

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Комарова Светлана Юриевна
Должность: Проректор по образовательной деятельности
Дата подписания: 30.07.2026 21:16:02
Уникальный программный ключ:
43ba42f5deae4116bbfcb9ac98e391080312f5e3110c4d4c41b9e0c

Приложение

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Омский государственный аграрный университет
имени П.А. Столыпина»**

Университетский колледж агробизнеса

09.02.11 Разработка и управление программным обеспечением

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по ПП.03.01 Производственная практика (по профилю специальности)**

Обеспечивающее преподавание дисциплины
подразделение

Инженерное отделение

Разработчик:

Преподаватель

А.В. Кортусов

**Омск
2026**

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	3
2. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ	4
3. РАСПРЕДЕЛЕНИЕ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ И ТИПОВ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ЭЛЕМЕНТАМ ЗНАНИЙ И УМЕНИЙ	6
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ	7
5. ПОКАЗАТЕЛИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ	12

1. ОБЩИЕ ПОЛОЖЕНИЯ

1. Фонд оценочных средств (далее – ФОС) предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу ПП 03.01 Производственная практика.
2. ФОС включает оценочные материалы для проведения текущего контроля и промежуточной аттестации в форме зачета .
- 3.ФОС позволяет оценивать знания, умения, направленные на формирование компетенций
4. ФОС разработан на основании положений основной образовательной программы по специальности 09.02.11 Разработка и управление программным обеспечением и рабочей программы ПП 03.01 Производственная практика
5. ФОС является обязательным обособленным приложением к рабочей программе.

2. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ

Результаты обучения (освоенные навыки, умения)	Показатели оценки образовательных результатов
анализа деятельности организации	Обучающийся владеет навыками анализа деятельности организации
выявления потребностей в автоматизации	Обучающийся владеет навыками выявления потребностей в автоматизации
составления описания требований	Обучающийся владеет навыками составления описания требований
разработки общей схемы работы системы	Обучающийся владеет навыками разработки общей схемы работы системы
определения состава и связей частей	Обучающийся владеет навыками определения состава и связей частей
проектирования экранных форм и отчетов	Обучающийся владеет навыками проектирования экранных форм и отчетов
описания правил обработки информации	Обучающийся владеет навыками описания правил обработки информации
подбора технических средств	Обучающийся владеет навыками подбора технических средств
тестирования информационной системы	Обучающийся владеет навыками тестирования информационной системы
подготовки документов для разработчиков	Обучающийся владеет навыками подготовки документов для разработчиков
согласования решений с заказчиком	Обучающийся владеет навыками согласования решений с заказчиком
оценки сроков и стоимости создания	Обучающийся владеет навыками оценки сроков и стоимости создания
учета требований безопасности	Обучающийся владеет навыками учета требований безопасности
проводить изучение деятельности организации	Обучающийся умеет проводить изучение деятельности организации
выявлять потребности в автоматизации	Обучающийся умеет выявлять потребности в автоматизации
составлять описание требований к будущей системе	Обучающийся умеет составлять описание требований к будущей системе
разрабатывать общую схему работы системы	Обучающийся умеет разрабатывать общую схему работы системы
определять состав и взаимосвязи частей системы	Обучающийся умеет определять состав и взаимосвязи частей системы
проектировать экранные формы и отчеты	Обучающийся умеет проектировать экранные формы и отчеты
описывать правила обработки информации	Обучающийся умеет описывать правила обработки информации
определять необходимый состав технических средств	Обучающийся умеет определять необходимый состав технических средств
осуществлять тестирование информационной системы	Обучающийся умеет осуществлять тестирование информационной системы

готовить документы для разработчиков	Обучающийся умеет готовить документы для разработчиков
согласовывать проектные решения с заказчиком	Обучающийся умеет согласовывать проектные решения с заказчиком
оценивать сроки и стоимость создания системы	Обучающийся умеет оценивать сроки и стоимость создания системы
учитывать требования безопасности информации	Обучающийся умеет учитывать требования безопасности информации

3. РАСПРЕДЕЛЕНИЕ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ И ТИПОВ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ЭЛЕМЕНТАМ ЗНАНИЙ И УМЕНИЙ

