание №1 (10 минут)

Фрагмент PHP-кода, в котором используется глобальная переменная и локальная переменная с таким же именем внутри функции (эффект «затенения»). *Параметры задания:*

```php
<?php
$x = 10;
function change_x() {
    $x = 5;
    echo "Внутри функции: " . $x . "\n";
}
change_x();
echo "Снаружи функции: " . $x . "\n";
?>
```

Что выведет программа? Как изменить код внутри функции так, чтобы она реально изменила значение глобальной переменной $x?

Оценка	***Показатели оценки***
5	Верно предсказан вывод программы (5, затем 10). Точно объяснена разница между локальной и глобальной областью видимости в PHP. Самостоятельно и без ошибок продемонстрирован способ изменения глобальной переменной из функции.
4	Верно предсказан вывод программы и понята суть локальных/глобальных переменных, но допущена небольшая синтаксическая ошибка при демонстрации способа изменения глобальной переменной.
3	Неверно предсказан вывод программы (студент считает, что локальное присваивание изменит глобальную переменную, и ответит «5, затем 5») ИЛИ не может объяснить разницу между областями видимости без наводящих вопросов преподавателя.

Дидактическая единица: 1.11 Классификацию типов данных
Занятие(-я):
1.1.1.Основы использования типов данных в программировании.
Задание №1 (10 минут)
Параметры задания: Список: 42, 3.14, "True", false, [1, 2, 3], null, "100". *Вопросы:*

1. Определите тип каждого значения (скалярный или составной).
2. Что произойдет в PHP при попытке сложить целое число 5 и строку "5" ($result = 5 + "5";)? Чему будет равен $result и какой у него тип?

Оценка	***Показатели оценки***
5	Точно определены типы всех значений, верно разделены скалярные (int, float, string, bool, null) и составные (array) типы. Правильно объяснено поведение PHP при сложении 5 + "5" (сработает неявное приведение типов / type juggling, результат будет 10 типа int).
4	Правильно определены базовые типы, но допущена ошибка в классификации (например, null назван составным типом) ИЛИ студент знает, что строка превратится в число, но ошибается в конечном типе результата (называет его float или string).
3	Допущены грубые ошибки в определении типов (например, строка "True" названа логическим типом bool, а "100" - числовым int) ИЛИ студент считает, что в PHP сложение числа и строки вызовет фатальную ошибку (как в строго типизированных языках), не зная о механизме приведения типов.

Дидактическая единица: 1.12 Понятие линейных конструкций
Занятие(-я):
1.1.4.Последовательное выполнение программных команд.
1.1.5.Графическое описание алгоритмов.
Задание №1 (15 минут)
Параметры задания: «Поменять местами значения двух переменных $a и $b, используя третью вспомогательную переменную $c». *Требование:* Написать последовательный PHP-код и нарисовать соответствующую блок-схему.

Оценка	***Показатели оценки***
5	Написан корректный линейный PHP-код без логических ошибок (с использованием $, ;), построена полная блок-схема со всеми стандартными блоками (начало/конец, ввод/вывод, процесс) и строгой последовательностью связей.

4	Код написан верно, но в блок-схеме допущены незначительные недочеты (например, отсутствуют блоки начала/конца, использован нестандартный шрифт/форма блока или не подписаны линии потоков).
3	В коде допущена логическая ошибка (например, потеря значения при присваивании $a = $b; до использования $a в $c), ИЛИ блок-схема не соответствует коду, либо отсутствуют обязательные графические элементы.

Дидактическая единица: 2.4 реализовывать программные функции
Занятие(-я):
3.1.3.Создание пользовательских программных модулей.
Задание №1 (10 минут)
Напишите PHP-функцию calculate_discount($price, $percent), которая принимает стоимость товара и процент скидки, а возвращает итоговую стоимость товара со скидкой. Как вызвать эту функцию и сохранить результат в переменную? Что произойдет, если передать в нее строку "abc" вместо числа (в контексте PHP 8 и выше)?

Оценка	***Показатели оценки***
5	Функция написана синтаксически верно, корректно использует параметры (с $), содержит оператор return для возврата результата. Студент уверенно отвечает на вопросы о вызове функции и знает, что передача нечисловой строки "abc" в математической операции в PHP 8+ вызовет ошибку TypeError (или приведет к 0 в старых версиях, если студент это уточнит).
4	Функция работает и возвращает верный результат, но допущены мелкие стилистические ошибки (плохие имена параметров) ИЛИ студент затрудняется ответить на вопрос о поведении PHP при передаче некорректных строковых данных в математическую операцию.
3	Функция не возвращает значение (использует echo вместо return), ИЛИ содержит синтаксические ошибки (забыты знаки $ перед переменными, пропущены точки с запятой), из-за которых код не может быть выполнен.

2.4 Текущий контроль (ТК) № 4 (45 минут)

Тема занятия: 4.2.2.Обработка табличных структур данных.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ

Дидактическая единица: 1.3 Назначение циклических операторов
Занятие(-я):
2.2.1.Организация повторяющихся вычислительных процессов.
Задание №1 (5 минут)
Ответьте на следующие вопросы:

1. В чем заключается основное назначение циклических операторов? В каких случаях их использование обязательно, а в каких - просто сокращает объем кода?
2. В чем концептуальная разница между циклами for, while и do-while? Приведите пример ситуации, когда лучше использовать do-while, а не while.
3. Что такое «бесконечный цикл», как его можно намеренно создать в PHP и как из него выйти до завершения программы?

Оценка	***Показатели оценки***
5	Студент четко объясняет, что циклы нужны для многократного повторения блока кода без его дублирования. Точно описывает разницу между циклами (например, do-while гарантирует минимум одно выполнение тела цикла). Знает, как создать бесконечный цикл (while(true)) и как его прервать (break).
4	Студент понимает назначение циклов и может сравнить for и while, но затрудняется с примером для do-while или не может четко объяснить механизм прерывания бесконечного цикла без наводящих вопросов.
3	Студент дает размытые ответы (например, «циклы нужны, чтобы программа работала дольше»), путает синтаксис и назначение разных типов циклов, не понимает концепцию бесконечного цикла.

Дидактическая единица: 1.4 Перечень операций с массивами
Занятие(-я):
4.1.1.Способы обработки наборов данных.
Задание №1 (10 минут)
Ответьте на следующие вопросы:

1. Что такое массив в PHP? Чем индексированный (числовой) массив отличается от ассоциативного?
2. Назовите и опишите назначение 4-5 стандартных встроенных функций PHP для работы с массивами (например, для добавления, удаления, сортировки, поиска или подсчета элементов).
3. Какими способами можно перебрать все элементы массива в PHP? В чем

преимущество конструкции foreach перед обычным циклом for?

