Министерство образования Республики Беларусь

Учреждение образования

Белорусский Государственный университет информатики и радиоэлектроники

Инженерно-экономический факультет

Кафедра экономической информатики

Отчет

по лабораторной работе № 3

«Создание и использование объектов и встроенных классов. Наследование, инкапсуляция и интерфейсы»

|  |  |
| --- | --- |
| Выполнил: | Летко А.С. |
|  | студент группы 377901 |
| Проверил: | Сторожев Д.А. |
|  | канд.техн.наук, доцент |

Минск 2025

**Цель работы**

Изучить основные принципы объектно-ориентированного программирования в PHP и применить их на практике: создание собственных и использование встроенных классов, реализовать наследование, инкапсуляцию и интерфейсы.

**Задачи**

* определить и создать собственный класс с конструктором и методами;
* ознакомиться с примерами встроенных классов PHP (например, DateTime, SplFileObject) и продемонстрировать их применение;
* реализовать наследование между двумя пользовательскими классами;
* показать механизм инкапсуляции: работа с приватными и защищёнными свойствами.

**Теоретическая часть**

Класс – это шаблон, описывающий свойства (переменные) и методы (функции), а объект – экземпляр этого шаблона.

Инкапсуляция скрывает внутренние данные объекта и предоставляет доступ через методы-аксессоры. Это защищает состояние от некорректных изменений.

Наследование позволяет создавать подклассы на базе существующих, чтобы расширять или переопределять функциональность родителя.

В PHP используется ключевое слово extends для указания, что класс наследует другой. Дочерний класс может:

* использовать методы и свойства родителя;
* переопределять методы родителя;
* добавлять собственные методы и свойства.

Видимость:

public – доступно везде.

protected – доступно в родителе и потомке.

private – доступно только в родителе, не наследуется.

**Использование parent::**

Иногда нужно вызвать метод родителя внутри переопределённого метода:

**Ограничения**

* Нельзя наследовать private методы и свойства.
* PHP не поддерживает множественное наследование (один класс может наследовать только один родитель).

– Интерфейс определяет контракт – набор методов без реализации. Классы-реализаторы обязаны предоставить свою логику. Это даёт возможность работать с разными объектами через единый тип.

*– Интерфейс определяет контракт: какие методы должен реализовать класс. Он не содержит реализаций.*

– Трейты позволяют «внедрять» методы в классы, обходя ограничения одиночного наследования.

**Индивидуальное задание.**

Листинг кода.

Задание 1.

Customer.php:

<?php  
  
namespace App\Models;  
  
class Customer  
{  
 private string $name;  
 private string $email;  
  
 public function \_\_construct(string $name, string $email)  
 {  
 $this->name = $name;  
 $this->email = $email;  
 }  
  
 public function getName(): string  
 {  
 return $this->name;  
 }  
  
 public function getEmail(): string  
 {  
 return $this->email;  
 }  
  
 public function getInfo(): string  
 {  
 return "Клиент: {$this->name} ({$this->email})";  
 }  
}

index.php:

<?php  
declare(strict\_types=1);  
  
require\_once *\_\_DIR\_\_* . '/../src/Interfaces/Discountable.php';  
require\_once *\_\_DIR\_\_* . '/../src/Traits/Loggable.php';  
require\_once *\_\_DIR\_\_* . '/../src/Models/Product.php';  
require\_once *\_\_DIR\_\_* . '/../src/Models/Book.php';  
require\_once *\_\_DIR\_\_* . '/../src/Models/Electronic.php';  
require\_once *\_\_DIR\_\_* . '/../src/Models/Cart.php';  
require\_once *\_\_DIR\_\_* . '/../src/Models/Customer.php';  
  
session\_start();  
use App\Models\Book;  
use App\Models\Electronic;  
use App\Models\Cart;  
use App\Models\Customer;  
  
if (!isset($\_SESSION['products'])) {  
 $\_SESSION['products'] = [  
 new Book("PHP для начинающих", 80, "Иван Иванов"),  
 new Book("Паттерны проектирования", 60, "Гамма и др."),  
 new Electronic("Смартфон Honor 200", 1200, "HonorBrand", 24),  
 new Electronic("Наушники ProMax", 120, "SoundMax", 12),  
 ];  
}  
if (!isset($\_SESSION['cart'])) {  
 $\_SESSION['cart'] = new Cart();  
}  
if (!isset($\_SESSION['customer'])) {  
 $\_SESSION['customer'] = new Customer("Алексей Петров", "alex@example.com");  
}  
  