Содержание курса	Форма контроля	Умения	Навыки
Текущий контроль			
Раздел 1. Организационный этап			
Прохождение вводного инструктажа. Получение и обсуждение задания на практику.	Проверка отчета по учебной практике	Уо 02.01, Уо 02.02, Уо 05.01	-
Раздел 2. Основной этап			
Выполнение работ.	наблюдение и оценка в процессе практики; анализ отчетной документации; экспертная оценка выполнения индивидуальных заданий.	У 3.1.1, У 3.1.2, У 3.2.1, У 3.2.2, У 3.3.1, У 3.4.1, У 3.4.2, У 3.5.1, У 3.6.1, У 3.7.1, У 3.7.2, У 3.8.1, У 3.8.2	Н 3.1.1, Н 3.1.2, Н 3.2.1, Н 3.2.2, Н 3.3.1, Н 3.4.1, Н 3.4.2, Н 3.5.1, Н 3.6.1, Н 3.7.1, Н 3.7.2, Н 3.8.1, Н 3.8.2, Н 3.8.3
Раздел 3. Заключительный этап			
Оформление введения. Оформление основной части. Оформление заключения. Оформление списка использованных источников и приложений. Оформление отчета и приложений. Прохождение собеседования (зачет)	Проверка отчета по учебной практике, собеседование	Уо 02.01, Уо 02.02, Уо 05.01	-
Промежуточный контроль			
Зачет	Проверка отчета по учебной практике, собеседование	У 3.1.1, У 3.1.2, У 3.2.1, У 3.2.2, У 3.3.1, У 3.4.1, У 3.4.2, У 3.5.1, У 3.6.1, У 3.7.1, У 3.7.2, У 3.8.1, У 3.8.2, Уо 02.01, Уо 02.02, Уо 05.01	Н 3.1.1, Н 3.1.2, Н 3.2.1, Н 3.2.2, Н 3.3.1, Н 3.4.1, Н 3.4.2, Н 3.5.1, Н 3.6.1, Н 3.7.1, Н 3.7.2, Н 3.8.1, Н 3.8.2, Н 3.8.3

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОЦЕНКИ УМЕНИЙ, НАВЫКОВ

4.1. Оценочные средства, применяемые для текущего контроля.

4.1.1. Разработка полнофункциональной ИС для управления заказами

Разработайте информационную систему для интернет-магазина:

1. Проектирование БД (HeidiSQL):
Товары, Категории, Клиенты, Заказы, Позиции заказа, Способы оплаты.
Не менее 6 таблиц с правильными связями.
Индексы для ускорения запросов.
2. Серверная часть (Flask + PyMySQL):
Полный набор CRUD-операций.
Создание заказа с проверкой наличия товаров.
Изменение статуса заказа.
Формирование отчетов.

Пример кода:

```
# models.py
from flask import Flask, request, jsonify
import pymysql
from datetime import datetime

class OrderSystem:
    def __init__(self):
        self.conn = pymysql.connect(
            host='localhost', user='root', password="",
            database='store', cursorclass=pymysql.cursors.DictCursor
        )

    def create_order(self, customer_id, items):
        """Создание заказа: items = [{'product_id': 1, 'quantity': 2}]"""
        total = 0
        with self.conn.cursor() as cursor:
            # Проверка наличия и расчет суммы
            for item in items:
                cursor.execute('SELECT price, stock FROM Products WHERE id = %s',
                    (item['product_id'],))
                product = cursor.fetchone()
                if not product or product['stock'] < item['quantity']:
                    return {'error': f'Товар {item["product_id"]} недоступен'}
                total += product['price'] * item['quantity']

            # Создание заказа
            cursor.execute("""
                INSERT INTO Orders (customer_id, order_date, status, total_amount)
                VALUES (%s, %s, 'new', %s)
            """, (customer_id, datetime.now(), total))
            order_id = cursor.lastrowid

            # Добавление позиций и обновление остатков
            for item in items:
```

```

        cursor.execute('SELECT price FROM Products WHERE id = %s',
(item['product_id'],))
        price = cursor.fetchone()['price']

        cursor.execute("""
            INSERT INTO OrderItems (order_id, product_id, quantity, price_at_time)
            VALUES (%s, %s, %s, %s)
            """, (order_id, item['product_id'], item['quantity'], price))

        cursor.execute('UPDATE Products SET stock = stock - %s WHERE id = %s',
            (item['quantity'], item['product_id']))

        self.conn.commit()
        return {'order_id': order_id, 'total': total}

```

Форма выполнения: Исходный код, скриншоты работы, тесты.