Оценка	***Показатели оценки***
5	Студент дает точные определения, четко разделяет индексированные и ассоциативные массивы. Свободно называет встроенные функции (array_push, sort, in_array, count и тому подобное) и объясняет, что foreach удобнее и безопаснее для перебора, так как не требует ручного управления счетчиком и индексами.
4	Студент понимает разницу между типами массивов, называет основные функции, но может забыть 1-2 из них или неточно сформулировать их синтаксис. Понимает преимущество foreach, но объясняет его общими словами.
3	Студент путает массивы с обычными переменными, не знает разницы между индексированным и ассоциативным массивом. Не может назвать ни одной встроенной функции, предлагая писать все операции (поиск, сортировку) вручную через циклы.

Дидактическая единица: 2.3 работать с одномерными массивами
Занятие(-я):
4.1.2.Обработка линейных массивов данных.
4.1.3.Алгоритмы обработки одномерных массивов.
Задание №1 (10 минут)
Написать PHP-скрипт для анализа набора данных. *Параметры задания:*
Сгенерировать одномерный массив из 15 случайных целых чисел в диапазоне от -50 до 50. Найти и вывести на экран:

1. Максимальный и минимальный элементы массива.
2. Сумму всех положительных элементов.
3. Индекс первого найденного отрицательного элемента.

Оценка	***Показатели оценки***
5	Скрипт успешно выполняется. Массив сгенерирован верно. Все три метрики вычислены абсолютно точно. Код оптимизирован (например, поиск максимума/минимума и суммы выполнен за один проход по массиву).
4	Скрипт работает, но допущена небольшая логическая ошибка в вычислениях (например, посчитана сумма модулей положительных чисел, или найден индекс *последнего* отрицательного элемента вместо первого).

3	Студент не может корректно сгенерировать массив случайных чисел ИЛИ не может реализовать алгоритм поиска (например, просто выводит весь массив через print_r без какой-либо аналитической обработки).

Дидактическая единица: 2.8 использовать операторы цикла
Занятие(-я):
2.2.2.Создание циклических алгоритмов.
2.2.3.Создание циклических алгоритмов.
2.2.4.Программирование циклов с параметром и условием.
2.2.5.Решение задач с вложенными циклами.
Задание №1 (10 минут)
Решить две задачи на разные типы циклов. *Параметры задания:*

1. С помощью **вложенного цикла for** вывести на экран прямоугольный треугольник из символов * высотой 6 строк (в первой строке 1 звезда, во второй 2, и тому подобное).
2. С помощью **цикла while** вычислить сумму ряда 1+1/2+1/4+1/8+...1+1/2+1/4+1/8+... до тех пор, пока добавляемое слагаемое не станет меньше 0.01. Вывести итоговую сумму.

Оценка	***Показатели оценки***
5	Обе задачи решены верно. Треугольник выведен с правильным форматированием. Цикл while корректно завершается при достижении заданной точности, математическая логика не нарушена.
4	Треугольник выведен, но имеет нарушения форматирования (например, лишние пробелы в конце строк). ИЛИ цикл while работает, но условие выхода задано неточно (например, < 0.1 вместо < 0.01), из-за чего сумма немного отличается от эталонной.
3	Вложенный цикл выводит не треугольник, а квадрат или прямую линию (ошибка в условии внутреннего цикла). ИЛИ цикл while зацикливается (бесконечный цикл) из-за того, что студент забыл обновлять знаменатель дроби.

Дидактическая единица: 2.9 работать с двумерными массивами
Занятие(-я):
4.2.1.Обработка табличных структур данных.
Задание №1 (10 минут)
Написать скрипт для обработки матрицы. *Параметры задания:* Создать двумерный

массив (матрицу) размером 4x4, заполненный случайными числами от 1 до 10.

1. Вывести матрицу на экран в виде аккуратной HTML-таблицы (<table>) или выровненного текстового блока.
2. Вычислить и вывести сумму элементов, находящихся на главной диагонали матрицы.

Оценка	***Показатели оценки***
5	Матрица корректно сгенерирована и красиво выведена в табличном виде (использованы вложенные циклы и теги <tr>, <td>). Сумма главной диагонали вычислена верно (использовано условие $i == $j или прямой проход по индексам).
4	Матрица выведена, но форматирование таблицы «плавает» (не заданы границы, нет отступов, или использован print_r вместо HTML-таблицы, хотя задание требовало визуального вывода). ИЛИ сумма посчитана верно, но для побочной диагонали вместо главной (ошибка в индексах).
3	Студент не понимает структуру двумерного массива (пытается обращаться к нему как к одномерному), не может вывести его в табличном виде и не может реализовать алгоритм обхода диагонали.

2.5 Текущий контроль (ТК) № 5 (45 минут)

Тема занятия: 5.2.2.Разработка объектно-ориентированных приложений.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 1.8 Принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, абстракция, наследование
Занятие(-я):
5.1.2.Основные механизмы объектно-ориентированного подхода.
Задание №1 (10 минут)
Ответьте на следующие вопросы:

1. Дайте краткое определение каждому из четырех принципов ООП: инкапсуляция, наследование, полиморфизм, абстракция.
2. Что такое модификаторы доступа (public, private, protected) и как они реализуют принцип инкапсуляции в PHP?
3. Приведите пример из реальной жизни или программирования, где наглядно проявляется принцип наследования и принцип полиморфизма.

Оценка	***Показатели оценки***

5	Студент дает точные и четкие определения всем четырем принципам. Отлично понимает разницу между модификаторами доступа и может объяснить, зачем скрывать данные (защита от некорректного изменения состояния объекта). Приводит уместные и понятные примеры.
4	Студент понимает суть принципов ООП, но допускает неточности в формулировках. Может назвать модификаторы доступа, но затрудняется объяснить разницу между protected и private. Примеры приводит, но они немного размыты.
3	Студент дает поверхностные ответы. Не может четко разграничить принципы, путает наследование и полиморфизм. Не знает модификаторов доступа или считает, что все свойства класса всегда должны быть публичными.

Дидактическая единица: 1.13 Шаблоны проектирования в объектно-ориентированном программировании

Занятие(-я):

5.2.1.Подходы к проектированию структуры программ.

Задание №1 (10 минут)

Ответьте на следующие вопросы:

1. Что такое шаблон проектирования (Design Pattern)? Зачем они нужны при разработке ПО?
2. Назовите и опишите суть 1-2 базовых порождающих или структурных шаблонов.
3. Какую проблему решает использование шаблонов проектирования по сравнению с написанием кода «в лоб»? Приведите пример одного поведенческого шаблона (например, «Стратегия» или «Наблюдатель») и опишите ситуацию в разработке, где его применение позволяет избавиться от громоздких конструкций if-else или switch

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица: 2.5 реализовывать классы и объекты с учётом правил объектно-ориентированного программирования

Занятие(-я):

5.1.3.Создание классов и объектов.

5.1.4.Реализация принципов объектно-ориентированного проектирования (наследование, инкапсуляция, полиморфизм).