$action = $\_POST['action'] ?? null;  
$pid = $\_POST['product\_id'] ?? null;  
  
if ($action && $pid) {  
 $product = null;  
 foreach ($\_SESSION['products'] as $p) {  
 if ($p->getId() === $pid) { $product = $p; break; }  
 }  
 if ($product) {  
 switch ($action) {  
 case 'add\_to\_cart': $\_SESSION['cart']->add($product); break;  
 case 'remove\_from\_cart': $\_SESSION['cart']->remove($product->getId()); break;  
 case 'discount\_10': $product->applyDiscount(10); break;  
 case 'discount\_20': $product->applyDiscount(20); break;  
 }  
 }  
}  
if (($action ?? null) === 'checkout') {  
 $total = $\_SESSION['cart']->total();  
 $\_SESSION['message'] = "Заказ оформлен. Итог: " . number\_format($total, 2, '.', ' ') . " руб.";  
 $\_SESSION['cart']->clear();  
}  
if (($action ?? null) === 'clear\_cart') {  
 $\_SESSION['cart']->clear();  
 $\_SESSION['message'] = "Корзина очищена.";  
}  
  
if (($action ?? null) === 'clear\_session') {  
 session\_destroy();  
 header("Location: index.php");  
 exit;  
}  
  
?>  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <title>Интернет‑магазин (ООП)</title>  
 <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
<header class="wrap">  
 <h1>Интернет‑магазин — итоговый мини‑проект (ООП: наследование, интерфейсы, трейты)</h1>  
 <div class="muted">Демонстрация: Product → Book/Electronic, Discountable, Loggable, Cart, Customer</div>  
 <div class="customer-info">  
 <?= htmlspecialchars($\_SESSION['customer']->getInfo()) ?>  
 </div>  
</header>  
  
<main class="wrap grid">  
 <section class="card">  
 <h2>Каталог товаров</h2>  
 <div class="list">  
 <?php foreach ($\_SESSION['products'] as $product): ?>  
 <div class="item">  
 <div>  
 <div class="title"><?= htmlspecialchars($product->getInfo()) ?></div>  
 <div class="sub">ID: <span class="accent"><?= htmlspecialchars($product->getId()) ?></span></div>  
 </div>  
 <form method="post" class="btns">  
 <input type="hidden" name="product\_id" value="<?= htmlspecialchars($product->getId()) ?>">  
 <button name="action" value="add\_to\_cart">В корзину</button>  
 <button name="action" value="discount\_10">Скидка 10%</button>  
 <button name="action" value="discount\_20">Скидка 20%</button>  
 <button type="button" class="details-btn" data-product-id="<?= htmlspecialchars($product->getId()) ?>">Подробнее</button>  
 </form>  
 </div>  
 <?php endforeach; ?>  
 </div>  
 </section>  
  
 <aside class="card">  
 <h2>Корзина</h2>  
 <div class="list">  
 <?php foreach ($\_SESSION['cart']->getItems() as $p): ?>  
 <div class="item">  
 <div>  
 <div class="title"><?= htmlspecialchars($p->getName()) ?></div>  
 <div class="sub">Цена: <?= number\_format($p->getPrice(), 2, '.', ' ') ?> руб.</div>  
 </div>  
 <form method="post" class="btns">  
 <input type="hidden" name="product\_id" value="<?= htmlspecialchars($p->getId()) ?>">  
 <button class="remove" name="action" value="remove\_from\_cart">Убрать</button>  
 </form>  
 </div>  
 <?php endforeach; ?>  
 </div>  
  
 <div class="total">  
 <div class="muted">Итого: <strong><?= number\_format($\_SESSION['cart']->total(), 2, '.', ' ') ?></strong> руб.</div>  
 <form method="post">  
 <button class="checkout" name="action" value="checkout">Оформить заказ</button>  
 </form>  
 </div>  
  
 <div class="btns" style="margin:12px;">  
 <form method="post">  
 <button class="btn-clear-cart" name="action" value="clear\_cart">Очистить корзину</button>  
 </form>  
 <form method="post">  
 <button class="btn-clear-session" name="action" value="clear\_session">Сбросить сессию</button>  
 </form>  
 </div>  
  
 <?php if (!empty($\_SESSION['message'])): ?>  
 <div class="msg"><?= htmlspecialchars($\_SESSION['message']); unset($\_SESSION['message']); ?></div>  
 <?php endif; ?>  
 </aside>  
</main>  
  
<div id="productModal" class="modal">  
 <div class="modal-content">  
 <span class="close">&times;</span>  
 <div id="modalBody"></div>  
 </div>  
</div>  
  
<script>  
 document.addEventListener('DOMContentLoaded', function() {  
 const modal = document.getElementById('productModal');  
 const modalBody = document.getElementById('modalBody');  
 const closeBtn = document.querySelector('.close');  
  
 document.querySelectorAll('.details-btn').forEach(button => {  
 button.addEventListener('click', function() {  
 const productId = this.getA`ttribute('data-product-id');  
  
 const products = <?= json\_encode(array\_map(function($p) {  
 return [  
 'id' => $p->getId(),  
 'name' => $p->getName(),  
 'price' => $p->getPrice(),  
 'info' => $p->getInfo(),  
 'type' => get\_class($p)  
 ];  
 }, $\_SESSION['products'])) ?>;  
  
 const product = products.find(p => p.id === productId);  
  