4.1.2. Разработка REST API и документации

Создайте REST API со следующей документацией:

```

# api.py
from flask import Flask, request, jsonify
from models import OrderSystem

app = Flask(__name__)
system = OrderSystem()

@app.route('/api/products', methods=['GET'])
def get_products():
    """Получение списка товаров с фильтрацией по категории"""
    category = request.args.get('category')
    with system.conn.cursor() as cursor:
        if category:
            cursor.execute('SELECT * FROM Products WHERE category = %s', (category,))
        else:
            cursor.execute('SELECT * FROM Products')
    return jsonify(cursor.fetchall())

@app.route('/api/orders', methods=['POST'])
def create_order():
    """Создание заказа"""
    data = request.json
    result = system.create_order(data['customer_id'], data['items'])
    if 'error' in result:
        return jsonify(result), 400
    return jsonify(result), 201

@app.route('/api/orders/<int:order_id>/status', methods=['PUT'])
def update_status(order_id):
    """Обновление статуса заказа"""
    status = request.json.get('status')
    valid_statuses = ['new', 'processing', 'shipped', 'delivered', 'cancelled']

```

```

if status not in valid_statuses:
    return jsonify({'error': 'Invalid status'}), 400

with system.conn.cursor() as cursor:
    cursor.execute('UPDATE Orders SET status = %s WHERE id = %s', (status, order_id))
    system.conn.commit()
    return jsonify({'message': 'Status updated'})

@app.route('/api/reports/sales', methods=['GET'])
def sales_report():
    """Отчет по продажам за период"""
    date_from = request.args.get('from', datetime.now().strftime('%Y-%m-01'))
    date_to = request.args.get('to', datetime.now().strftime('%Y-%m-%d'))

    with system.conn.cursor() as cursor:
        cursor.execute("""
            SELECT p.name, SUM(oi.quantity) as total_sold, SUM(oi.price_at_time *
oi.quantity) as total_revenue
            FROM OrderItems oi
            JOIN Products p ON oi.product_id = p.id
            JOIN Orders o ON oi.order_id = o.id
            WHERE o.order_date BETWEEN %s AND %s AND o.status = 'delivered'
            GROUP BY oi.product_id
            ORDER BY total_revenue DESC
            """, (date_from, date_to))
        return jsonify(cursor.fetchall())

if __name__ == '__main__':
    app.run(debug=True)

```

Форма выполнения: Исходный код API, документация в формате Swagger/OpenAPI.

4.1.3. Тестирование и отладка производственной системы

Разработайте комплексные тесты:

```

# test_integration.py
import pytest
import json
from api import app

@pytest.fixture
def client():
    app.config['TESTING'] = True
    with app.test_client() as client:
        yield client

def test_full_order_flow(client):
    """Полный цикл: добавление товара → создание заказа → изменение статуса"""

    # 1. Добавление товара
    add_resp = client.post('/api/products', json={
        'name': 'Ноутбук',

```

```

    'price': 50000,
    'stock': 5,
    'category': 'Электроника'
})
assert add_resp.status_code == 201
product_id = json.loads(add_resp.data)['id']

# 2. Создание заказа
order_resp = client.post('/api/orders', json={
    'customer_id': 1,
    'items': [{'product_id': product_id, 'quantity': 2}]
})
assert order_resp.status_code == 201
order_id = json.loads(order_resp.data)['order_id']

# 3. Проверка остатка
get_resp = client.get(f'/api/products/{product_id}')
assert json.loads(get_resp.data)['stock'] == 3

# 4. Изменение статуса
status_resp = client.put(f'/api/orders/{order_id}/status', json={'status': 'delivered'})
assert status_resp.status_code == 200

```

Форма выполнения: Тесты, отчет о результатах, журнал ошибок.