Задание №1 (25 минут)

Написать PHP-скрипт, реализующий небольшую объектную модель. *Параметры задания:*

1. Создать базовый класс BankAccount (Банковский счет) с **приватным** свойством $balance (инкапсуляция). Реализовать публичные методы для пополнения, снятия средств и получения текущего баланса.
2. Создать дочерний класс SavingsAccount (Накопительный счет), который **наследуется** от BankAccount. Добавить ему свойство $interestRate (процентная ставка).
3. Реализовать **полиморфизм**: переопределить метод снятия средств в дочернем классе так, чтобы он запрещал снятие, если после операции баланс упадет ниже неснижаемого остатка (например, 100 рублей), и выводить соответствующее сообщение.
4. В основном коде программы создать объекты обоих классов, продемонстрировать их работу и показать разницу в поведении метода снятия средств.

Оценка	***Показатели оценки***
5	Код полностью работоспособен и соответствует всем требованиям. Свойство $balance объявлено как private, доступ к нему осуществляется только через геттеры/сеттеры или методы класса. Корректно использовано ключевое слово extends, вызван родительский конструктор (если он есть). Метод снятия средств успешно переопределен в дочернем классе (полиморфизм), логика проверки остатка работает верно.
4	Код работает и логика верна, но допущены нарушения принципов инкапсуляции (например, свойство $balance объявлено как public или protected вместо private). ИЛИ при переопределении метода в дочернем классе не вызван родительский метод через parent::, из-за чего дублируется код базовой логики.
3	Студент не может реализовать наследование (создает два независимых класса) ИЛИ не использует модификаторы доступа (весь код публичный). Код содержит синтаксические ошибки, не создает объекты или не демонстрирует разницу в работе методов (полиморфизм не показан).

2.6 Текущий контроль (ТК) № 6 (45 минут)

Тема занятия: 6.1.4.Тестирование программ.
Метод и форма контроля: Практическая работа (Опрос)
Вид контроля: Практическая работа с применением ИКТ
Дидактическая единица: 1.7 Понятие объекта и класса
Занятие(-я):
5.1.1.Основы объектной модели программирования.

Задание №1 (5 минут)

Ответьте на следующие вопросы:

1. Что такое класс и что такое объект? В чем разница между ними? Приведите простую аналогию из реальной жизни.
2. Чем свойство (атрибут) класса отличается от метода? Что такое $this и зачем оно нужно внутри методов?
3. Что такое инстанцирование (создание экземпляра)? Какой синтаксис используется в PHP для создания нового объекта на основе класса?

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица: 1.8 Принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, абстракция, наследование
Занятие(-я):

Задание №1 (10 минут)

Ответьте на следующие вопросы:

1. Как в PHP реализуется наследование? Что происходит, если дочерний класс переопределяет метод родительского класса, и как при этом вызвать исходный метод родителя?
2. В чем разница между абстрактным классом (abstract class) и интерфейсом (interface) в PHP? В каких случаях вы будете использовать интерфейс, а в каких - абстрактный класс?
3. Как полиморфизм связан с типизацией (type hinting) в PHP? Приведите пример, как функция может принимать объекты разных классов.

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица: 1.9 Правила отладки программ

Занятие(-я):

6.1.1.Методы поиска и устранения ошибок в коде.

Задание №1 (10 минут)

Ответьте на следующие вопросы:

1. Назовите три основных типа ошибок в программировании (синтаксические, ошибки времени выполнения, логические). Приведите пример каждого типа.
2. Какие встроенные средства PHP вы используете для быстрой отладки (вывода данных на экран)? В чем разница между echo, print_r() и var_dump()?
3. Чем использование отладчика (например, Xdebug с точками останова / breakpoints) принципиально отличается от расстановки var_dump() по всему коду?

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица: 2.5 реализовывать классы и объекты с учётом правил объектно-ориентированного программирования

Занятие(-я):

5.2.2.Разработка объектно-ориентированных приложений.

5.2.3.Разработка объектно-ориентированных приложений.

5.2.4.Разработка объектно-ориентированных приложений.

Задание №1 (10 минут)

Написать PHP-скрипт, реализующий систему расчета геометрических фигур с использованием абстракции и полиморфизма. *Параметры задания:*

1. Создать **интерфейс** ShapeInterface с методом getArea().
2. Создать **абстрактный класс** Shape, который реализует этот интерфейс. В абстрактном классе объявить защищенное свойство $color и метод getColor().
3. Создать два конкретных класса: Circle (Круг) и Rectangle (Прямоугольник), которые наследуются от Shape. Реализовать в них метод getArea() по соответствующим математическим формулам.
4. Написать функцию printShapeInfo(ShapeInterface $shape), которая принимает любой объект, реализующий интерфейс, выводит его цвет и площадь. Продемонстрировать работу функции, передав в нее объекты круга и прямоугольника.

Оценка	***Показатели оценки***

5	Код полностью работоспособен. Корректно использованы interface, abstract, extends, implements. Математические формулы верны. Функция printShapeInfo использует type hinting по интерфейсу, что демонстрирует понимание полиморфизма. Соблюдены модификаторы доступа.
4	Логика и математика верны, но допущены нарушения в архитектуре ООП: например, вместо интерфейса использован обычный класс, или абстрактный класс не реализует интерфейс. Либо в функции printShapeInfo отсутствует type hinting (принимает mixed или вообще без типа).
3	Студент не использует абстракцию и интерфейсы (пишет все в одном классе или через обычные функции). Код содержит синтаксические ошибки, не создает объекты или не может продемонстрировать полиморфное поведение функции.

Дидактическая единица: 2.6 выполнять отладку программного кода
Занятие(-я):
6.1.2.Практика диагностики программного кода.
6.1.3.Поиск и исправление синтаксических и логических ошибок.
Задание №1 (10 минут)
Задача: Ниже представлен код, вам нужно найти, исправить и объяснить ошибки, почему код не работал.

Параметры задания:

1. **Синтаксическая ошибка**.
2. **Ошибка времени выполнения**.
3. **Логическая ошибка**.

```php
<?php

$numbers = [10, 25, 30, 45, 60];
$targetSum = 100;
$currentSum = 0;
$i = 0;

while ($i > 0) {
    $currentSum += $numbers[$i]

    if ($currentSum >= $targetSum) {
        break;
    }

    $i++;
}

echo "Итоговая сумма: " . $currentSum . "\n";
echo "Количество обработанных элементов: " . $i . "\n";

?>
```

Оценка	***Показатели оценки***
5	Студент нашел и исправил все 3 ошибки. Код успешно запускается и выдает корректный результат. Студент может четко объяснить природу каждой ошибки (почему возникла, как интерпретатор на нее отреагировал) и предложить способ, как избежать подобных ошибок в будущем.
4	Студент нашел и исправил 2 ошибки из 3 (чаще всего находит синтаксическую и runtime, но не замечает логическую, либо наоборот). Код запускается, но выдает неверный результат из-за оставшейся логической ошибки.
3	Студент нашел только синтаксическую ошибку (интерпретатор сам указал на нее). Не может найти ошибку времени выполнения и логическую ошибку. Пытается «угадать» исправления, меняя код хаотично, без понимания причины сбоя.

3. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ, ИСПОЛЬЗУЕМЫЙ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

№ семестра	Вид промежуточной аттестации
3	Экзамен

Экзамен может быть выставлен автоматически по результатам текущих контролей
Текущий контроль №1
Текущий контроль №2
Текущий контроль №3
Текущий контроль №4
Текущий контроль №5
Текущий контроль №6

Метод и форма контроля: Практическая работа (Информационно-аналитический)
Вид контроля: По выбору выполнить 1 теоретическое задание и 1 практическое задание
Дидактическая единица для контроля:
1.10 Правила объявления и взаимодействие с переменными
Задание №1 (из текущего контроля) (5 минут)
Ответьте на следующие вопросы: Какие существуют базовые правила именования переменных? Что происходит с переменной в оперативной памяти при повторном присваивании ей нового значения?

Оценка	*Показатели оценки*
5	Даны подробные ответы на оба вопроса: перечислены все базовые правила именования переменных (начальный символ, допустимые символы, запрет на ключевые слова, регистрозависимость) и точно описан механизм перезаписи значения в оперативной памяти с упоминанием освобождения памяти сборщиком мусора.
4	Представлены ответы на оба вопроса, но с небольшими неточностями или неполно: перечислены основные правила именования переменных и описан процесс изменения значения переменной, однако без упоминания деталей работы с памятью или сбора мусора.

3	Дан развернутый ответ только на один из вопросов ИЛИ ответы на оба вопроса даны поверхностно без конкретизации правил именования и понимания механизма перезаписи значения в памяти.

Дидактическая единица для контроля:
2.6 выполнять отладку программного кода
Задание №1 (из текущего контроля) (10 минут)
Задача: Ниже представлен код, вам нужно найти, исправить и объяснить ошибки, почему код не работал.

Параметры задания:

1. **Синтаксическая ошибка**.
2. **Ошибка времени выполнения**.
3. **Логическая ошибка**.

```php
<?php

$numbers = [10, 25, 30, 45, 60];
$targetSum = 100;
$currentSum = 0;
$i = 0;

while ($i > 0) {
    $currentSum += $numbers[$i]

    if ($currentSum >= $targetSum) {
        break;
    }

    $i++;
}

echo "Итоговая сумма: " . $currentSum . "\n";
echo "Количество обработанных элементов: " . $i . "\n";

?>
```

Оценка	*Показатели оценки*
5	Студент нашел и исправил все 3 ошибки. Код успешно запускается и выдает корректный результат. Студент может четко объяснить природу каждой ошибки (почему возникла, как интерпретатор на нее отреагировал) и предложить способ, как избежать подобных ошибок в будущем.
4	Студент нашел и исправил 2 ошибки из 3 (чаще всего находит синтаксическую и runtime, но не замечает логическую, либо наоборот). Код запускается, но выдает неверный результат из-за оставшейся логической ошибки.
3	Студент нашел только синтаксическую ошибку (интерпретатор сам указал на нее). Не может найти ошибку времени выполнения и логическую ошибку. Пытается «угадать» исправления, меняя код хаотично, без понимания причины сбоя.

Задание №2 (15 минут)

Найдите и исправьте все ошибки в предложенном фрагменте кода на PHP. Код должен корректно вычислять сумму элементов массива и выводить результат на экран. Укажите тип каждой найденной ошибки.

php

```php
function calculateSum($arr) {
    $sum = 0;
    for ($i = 0; $i <= count($arr); $i++) {
        $sum += $arr[$i];
    }
    return $sum
}

$data = [10, 20, 30];
echo "Сумма: " calculateSum($data);
```

Оценка	*Показатели оценки*
5	Найдены и исправлены все 3 ошибки (синтаксические: отсутствие ; и оператора конкатенации .; логическая/времени выполнения: выход за границы массива из-за <=). Код работает корректно, типы ошибок определены верно.

4	Найдены 2 из 3 ошибок (например, исправлены только синтаксические), либо код работает, но выдает предупреждения (Warning/Notice) из-за неустраненной ошибки выхода за пределы массива.
3	Найдена только 1 ошибка (например, только пропущенная точка с запятой), код не выполняется или содержит критические логические сбои, типы ошибок не определены.

Дидактическая единица для контроля:

1.2 Назначение операторов ветвления

Задание №1 (из текущего контроля) (10 минут)

Ответьте на следующие вопросы:

1. Что такое операторы ветвления и для чего они используются в программировании?
2. Какие виды операторов ветвления существуют в PHP (if, if-else, if-elseif-else, switch)?
3. В чем разница между простой и полной формой условного оператора?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица для контроля:

2.5 реализовывать классы и объекты с учётом правил объектно-ориентированного программирования

Задание №1 (из текущего контроля) (10 минут)

Написать PHP-скрипт, реализующий систему расчета геометрических фигур с использованием абстракции и полиморфизма. *Параметры задания:*

1. Создать **интерфейс** ShapeInterface с методом getArea().
2. Создать **абстрактный класс** Shape, который реализует этот интерфейс. В абстрактном классе объявить защищенное свойство $color и метод getColor().
3. Создать два конкретных класса: Circle (Круг) и Rectangle (Прямоугольник), которые наследуются от Shape. Реализовать в них метод getArea() по соответствующим математическим формулам.
4. Написать функцию printShapeInfo(ShapeInterface $shape), которая принимает любой объект, реализующий интерфейс, выводит его цвет и площадь. Продемонстрировать работу функции, передав в нее объекты круга и прямоугольника.

Оценка	*Показатели оценки*
5	Код полностью работоспособен. Корректно использованы interface, abstract, extends, implements. Математические формулы верны. Функция printShapeInfo использует type hinting по интерфейсу, что демонстрирует понимание полиморфизма. Соблюдены модификаторы доступа.
4	Логика и математика верны, но допущены нарушения в архитектуре ООП: например, вместо интерфейса использован обычный класс, или абстрактный класс не реализует интерфейс. Либо в функции printShapeInfo отсутствует type hinting (принимает mixed или вообще без типа).
3	Студент не использует абстракцию и интерфейсы (пишет все в одном классе или через обычные функции). Код содержит синтаксические ошибки, не создает объекты или не может продемонстрировать полиморфное поведение функции.

Задание №2 (из текущего контроля) (25 минут)
Написать PHP-скрипт, реализующий небольшую объектную модель. *Параметры задания:*

1. Создать базовый класс BankAccount (Банковский счет) с **приватным** свойством $balance (инкапсуляция). Реализовать публичные методы для пополнения, снятия средств и получения текущего баланса.
2. Создать дочерний класс SavingsAccount (Накопительный счет), который **наследуется** от BankAccount. Добавить ему свойство $interestRate (процентная ставка).
3. Реализовать **полиморфизм**: переопределить метод снятия средств в дочернем классе так, чтобы он запрещал снятие, если после операции баланс упадет ниже неснижаемого остатка (например, 100 рублей), и выводить соответствующее сообщение.
4. В основном коде программы создать объекты обоих классов, продемонстрировать их работу и показать разницу в поведении метода снятия средств.