 if (product) {  
 let details = `<h3>${product.name}</h3>`;  
 details += `<p><strong>Цена:</strong> ${product.price.toFixed(2)} руб.</p>`;  
 details += `<p><strong>Тип:</strong> ${product.type === 'App\\Models\\Book' ? 'Книга' : 'Электроника'}</p>`;  
 details += `<p><strong>Описание:</strong> ${product.info}</p>`;  
  
 if (product.type === 'App\\Models\\Book') {  
 details += `<p><strong>Категория:</strong> Учебная литература</p>`;  
 } else {  
 details += `<p><strong>Категория:</strong> Электронные устройства</p>`;  
 }  
  
 modalBody.innerHTML = details;  
 modal.style.display = 'block';  
 }  
 });  
 });  
   
 closeBtn.addEventListener('click', function() {  
 modal.style.display = 'none';  
 });  
  
 window.addEventListener('click', function(event) {  
 if (event.target === modal) {  
 modal.style.display = 'none';  
 }  
 });  
 });  
</script>  
</body>  
</html>

Задание 2.

art\_gallery.php:

<?php  
abstract class Artwork {  
 protected string $title;  
 protected string $author;  
 protected int $year;  
 protected float $baseValue;  
  
 public function \_\_construct(string $title, string $author, int $year, float $baseValue) {  
 $this->title = $title;  
 $this->author = $author;  
 $this->year = $year;  
 $this->baseValue = $baseValue;  
 }  
  
 abstract public function display(): string;  
 abstract public function appraiseValue(): float;  
  
 public function getAuthor(): string {  
 return $this->author;  
 }  
  
 public function getYear(): int {  
 return $this->year;  
 }  
  
 public function info(): string {  
 return "{$this->title} ({$this->author}, {$this->year}) - Оценка: " .  
 number\_format($this->appraiseValue(), 2, '.', ' ') . " руб.";  
 }  
  
 public function getTitle(): string {  
 return $this->title;  
 }  
  
 public function getBaseValue(): float {  
 return $this->baseValue;  
 }  
}  
  
class Painting extends Artwork {  
 private string $style;  
 private string $medium;  
 private float $conditionMultiplier;  
  
 public function \_\_construct(string $title, string $author, int $year, float $baseValue,  
 string $style, string $medium, float $conditionMultiplier = 1.0) {  
 parent::\_\_construct($title, $author, $year, $baseValue);  
 $this->style = $style;  
 $this->medium = $medium;  
 $this->conditionMultiplier = max(0.1, min(2.0, $conditionMultiplier));  
 }  
  
 public function display(): string {  
 return "Картина: {$this->title}\n" .  
 "Стиль: {$this->style}, Техника: {$this->medium}\n" .  
 "Состояние: " . ($this->conditionMultiplier > 1 ? 'Отличное' :  
 ($this->conditionMultiplier < 1 ? 'Плохое' : 'Хорошее'));  
 }  
  
 public function appraiseValue(): float {  
 $ageMultiplier = max(1.0, (date('Y') - $this->year) \* 0.05);  
 $styleMultiplier = $this->getStyleMultiplier();  
  
 return $this->baseValue \* $ageMultiplier \* $styleMultiplier \* $this->conditionMultiplier;  
 }  
  
 private function getStyleMultiplier(): float {  
 return match($this->style) {  
 'импрессионизм' => 2.5,  
 'реализм' => 1.8,  
 'абстракционизм' => 1.5,  
 'сюрреализм' => 2.0,  
 default => 1.2  
 };  
 }  
  
 public function getStyle(): string {  
 return $this->style;  
 }  
  
 public function getMedium(): string {  
 return $this->medium;  
 }  
}  
  
class Sculpture extends Artwork {  
 private string $material;  
 private float $weight;  
 private string $condition;  
  
 public function \_\_construct(string $title, string $author, int $year, float $baseValue,  
 string $material, float $weight, string $condition = 'хорошее') {  
 parent::\_\_construct($title, $author, $year, $baseValue);  
 $this->material = $material;  
 $this->weight = $weight;  
 $this->condition = $condition;  
 }  
  
 public function display(): string {  
 return "Скульптура: {$this->title}\n" .  
 "Материал: {$this->material}, Вес: {$this->weight} кг\n" .  
 "Состояние: {$this->condition}";  
 }  
  
 public function appraiseValue(): float {  
 $materialMultiplier = $this->getMaterialMultiplier();  
 $weightMultiplier = min(3.0, $this->weight \* 0.1);  
 $conditionMultiplier = $this->getConditionMultiplier();  
 $ageMultiplier = max(1.0, (date('Y') - $this->year) \* 0.03);  
  
 return $this->baseValue \* $materialMultiplier \* $weightMultiplier \* $conditionMultiplier \* $ageMultiplier;  
 }  
  
 private function getMaterialMultiplier(): float {  
 return match($this->material) {  
 'бронза' => 2.5,  
 'мрамор' => 3.0,  
 'дерево' => 1.5,  
 'глина' => 1.2,  
 default => 1.0  
 };  
 }  
  
