

## Практическое занятие № 15

**Тема:** составление программ для работы с базами данных в IDE PyCharm Community

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community

**Постановка задачи №1:** Приложение ЗАКАЗЫ ТОВАРОВ для автоматизированного контроля заказов торговой фирмы. Таблица Заказы должна содержать информацию о товарах со следующей структурой записи: Код товара, Наименование товара, Заказчик (наименование организации, возможны повторяющиеся значения), Дата заказа, Срок исполнения (от 1 до 10 дней), Стоимость заказа.

**Текст программы №1:**

```
import pandas as pd

class Order:
    """Class to represent an order in the order management system."""

    def __init__(self, product_code, product_name, customer_name,
order_date, delivery_period, order_cost):
        self.product_code = product_code
        self.product_name = product_name
        self.customer_name = customer_name
        self.order_date = order_date
        self.delivery_period = delivery_period
        self.order_cost = order_cost

class OrderManagementSystem:
    """Class to manage orders in the order management system."""

    def __init__(self):
        self.orders = []

    def add_order(self, order):
        """Add a new order to the system."""
        self.orders.append(order)

    def display_orders(self):
        """Display all orders in the system."""
        print("**Список заказов**")
        for order in self.orders:
            print(f"Код товара: {order.product_code}")
            print(f"Название товара: {order.product_name}")
            print(f"Заказчик: {order.customer_name}")
            print(f"Дата заказа: {order.order_date}")
```

```

        print(f"Срок исполнения: {order.delivery_period} дней")
        print(f"Стоимость заказа: {order.order_cost:.2f}")
        print("-" * 20)

def search_orders_by_customer(self, customer_name):
    """Search for orders by customer name."""
    matching_orders = []
    for order in self.orders:
        if order.customer_name == customer_name:
            matching_orders.append(order)

    if matching_orders:
        print(f"**Заказы для {customer_name}**")
        for order in matching_orders:
            print(f"Код товара: {order.product_code}")
            print(f"Название товара: {order.product_name}")
            print(f"Дата заказа: {order.order_date}")
            print(f"Срок исполнения: {order.delivery_period} дней")
            print(f"Стоимость заказа: {order.order_cost:.2f}")
            print("-" * 20)
    else:
        print(f"Заказов для {customer_name} не найдено.")

def save_orders_to_csv(self, filename):
    """Save orders to a CSV file."""
    order_data = []
    for order in self.orders:
        order_data.append({
            "Код товара": order.product_code,
            "Название товара": order.product_name,
            "Заказчик": order.customer_name,
            "Дата заказа": order.order_date,
            "Срок исполнения": order.delivery_period,
            "Стоимость заказа": order.order_cost,
        })

    orders_df = pd.DataFrame(order_data)
    orders_df.to_csv(filename, index=False)
    print(f"Заказы успешно сохранены в файл {filename}.")

# Create an order management system instance
oms = OrderManagementSystem()

```

```
# Add some sample orders
order1 = Order("PR123", "Смартфон", "ООО 'Техномир'", "2024-06-25", 5,
23500.00)
order2 = Order("LT456", "Ноутбук", "ООО 'Лайф'", "2024-06-22", 7,
48900.00)
order3 = Order("PR123", "Смартфон", "ООО 'Техномир'", "2024-06-23", 5,
33000.00)

oms.add_order(order1)
oms.add_order(order2)
oms.add_order(order3)

# Display all orders
oms.display_orders()
```

**Вывод:** я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.