Оценка	*Показатели оценки*

5	Код полностью работоспособен и соответствует всем требованиям. Свойство $balance объявлено как private, доступ к нему осуществляется только через геттеры/сеттеры или методы класса. Корректно использовано ключевое слово extends, вызван родительский конструктор (если он есть). Метод снятия средств успешно переопределен в дочернем классе (полиморфизм), логика проверки остатка работает верно.
4	Код работает и логика верна, но допущены нарушения принципов инкапсуляции (например, свойство $balance объявлено как public или protected вместо private). ИЛИ при переопределении метода в дочернем классе не вызван родительский метод через parent::, из-за чего дублируется код базовой логики.
3	Студент не может реализовать наследование (создает два независимых класса) ИЛИ не использует модификаторы доступа (весь код публичный). Код содержит синтаксические ошибки, не создает объекты или не демонстрирует разницу в работе методов (полиморфизм не показан).

Задание №3 (15 минут)
Создайте класс "Автомобиль" со свойствами $marka и $god. Создайте объект этого класса и присвойте значения свойствам.

Оценка	***Показатели оценки***
5	Класс создан правильно, объект создан, свойствам присвоены значения.
4	Класс и объект созданы, но есть мелкие ошибки.
3	Класс или объект созданы неверно.

Задание №4 (10 минут)
Создайте класс с приватным свойством $balance. Добавьте публичные методы getBalance() и setBalance() для доступа к нему.

Оценка	***Показатели оценки***
5	Инкапсуляция реализована правильно, методы работают корректно.
4	Код работает, но есть недочеты.
3	Методы реализованы неверно.

Задание №5 (10 минут)
Создайте родительский класс "Фигура" и дочерний класс "Круг", который наследует от родителя.

Оценка	*Показатели оценки*
5	Наследование реализовано правильно с использованием ключевого слова extends.
4	Классы созданы, но есть мелкие ошибки.
3	Наследование реализовано неверно.

Дидактическая единица для контроля:
1.11 Классификацию типов данных
Задание №1 (из текущего контроля) (10 минут)
Параметры задания: Список: 42, 3.14, "True", false, [1, 2, 3], null, "100". *Вопросы:*

1. Определите тип каждого значения (скалярный или составной).
2. Что произойдет в PHP при попытке сложить целое число 5 и строку "5" ($result = 5 + "5";)? Чему будет равен $result и какой у него тип?

Оценка	*Показатели оценки*
5	Точно определены типы всех значений, верно разделены скалярные (int, float, string, bool, null) и составные (array) типы. Правильно объяснено поведение PHP при сложении 5 + "5" (сработает неявное приведение типов / type juggling, результат будет 10 типа int).
4	Правильно определены базовые типы, но допущена ошибка в классификации (например, null назван составным типом) ИЛИ студент знает, что строка превратится в число, но ошибается в конечном типе результата (называет его float или string).
3	Допущены грубые ошибки в определении типов (например, строка "True" названа логическим типом bool, а "100" - числовым int) ИЛИ студент считает, что в PHP сложение числа и строки вызовет фатальную ошибку (как в строго типизированных языках), не зная о механизме приведения типов.

Дидактическая единица для контроля:
2.3 работать с одномерными массивами
Задание №1 (из текущего контроля) (10 минут)
Написать PHP-скрипт для анализа набора данных. *Параметры задания:*
Сгенерировать одномерный массив из 15 случайных целых чисел в диапазоне от -50

до 50. Найти и вывести на экран:

1. Максимальный и минимальный элементы массива.
2. Сумму всех положительных элементов.
3. Индекс первого найденного отрицательного элемента.

Оценка	***Показатели оценки***
5	Скрипт успешно выполняется. Массив сгенерирован верно. Все три метрики вычислены абсолютно точно. Код оптимизирован (например, поиск максимума/минимума и суммы выполнен за один проход по массиву).
4	Скрипт работает, но допущена небольшая логическая ошибка в вычислениях (например, посчитана сумма модулей положительных чисел, или найден индекс *последнего* отрицательного элемента вместо первого).
3	Студент не может корректно сгенерировать массив случайных чисел ИЛИ не может реализовать алгоритм поиска (например, просто выводит весь массив через print_r без какой-либо аналитической обработки).

Задание №2 (10 минут)
Создайте индексный массив из 5 элементов (числа). Выведите третий элемент массива.

Оценка	***Показатели оценки***
5	Массив создан правильно, третий элемент выведен корректно (индекс 2).
4	Элемент выведен, но есть мелкие недочеты.
3	Массив создан неверно или выведен не тот элемент.

Задание №3 (15 минут)
Создайте массив из 5 чисел. С помощью цикла foreach найдите и выведите сумму всех элементов.

Оценка	***Показатели оценки***
5	Foreach использован правильно, сумма вычислена верно.
4	Сумма вычислена, но есть недочеты в коде.
3	Foreach использован неверно или сумма не вычислена.

Задание №4 (15 минут)

Создайте ассоциативный массив, где ключи - названия товаров, значения - их цены. Выведите цену одного товара по ключу.

Оценка	***Показатели оценки***
5	Ассоциативный массив создан правильно, цена выведена корректно.
4	Цена выведена, но есть мелкие ошибки.
3	Массив создан неверно или цена не выведена.

Задание №5 (15 минут)
Создайте массив из 5 чисел. Отсортируйте его по возрастанию с помощью функции sort() и выведите результат.

Оценка	***Показатели оценки***
5	Массив отсортирован корректно, результат выведен правильно.
4	Сортировка выполнена, но есть недочеты.
3	Функция sort() использована неверно или массив не отсортирован.

Дидактическая единица для контроля:
2.8 использовать операторы цикла
Задание №1 (из текущего контроля) (10 минут)
Решить две задачи на разные типы циклов. *Параметры задания:*

1. С помощью **вложенного цикла for** вывести на экран прямоугольный треугольник из символов * высотой 6 строк (в первой строке 1 звезда, во второй 2, и тому подобное).
2. С помощью **цикла while** вычислить сумму ряда 1+1/2+1/4+1/8+...1+1/2+1/4+1/8+... до тех пор, пока добавляемое слагаемое не станет меньше 0.01. Вывести итоговую сумму.

Оценка	***Показатели оценки***
5	Обе задачи решены верно. Треугольник выведен с правильным форматированием. Цикл while корректно завершается при достижении заданной точности, математическая логика не нарушена.