 private function getConditionMultiplier(): float {  
 return match($this->condition) {  
 'отличное' => 1.5,  
 'хорошее' => 1.0,  
 'удовлетворительное' => 0.7,  
 'плохое' => 0.3,  
 default => 0.5  
 };  
 }  
  
 public function getMaterial(): string {  
 return $this->material;  
 }  
  
 public function getWeight(): float {  
 return $this->weight;  
 }  
}  
  
class DigitalArt extends Artwork {  
 private string $format;  
 private int $resolution;  
 private bool $isLimitedEdition;  
 private int $editionNumber;  
  
 public function \_\_construct(string $title, string $author, int $year, float $baseValue,  
 string $format, int $resolution, bool $isLimitedEdition = false,  
 int $editionNumber = 1) {  
 parent::\_\_construct($title, $author, $year, $baseValue);  
 $this->format = $format;  
 $this->resolution = $resolution;  
 $this->isLimitedEdition = $isLimitedEdition;  
 $this->editionNumber = $editionNumber;  
 }  
  
 public function display(): string {  
 $editionInfo = $this->isLimitedEdition ?  
 "Ограниченная серия №{$this->editionNumber}" : "Без ограничений";  
  
 return "Цифровое искусство: {$this->title}\n" .  
 "Формат: {$this->format}, Разрешение: {$this->resolution}p\n" .  
 "Тип: {$editionInfo}";  
 }  
  
 public function appraiseValue(): float {  
 $resolutionMultiplier = min(2.0, $this->resolution / 1000);  
 $editionMultiplier = $this->isLimitedEdition ? 2.0 : 0.5;  
 $formatMultiplier = $this->getFormatMultiplier();  
 $noveltyMultiplier = max(0.5, 2.0 - (date('Y') - $this->year) \* 0.1);  
  
 return $this->baseValue \* $resolutionMultiplier \* $editionMultiplier \* $formatMultiplier \* $noveltyMultiplier;  
 }  
  
 private function getFormatMultiplier(): float {  
 return match($this->format) {  
 '4K' => 1.8,  
 '2K' => 1.5,  
 'Full HD' => 1.2,  
 'HD' => 1.0,  
 default => 0.8  
 };  
 }  
  
 public function getFormat(): string {  
 return $this->format;  
 }  
  
 public function getResolution(): int {  
 return $this->resolution;  
 }  
  
 public function isLimitedEdition(): bool {  
 return $this->isLimitedEdition;  
 }  
}  
  
$artCollection = [  
 new Painting("Звездная ночь", "Винсент Ван Гог", 1889, 50000000, "постимпрессионизм", "масло", 1.2),  
 new Painting("Крик", "Эдвард Мунк", 1893, 45000000, "экспрессионизм", "масло, темпера", 0.9),  
 new Sculpture("Давид", "Микеланджело", 1504, 80000000, "мрамор", 5600, "отличное"),  
 new Sculpture("Мыслитель", "Огюст Роден", 1902, 35000000, "бронза", 180, "хорошее"),  
 new DigitalArt("Каждый день: первые 5000 дней", "Бипл", 2021, 1000000, "4K", 4096, true, 1),  
 new DigitalArt("Цифровая абстракция", "Алексей Петров", 2023, 50000, "2K", 2048, false)  
];  
?>  
  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <title>Галерея искусств</title>  
 <style>  
 body {  
 font-family: Arial, sans-serif;  
 margin: 20px;  
 background: #f5f5f5;  
 }  
 .art-gallery {  
 max-width: 1200px;  
 margin: 0 auto;  
 }  
 .artwork {  
 background: white;  
 margin: 15px 0;  
 padding: 20px;  
 border-radius: 10px;  
 box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
 }  
 .painting { border-left: 5px solid #e74c3c; }  
 .sculpture { border-left: 5px solid #3498db; }  
 .digital { border-left: 5px solid #2ecc71; }  
 h2 { color: #2c3e50; margin-top: 0; }  
 .details {  
 background: #ecf0f1;  
 padding: 10px;  
 margin: 10px 0;  
 border-radius: 5px;  
 }  
 .value {  
 font-weight: bold;  
 color: #e74c3c;  
 font-size: 1.2em;  
 }  
 .type {  
 display: inline-block;  
 padding: 3px 8px;  
 border-radius: 3px;  
 color: white;  
 font-size: 0.8em;  
 }  
 .type-painting { background: #e74c3c; }  
 .type-sculpture { background: #3498db; }  
 .type-digital { background: #2ecc71; }  
 </style>  
</head>  
<body>  
<div class="art-gallery">  
 <h1>🎨 Галерея искусств</h1>  
 <p>Демонстрация наследования: Artwork → Painting, Sculpture, DigitalArt</p>  
  
 <?php foreach ($artCollection as $art): ?>  
 <div class="artwork <?= get\_class($art) === 'Painting' ? 'painting' :  
 (get\_class($art) === 'Sculpture' ? 'sculpture' : 'digital') ?>">  
  