4.1.4. Развертывание и администрирование системы

Настройте развертывание информационной системы:

1. Настройка HeidiSQL:
 - a. Создание пользователей и ролей.
 - b. Настройка прав доступа.
2. Настройка сервера Python:
 - a. Запуск Flask через Gunicorn.
 - b. Настройка переменных окружения.
3. Мониторинг:
 - a. Логирование запросов.
 - b. Мониторинг производительности.

```

# gunicorn.conf.py
import multiprocessing

bind = "0.0.0.0:8000"
workers = multiprocessing.cpu_count() * 2 + 1
worker_class = "sync"
accesslog = "/var/log/app/access.log"
errorlog = "/var/log/app/error.log"
loglevel = "info"

# config.py
import os

class Config:
    DB_HOST = os.getenv('DB_HOST', 'localhost')

```

```
DB_USER = os.getenv('DB_USER', 'app_user')
DB_PASSWORD = os.getenv('DB_PASSWORD', 'secure_password')
DB_NAME = os.getenv('DB_NAME', 'store')
SECRET_KEY = os.getenv('SECRET_KEY', 'dev-secret-key')
DEBUG = os.getenv('DEBUG', 'False') == 'True'
```

Форма выполнения: Конфигурационные файлы, инструкция по развертыванию, логи.

4.2. Оценочные средства, применяемые для промежуточной аттестации по итогам изучения дисциплины

Зачет проводится по завершении производственной практики.

Промежуточная аттестация в форме зачета осуществляется по результатам сдачи отчета по практике и с учетом текущего контроля успеваемости при выполнении всех видов текущего контроля.

Обучающиеся, не выполнившие виды работ, предусмотренные рабочей программой; пропустившие более 50% практики без уважительной причины, не допускаются к зачету.

Промежуточная аттестация таких лиц проводится только после прохождения ими всех видов текущего контроля.

**V. ПОКАЗАТЕЛИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОСВОЕНИЯ
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

Уровень сформированности компетенций	Оценка	Критерии оценивания по видам работ	
		тестирование (процент правильных ответов)	прочие виды работ по практике
Высокий	Отлично	90-100%	Обучающийся глубоко и прочно усвоил теоретический и освоил практический материал. Дает логичные и грамотные ответы. Демонстрирует знание не только основного, но и дополнительного материала, быстро ориентируется, отвечая на дополнительные вопросы. Свободно справляется с поставленными задачами, аргументировано и верно обосновывает принятые решения.
Повышенный	Хорошо	70-89%	Обучающийся твердо знает программный материал, грамотно и по существу излагает его. Не допускает существенных неточностей при ответах на вопросы, правильно применяет теоретические положения при решении практических задач, владеет навыками и приемами их выполнения.
Базовый	Удовлетворительно	50-69%	Обучающийся демонстрирует знания только основного материала, но не усвоил его детали, испытывает затруднения при решении практических задач. В ответах на поставленные вопросы допускает неточности. Дает определения понятий, искажающие их смысл. Нарушает последовательность изложения программного материала.
Не сформирована	Неудовлетворительно	0-49%	Обучающийся не знает, не выполняет или неправильно выполняет большую часть учебного материала. Допускает ошибки в формулировке определений, искажающие их смысл, беспорядочно и неуверенно излагает материал. Ответы на дополнительные вопросы отсутствуют. Не выполняет задания.

ЛИСТ РАССМОТРЕНИЙ И ОДОБРЕНИЙ

рабочей программы

ПП 03.01 Производственная практика (по профилю специальности)

в составе ООП 09.02.11 Разработка и управление программным обеспечением

1) <u>Рассмотрена и одобрена:</u>
а) На заседании предметно-цикловой методической комиссии протокол № 6 от 10.03.2026 г. Председатель ПЦМК  С.М. Нурбаева
б) На заседании методического совета протокол №4 от 21.04.2026 г. Председатель методического совета  М.В. Иваницкая
2) <u>Рассмотрена и одобрена внешним экспертом</u>
а) директор ООО «Сатори Партнер» А.Б. Мальцев