4	Треугольник выведен, но имеет нарушения форматирования (например, лишние пробелы в конце строк). ИЛИ цикл while работает, но условие выхода задано неточно (например, < 0.1 вместо < 0.01), из-за чего сумма немного отличается от эталонной.
3	Вложенный цикл выводит не треугольник, а квадрат или прямую линию (ошибка в условии внутреннего цикла). ИЛИ цикл while зацикливается (бесконечный цикл) из-за того, что студент забыл обновлять знаменатель дроби.

Задание №2 (10 минут)
С помощью цикла for выведите все четные числа от 1 до 20.

Оценка	***Показатели оценки***
5	Цикл реализован верно, выведены все четные числа (2, 4, 6, ..., 20).
4	Числа выведены, но есть неоптимальное решение или мелкие ошибки.
3	Цикл некоректно работает или выведены не все четные числа.

Задание №3 (10 минут)
Напишите цикл while, который вычисляет сумму чисел от 1 до 10. Выведите результат.

Оценка	***Показатели оценки***
5	Цикл работает корректно, сумма вычислена верно.
4	Сумма вычислена, но есть недочеты в коде или объяснении.
3	Цикл некорректно работает или сумма вычислена неверно.

Задание №4 (10 минут)
Напишите цикл от 1 до 10, который пропускает число 5 (не выводит его) с помощью оператора continue.

Оценка	***Показатели оценки***
5	Код работает корректно, число 5 пропущено, остальные выведены.
4	Код работает, но есть мелкие недочеты.
3	Continue использован неверно или число 5 не пропущено.

Задание №5 (10 минут)
С помощью вложенных циклов выведите таблицу умножения 3x3 (от 1 до 3).

Оценка	*Показатели оценки*
5	Таблица умножения выведена корректно с помощью вложенных циклов.
4	Таблица выведена, но есть недочеты в форматировании.
3	Таблица не выведена или выведена без вложенных циклов.

Дидактическая единица для контроля:
1.8 Принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, абстракция, наследование
Задание №1 (из текущего контроля) (10 минут)
Ответьте на следующие вопросы:

1. Дайте краткое определение каждому из четырех принципов ООП: инкапсуляция, наследование, полиморфизм, абстракция.
2. Что такое модификаторы доступа (public, private, protected) и как они реализуют принцип инкапсуляции в PHP?
3. Приведите пример из реальной жизни или программирования, где наглядно проявляется принцип наследования и принцип полиморфизма.

Оценка	*Показатели оценки*
5	Студент дает точные и четкие определения всем четырем принципам. Отлично понимает разницу между модификаторами доступа и может объяснить, зачем скрывать данные (защита от некорректного изменения состояния объекта). Приводит уместные и понятные примеры.
4	Студент понимает суть принципов ООП, но допускает неточности в формулировках. Может назвать модификаторы доступа, но затрудняется объяснить разницу между protected и private. Примеры приводит, но они немного размыты.
3	Студент дает поверхностные ответы. Не может четко разграничить принципы, путает наследование и полиморфизм. Не знает модификаторов доступа или считает, что все свойства класса всегда должны быть публичными.

Задание №2 (из текущего контроля) (10 минут)
Ответьте на следующие вопросы:

1. Как в PHP реализуется наследование? Что происходит, если дочерний класс

переопределяет метод родительского класса, и как при этом вызвать исходный метод родителя?
2. В чем разница между абстрактным классом (abstract class) и интерфейсом (interface) в PHP? В каких случаях вы будете использовать интерфейс, а в каких - абстрактный класс?
3. Как полиморфизм связан с типизацией (type hinting) в PHP? Приведите пример, как функция может принимать объекты разных классов.

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Задание №3 (5 минут)

1. Что такое класс в ООП?
2. Что такое объект и как он создается?
3. Что такое свойства и методы класса?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Задание №4 (10 минут)

1. Что такое наследование в ООП?
2. Что такое родительский и дочерний класс?
3. Как создать дочерний класс в PHP?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица для контроля:

1.5 Понятия в программировании: функция, параметр функции

Задание №1 (из текущего контроля) (5 минут)

Дайте определения понятиям «функция», «параметр» и «аргумент».В чем разница

между параметром и аргументом?

Оценка	*Показатели оценки*
5	Студент дает **точные определения** всех трех понятий без существенных ошибок, объяснив разницу между параметром и аргументом.
4	Студент дает **в целом верные определения**, но с небольшими неточностями в формулировках.
3	П**риблизительные определения**: студент путает функцию с процедурой/методом, не может четко сформулировать, что такое параметр.

Дидактическая единица для контроля:

2.7 использовать операторы ветвления

Задание №1 (из текущего контроля) (35 минут)

Создайте программу, которая определяет, является ли число четным или нечетным используя тернарный оператор. Выведите соответствующее сообщение.

Оценка	*Показатели оценки*
5	Программа работает корректно, тернарный оператор использован правильно, вывод сообщения соответствует результату проверки.
4	Программа работает, тернарный оператор использован, но допущены незначительные недочеты в коде или выводе сообщений.
3	Тернарный оператор использован неверно или заменен на условный оператор if, программа работает некорректно или содержит синтаксические ошибки.

Задание №2 (10 минут)

Напишите программу, которая по номеру дня недели (1-7) выводит его название. Используйте оператор switch.

Оценка	*Показатели оценки*
5	Программа корректно выводит названия всех 7 дней недели.
4	Программа работает, но есть недочеты (нет break или default).
3	Программа некорректно работает или выводит неверные названия.

Дидактическая единица для контроля:

1.1 Порядок выполнения инструкций программного кода

Задание №1 (из текущего контроля) (5 минут)

Ответьте на следующие вопросы:

1. Что такое «поток выполнения» (execution flow) программы?
2. В каком порядке выполняются инструкции в линейном алгоритме?

3. Может ли порядок выполнения строк кода измениться без использования циклов и условий?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица для контроля:

2.1 анализировать условие задачи для выявления входных и выходных данных

Задание №1 (из текущего контроля) (15 минут)

Задание: Заполнить таблицу анализа задачи. Для каждой задачи выписать: 1) Что дано (Вход), 2) Что нужно получить (Выход), 3) Ограничения/допущения

Параметры задач:

1. *Математическая:* «Дан массив оценок студента. Нужно вычислить его средний балл и узнать, есть ли у него двойки».
2. *Бытовая/Алгоритмическая:* «Система умного дома должна включать отопление, если температура на улице падает ниже заданного порога, и выключать, если превышает».
3. *Экономическая:* «Рассчитать итоговую стоимость корзины товаров в интернет-магазине с учетом скидки, если сумма покупок превышает N рублей».

Оценка	***Показатели оценки***
5	Таблица анализа задачи заполнена полностью и верно для всех трех задач: точно выделены все входные и выходные данные, а также корректно сформулированы ограничения и допущения без логических ошибок.
4	Таблица заполнена для всех трех задач, но допущены незначительные неточности: упущены некоторые входные данные или ограничения, либо некорректно сформулированы выходные данные для одной из задач.