 <?php  
 $typeClass = match(get\_class($art)) {  
 'Painting' => 'type-painting',  
 'Sculpture' => 'type-sculpture',  
 'DigitalArt' => 'type-digital'  
 };  
 $typeName = match(get\_class($art)) {  
 'Painting' => 'Картина',  
 'Sculpture' => 'Скульптура',  
 'DigitalArt' => 'Цифровое искусство'  
 };  
 ?>  
  
 <span class="type <?= $typeClass ?>"><?= $typeName ?></span>  
 <h2><?= htmlspecialchars($art->getTitle()) ?></h2>  
  
 <div class="details">  
 <strong>Информация:</strong><br>  
 <?= nl2br(htmlspecialchars($art->display())) ?>  
 </div>  
  
 <div>  
 <strong>Автор:</strong> <?= htmlspecialchars($art->getAuthor()) ?><br>  
 <strong>Год создания:</strong> <?= $art->getYear() ?><br>  
 <strong>Базовая стоимость:</strong> <?= number\_format($art->getBaseValue(), 2, '.', ' ') ?> руб.<br>  
 <strong>Текущая оценка:</strong> <span class="value"><?= number\_format($art->appraiseValue(), 2, '.', ' ') ?> руб.</span>  
 </div>  
  
 <div style="margin-top: 10px; font-style: italic;">  
 <strong>Полная информация:</strong> <?= htmlspecialchars($art->info()) ?>  
 </div>  
  
 <?php if (method\_exists($art, 'getStyle')): ?>  
 <div><small>Стиль: <?= htmlspecialchars($art->getStyle()) ?></small></div>  
 <?php endif; ?>  
  
 <?php if (method\_exists($art, 'getMaterial')): ?>  
 <div><small>Материал: <?= htmlspecialchars($art->getMaterial()) ?></small></div>  
 <?php endif; ?>  
  
 <?php if (method\_exists($art, 'getFormat')): ?>  
 <div><small>Формат: <?= htmlspecialchars($art->getFormat()) ?></small></div>  
 <?php endif; ?>  
 </div>  
 <?php endforeach; ?>  
  
 <div style="margin-top: 30px; padding: 20px; background: #34495e; color: white; border-radius: 10px;">  
 <h3>📊 Статистика коллекции</h3>  
 <?php  
 $totalValue = array\_reduce($artCollection, fn($sum, $art) => $sum + $art->appraiseValue(), 0);  
 $averageValue = $totalValue / count($artCollection);  
 $mostValuable = array\_reduce($artCollection, fn($max, $art) =>  
 $art->appraiseValue() > $max->appraiseValue() ? $art : $max, $artCollection[0]);  
 ?>  
 <p><strong>Общая стоимость коллекции:</strong> <?= number\_format($totalValue, 2, '.', ' ') ?> руб.</p>  
 <p><strong>Средняя стоимость произведения:</strong> <?= number\_format($averageValue, 2, '.', ' ') ?> руб.</p>  
 <p><strong>Самое ценное произведение:</strong> "<?= htmlspecialchars($mostValuable->getTitle()) ?>"  
 (<?= number\_format($mostValuable->appraiseValue(), 2, '.', ' ') ?> руб.)</p>  
 </div>  
</div>  
</body>  
</html>

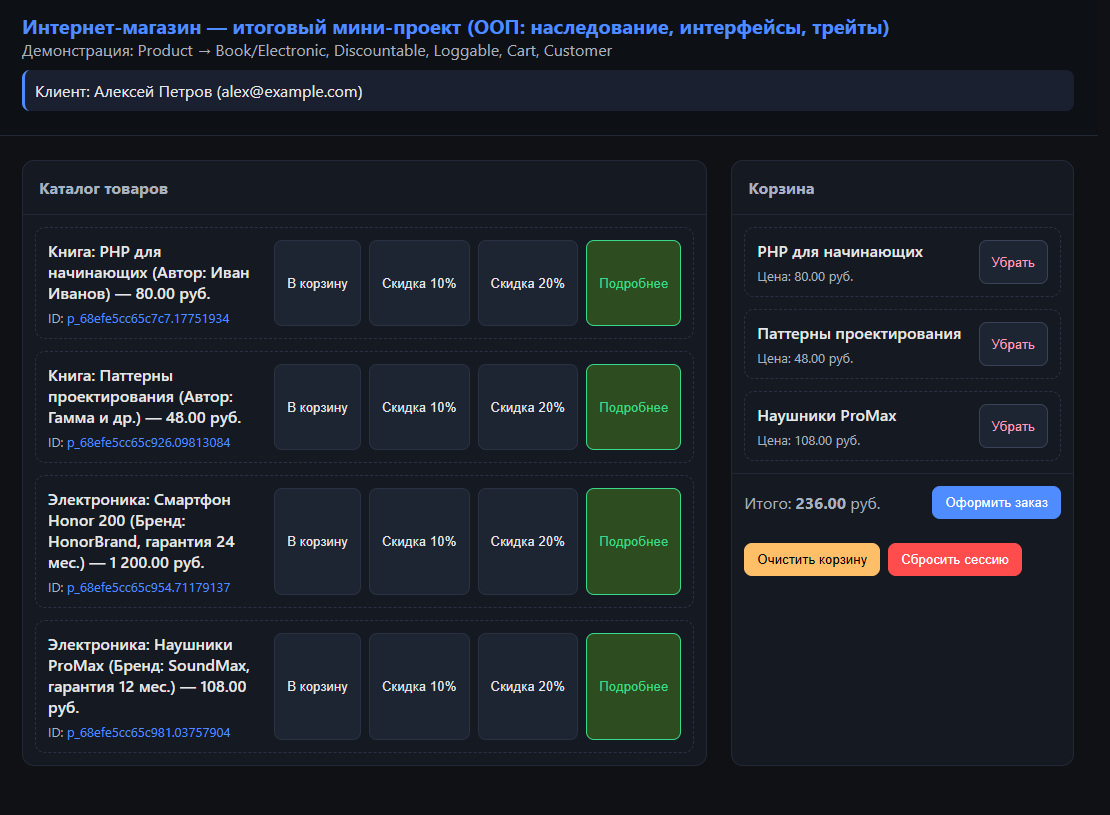


Рисунок 1 – Интернет-магазин

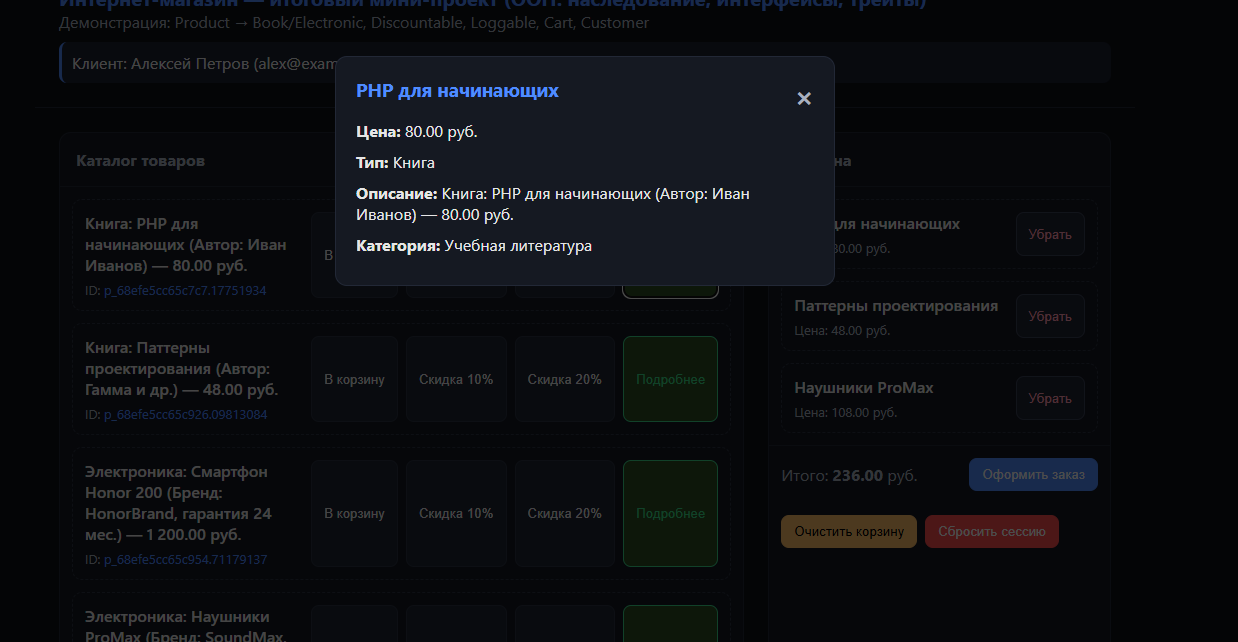


Рисунок 2 – «Подробнее»