3	Таблица заполнена только для одной-двух задач ИЛИ анализ выполнен поверхностно: полностью отсутствуют ограничения и допущения, либо грубо перепутаны входные и выходные данные.

Дидактическая единица для контроля:

1.6 Области видимости данных

Задание №1 (из текущего контроля) (10 минут)

Фрагмент PHP-кода, в котором используется глобальная переменная и локальная переменная с таким же именем внутри функции (эффект «затенения»). *Параметры задания:*

```php
<?php
$x = 10;
function change_x() {
    $x = 5;
    echo "Внутри функции: " . $x . "\n";
}
change_x();
echo "Снаружи функции: " . $x . "\n";
?>
```

Что выведет программа? Как изменить код внутри функции так, чтобы она реально изменила значение глобальной переменной $x?

Оценка	***Показатели оценки***
5	Верно предсказан вывод программы (5, затем 10). Точно объяснена разница между локальной и глобальной областью видимости в PHP. Самостоятельно и без ошибок продемонстрирован способ изменения глобальной переменной из функции.
4	Верно предсказан вывод программы и понята суть локальных/глобальных переменных, но допущена небольшая синтаксическая ошибка при демонстрации способа изменения глобальной переменной.
3	Неверно предсказан вывод программы (студент считает, что локальное присваивание изменит глобальную переменную, и ответит «5, затем 5») ИЛИ не может объяснить разницу между областями видимости без наводящих вопросов преподавателя.

Дидактическая единица для контроля:

2.4 реализовывать программные функции

Задание №1 (из текущего контроля) (10 минут)

Напишите PHP-функцию calculate_discount($price, $percent), которая принимает стоимость товара и процент скидки, а возвращает итоговую стоимость товара со скидкой. Как вызвать эту функцию и сохранить результат в переменную? Что произойдет, если передать в нее строку "abc" вместо числа (в контексте PHP 8 и выше)?

Оценка	***Показатели оценки***
5	Функция написана синтаксически верно, корректно использует параметры (с $), содержит оператор return для возврата результата. Студент уверенно отвечает на вопросы о вызове функции и знает, что передача нечисловой строки "abc" в математической операции в PHP 8+ вызовет ошибку TypeError (или приведет к 0 в старых версиях, если студент это уточнит).
4	Функция работает и возвращает верный результат, но допущены мелкие стилистические ошибки (плохие имена параметров) ИЛИ студент затрудняется ответить на вопрос о поведении PHP при передаче некорректных строковых данных в математическую операцию.
3	Функция не возвращает значение (использует echo вместо return), ИЛИ содержит синтаксические ошибки (забыты знаки $ перед переменными, пропущены точки с запятой), из-за которых код не может быть выполнен.

Дидактическая единица для контроля:

1.3 Назначение циклических операторов

Задание №1 (из текущего контроля) (5 минут)

Ответьте на следующие вопросы:

1. В чем заключается основное назначение циклических операторов? В каких случаях их использование обязательно, а в каких - просто сокращает объем кода?
2. В чем концептуальная разница между циклами for, while и do-while? Приведите пример ситуации, когда лучше использовать do-while, а не while.
3. Что такое «бесконечный цикл», как его можно намеренно создать в PHP и как из него выйти до завершения программы?

Оценка	***Показатели оценки***

5	Студент четко объясняет, что циклы нужны для многократного повторения блока кода без его дублирования. Точно описывает разницу между циклами (например, do-while гарантирует минимум одно выполнение тела цикла). Знает, как создать бесконечный цикл (while(true)) и как его прервать (break).
4	Студент понимает назначение циклов и может сравнить for и while, но затрудняется с примером для do-while или не может четко объяснить механизм прерывания бесконечного цикла без наводящих вопросов.
3	Студент дает размытые ответы (например, «циклы нужны, чтобы программа работала дольше»), путает синтаксис и назначение разных типов циклов, не понимает концепцию бесконечного цикла.

Задание №2 (5 минут)

1. Как пройти по всем элементам массива с помощью цикла?
2. Что такое функция foreach?
3. Когда использовать foreach вместо for?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица для контроля:

2.9 работать с двумерными массивами

Задание №1 (из текущего контроля) (10 минут)

Написать скрипт для обработки матрицы. *Параметры задания:* Создать двумерный массив (матрицу) размером 4x4, заполненный случайными числами от 1 до 10.

1. Вывести матрицу на экран в виде аккуратной HTML-таблицы (<table>) или выровненного текстового блока.
2. Вычислить и вывести сумму элементов, находящихся на главной диагонали матрицы.

Оценка	***Показатели оценки***

5	Матрица корректно сгенерирована и красиво выведена в табличном виде (использованы вложенные циклы и теги <tr>, <td>). Сумма главной диагонали вычислена верно (использовано условие $i == $j или прямой проход по индексам).
4	Матрица выведена, но форматирование таблицы «плавает» (не заданы границы, нет отступов, или использован print_r вместо HTML-таблицы, хотя задание требовало визуального вывода). ИЛИ сумма посчитана верно, но для побочной диагонали вместо главной (ошибка в индексах).
3	Студент не понимает структуру двумерного массива (пытается обращаться к нему как к одномерному), не может вывести его в табличном виде и не может реализовать алгоритм обхода диагонали.

Задание №2 (15 минут)
Создайте двумерный массив 2x3 (2 строки, 3 столбца). Выведите элемент из второй строки третьего столбца.

Оценка	***Показатели оценки***
5	Двумерный массив создан правильно, элемент выведен корректно (индексы [1][2]).
4	Элемент выведен, но есть мелкие ошибки.
3	Массив создан неверно или выведен не тот элемент.

Задание №3 (15 минут)
Создайте двумерный массив 3x3. Напишите код для вычисления суммы элементов первой строки.

Оценка	***Показатели оценки***
5	Сумма элементов первой строки вычислена верно с помощью циклов.
4	Сумма вычислена, но есть недочеты в коде.
3	Сумма вычислена неверно или без использования циклов.

Дидактическая единица для контроля:
1.9 Правила отладки программ
Задание №1 (из текущего контроля) (10 минут)
Ответьте на следующие вопросы:

1. Назовите три основных типа ошибок в программировании (синтаксические, ошибки времени выполнения, логические). Приведите пример каждого типа.
2. Какие встроенные средства PHP вы используете для быстрой отладки (вывода данных на экран)? В чем разница между echo, print_r() и var_dump()?
3. Чем использование отладчика (например, Xdebug с точками останова / breakpoints) принципиально отличается от расстановки var_dump() по всему коду?

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Задание №2 (5 минут)

1. Какие виды ошибок в коде вы знаете?
2. Что такое синтаксические, логические ошибки и ошибки времени выполнения?
3. Какие методы отладки программ вы знаете?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица для контроля:

1.7 Понятие объекта и класса

Задание №1 (из текущего контроля) (5 минут)

Ответьте на следующие вопросы:

1. Что такое класс и что такое объект? В чем разница между ними? Приведите простую аналогию из реальной жизни.
2. Чем свойство (атрибут) класса отличается от метода? Что такое $this и зачем оно нужно внутри методов?
3. Что такое инстанцирование (создание экземпляра)? Какой синтаксис используется в PHP для создания нового объекта на основе класса?