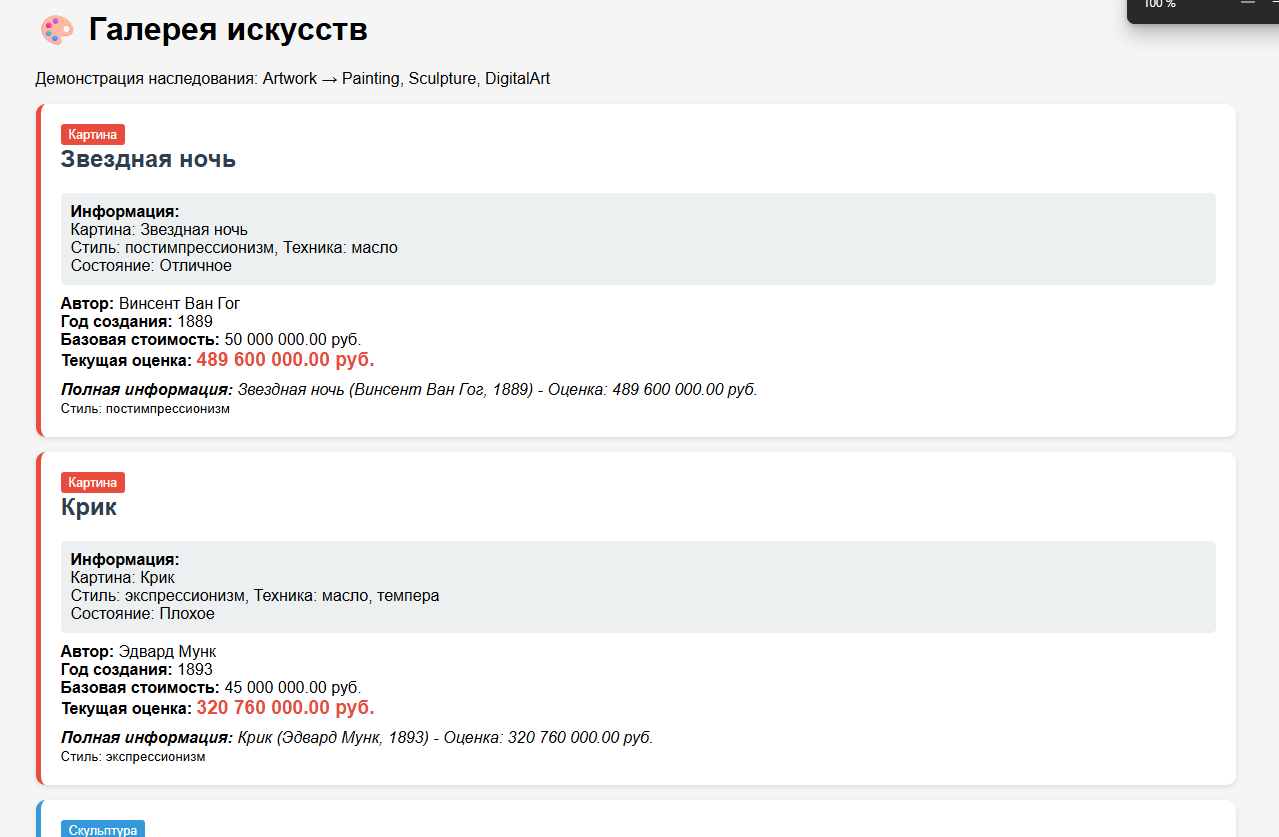


Рисунок 3 – 2 задание

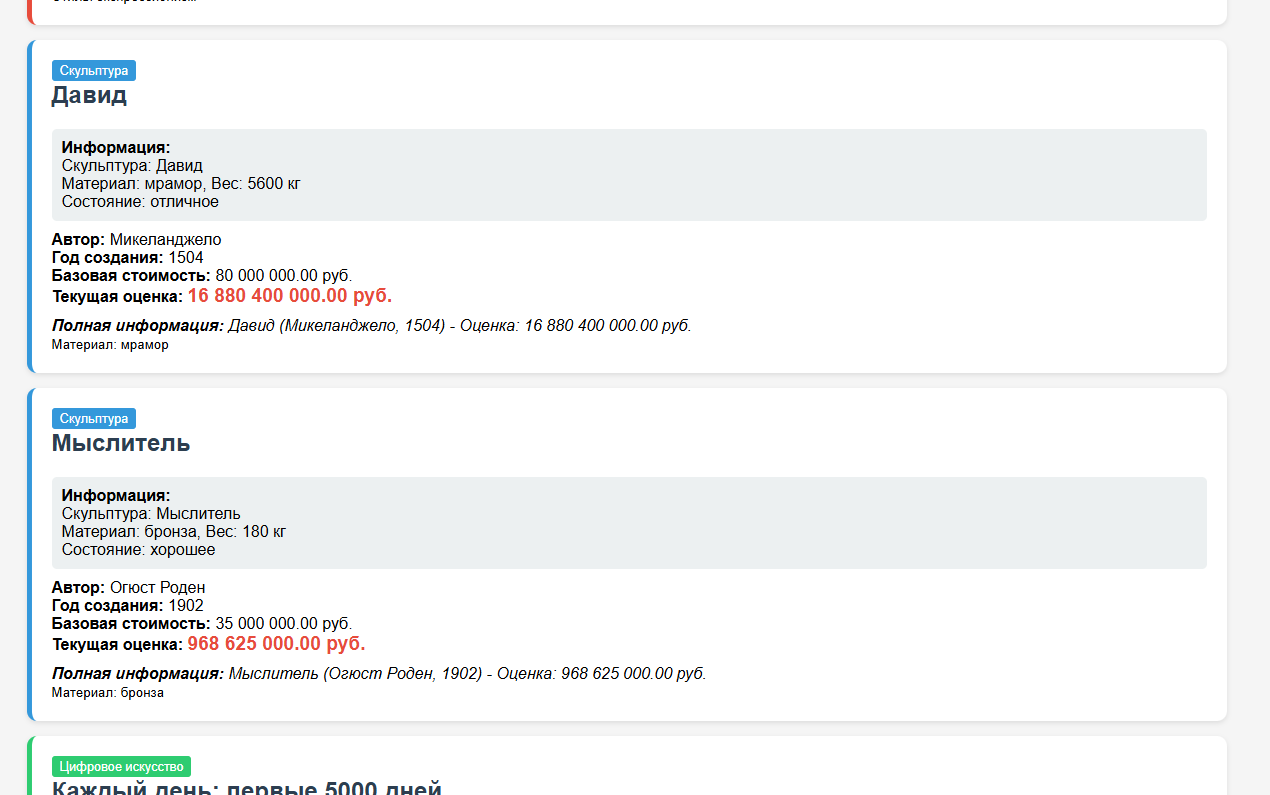


Рисунок 4 – 2 задание

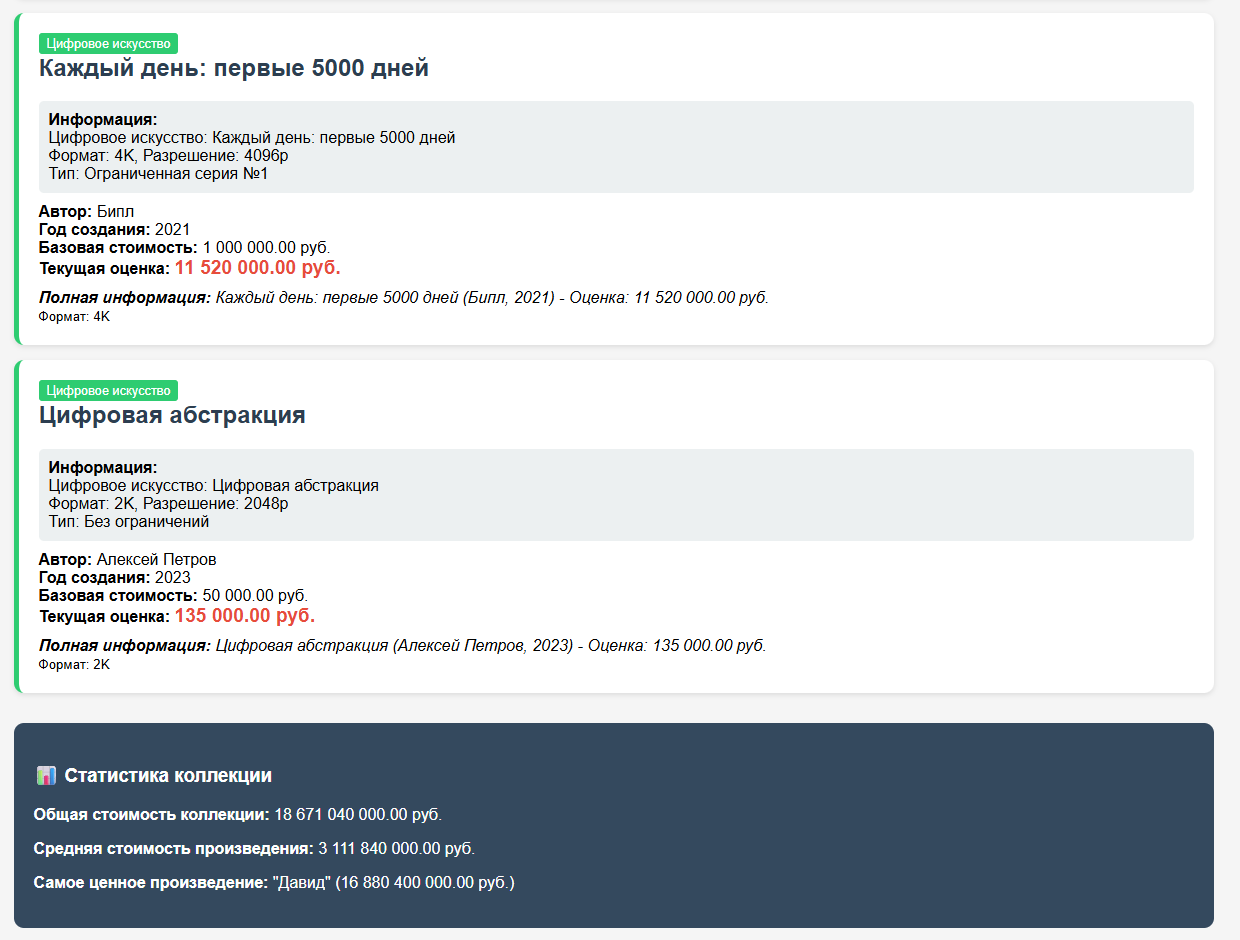


Рисунок 5 – Задание 2