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.

3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица для контроля:

2.2 представлять алгоритм в виде блок-схемы

Задание №1 (из текущего контроля) (15 минут)

Задание: Построить блок-схему алгоритма на бумаге или в графическом редакторе.

Параметры задания: Алгоритм «Вычисление значения кусочно-заданной функции (если x < 0, то y = x^2, иначе y = 2x)». Блок-схема должна содержать не менее 4-5 блоков, включая блок условия (ромб) с двумя ветками (Да/Нет).

Оценка	***Показатели оценки***
5	Блок-схема построена полностью корректно, содержит не менее 4-5 стандартных блоков (включая ромб с двумя ветками «Да/Нет») и точно отражает алгоритм вычисления заданной кусочно-заданной функции.
4	Блок-схема отражает верную логику алгоритма и содержит блок условия с двумя ветками, но допущены незначительные недочеты: отсутствуют блоки начала/конца, использованы нестандартные символы или есть мелкие ошибки в направлениях связей.
3	Блок-схема содержит грубые логические ошибки (неверные условия или формулы), отсутствует одна из веток условия, либо не соблюдены минимальные требования к количеству и типу используемых графических блоков.

Дидактическая единица для контроля:

1.4 Перечень операций с массивами

Задание №1 (из текущего контроля) (10 минут)

Ответьте на следующие вопросы:

1. Что такое массив в PHP? Чем индексированный (числовой) массив отличается от ассоциативного?
2. Назовите и опишите назначение 4-5 стандартных встроенных функций PHP для работы с массивами (например, для добавления, удаления, сортировки, поиска или подсчета элементов).
3. Какими способами можно перебрать все элементы массива в PHP? В чем преимущество конструкции foreach перед обычным циклом for?

Оценка	***Показатели оценки***

5	Студент дает точные определения, четко разделяет индексированные и ассоциативные массивы. Свободно называет встроенные функции (array_push, sort, in_array, count и тому подобное) и объясняет, что foreach удобнее и безопаснее для перебора, так как не требует ручного управления счетчиком и индексами.
4	Студент понимает разницу между типами массивов, называет основные функции, но может забыть 1-2 из них или неточно сформулировать их синтаксис. Понимает преимущество foreach, но объясняет его общими словами.
3	Студент путает массивы с обычными переменными, не знает разницы между индексированным и ассоциативным массивом. Не может назвать ни одной встроенной функции, предлагая писать все операции (поиск, сортировку) вручную через циклы.

Задание №2 (5 минут)

1. Что такое ассоциативный массив?
2. Как обратиться к элементу ассоциативного массива?
3. Где применяются ассоциативные массивы?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Задание №3 (5 минут)

1. Какие функции PHP для работы с массивами вы знаете?
2. Что делают функции count(), sort(), array_push()?
3. Как добавить элемент в массив?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Задание №4 (5 минут)

1. Что такое многомерный массив?
2. Как обратиться к элементу двумерного массива?
3. Где применяются двумерные массивы?

Оценка	*Показатели оценки*
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Задание №5 (5 минут)

1. Как пройти по всем элементам двумерного массива?
2. Какие циклы используются для обработки матриц?
3. Как вычислить сумму элементов строки матрицы?

Оценка	*Показатели оценки*
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Дидактическая единица для контроля:

1.12 Понятие линейных конструкций

Задание №1 (10 минут)

Составьте схему/таблицу, отражающую основные виды линейных конструкций с примерами из реальной жизни. Укажите их назначение и особенности применения.

Оценка	*Показатели оценки*
5	Работа выполнена без ошибок, полно раскрыты все виды линейных конструкций и приведены точные примеры их применения.
4	Материал раскрыт в основном верно, но присутствуют незначительные неточности в примерах или описании особенностей конструкций.
3	Тема раскрыта поверхностно, рассмотрен минимум видов конструкций, а примеры отсутствуют или описаны с существенными ошибками.

Задание №2 (из текущего контроля) (15 минут)

Параметры задания: «Поменять местами значения двух переменных $a и $b,

используя третью вспомогательную переменную $c». *Требование:* Написать последовательный PHP-код и нарисовать соответствующую блок-схему.

Оценка	***Показатели оценки***
5	Написан корректный линейный PHP-код без логических ошибок (с использованием $, ;), построена полная блок-схема со всеми стандартными блоками (начало/конец, ввод/вывод, процесс) и строгой последовательностью связей.
4	Код написан верно, но в блок-схеме допущены незначительные недочеты (например, отсутствуют блоки начала/конца, использован нестандартный шрифт/форма блока или не подписаны линии потоков).
3	В коде допущена логическая ошибка (например, потеря значения при присваивании $a = $b; до использования $a в $c), ИЛИ блок-схема не соответствует коду, либо отсутствуют обязательные графические элементы.

Дидактическая единица для контроля:

1.13 Шаблоны проектирования в объектно-ориентированном программировании

Задание №1 (10 минут)

Составьте таблицу, классифицирующую шаблоны на три группы (порождающие, структурные, поведенческие). Для каждой группы выберите по одному шаблону, укажите его назначение, приведите пример задачи для его применения и кратко поясните, какие преимущества он дает в коде.

Оценка	***Показатели оценки***
5	Полно и точно описаны все три группы шаблонов с корректными примерами, четко обоснована их практическая польза, работа логично структурирована и оформлена без ошибок.
4	Основные группы и шаблоны раскрыты верно, но присутствуют незначительные неточности в примерах или пояснениях назначения.
3	Тема рассмотрена поверхностно, классификация неполная, примеры отсутствуют или содержат существенные ошибки в понимании применения шаблонов.

Задание №2 (из текущего контроля) (10 минут)

Ответьте на следующие вопросы:

1. Что такое шаблон проектирования (Design Pattern)? Зачем они нужны при

разработке ПО?
2. Назовите и опишите суть 1-2 базовых порождающих или структурных шаблонов.
3. Какую проблему решает использование шаблонов проектирования по сравнению с написанием кода «в лоб»? Приведите пример одного поведенческого шаблона (например, «Стратегия» или «Наблюдатель») и опишите ситуацию в разработке, где его применение позволяет избавиться от громоздких конструкций if-else или switch

Оценка	***Показатели оценки***
5	Представлены развернутые ответы на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.

Задание №3 (10 минут)

1. Что такое инкапсуляция?
2. Какие модификаторы доступа вы знаете (public, private, protected)?
3. Зачем скрывать данные класса?

Оценка	***Показатели оценки***
5	Ответы представлены подробно на все вопросы.
4	Представлены развернутые ответы на 2 вопроса.
3	Представлен развернутый ответ на 1 вопрос.