

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Алтайский государственный технический университет им. И.И.
Ползунова»
Университетский технологический колледж
наименование подразделения
Кафедра Информационные системы в экономике
наименование кафедры
Направление Информационные системы и программирование

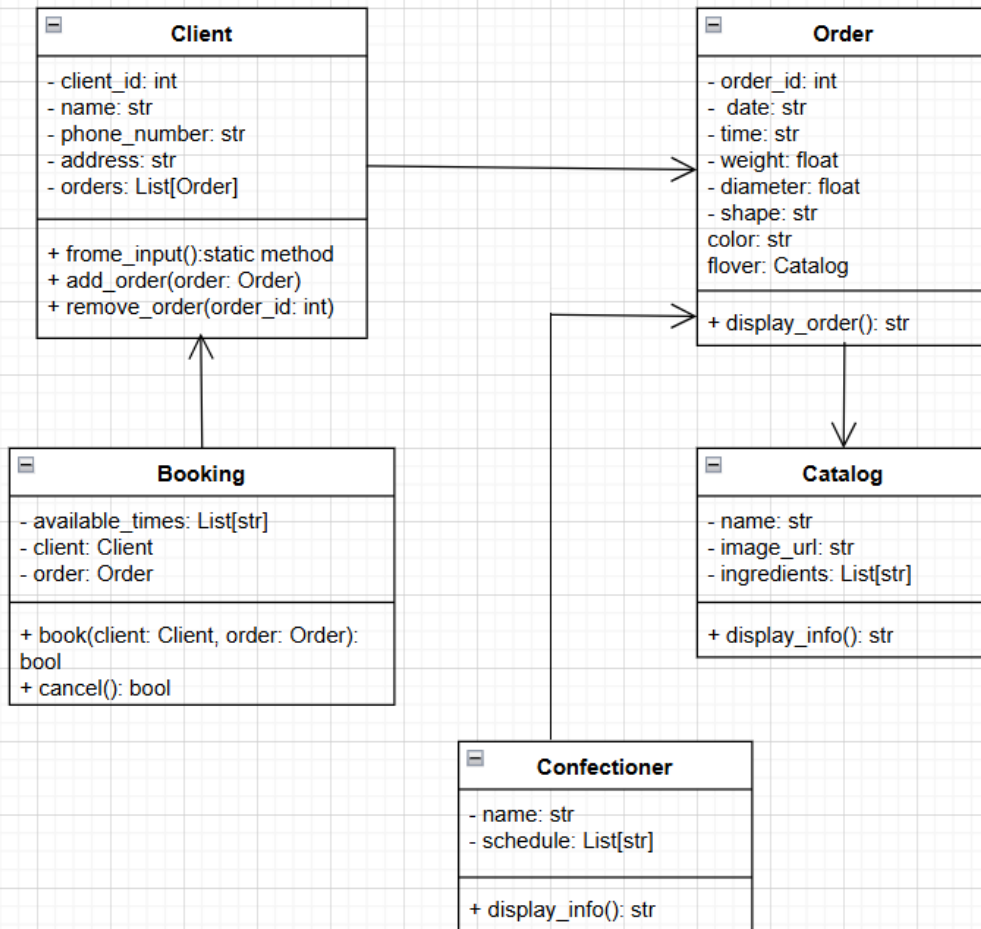
Отчёт защищен с оценкой _____
_____ К.В Воробьев
(подпись руководителя от вуза) (инициалы, фамилия)
“ ____ ” _____ 20 ____ г.

ОТЧЕТ
по лабораторной работе №2
Реализация АТД _____
тема лабораторной работы
по дисциплине Объектно-ориентированное программирование
ЛР 09.02.07.09.001 ПЗ
обозначение документа

Студент группы _____ 1ИСП-21 _____ А.И. Шнель
инициалы, фамилия

Руководитель работы преподаватель _____ К.В. Воробьев
должность, ученое звание инициалы, фамилия

Барнаул 2025



2 usages

```
class Client:
    def __init__(self, client_id, name, phone_number, address):
        self.client_id = client_id
        self.name = name
        self.phone_number = phone_number
        self.address = address
        self.orders = []
```

1 usage

@staticmethod

```
def from_input():
    name = input("Введите полное имя клиента: ")
    phone_number = input("Введите номер телефона клиента: ")
    address = input("Введите адрес клиента: ")
    return name, phone_number, address

def add_order(self, order):
    self.orders.append(order)

def remove_order(self, order_id):
    self.orders = [order for order in self.orders if order.order_id != order_id]
```

class Order:

```
    def __init__(self, order_id, date, time, weight, diameter, shape, color, flavor):
        self.order_id = order_id
        self.date = date
        self.time = time
```

```

class Order:
    def __init__(self, order_id, date, time, weight, diameter, shape, color, flavor):
        self.order_id = order_id
        self.date = date
        self.time = time
        self.weight = weight
        self.diameter = diameter
        self.shape = shape
        self.color = color
        self.flavor = flavor # Ссылка на объект из класса Catalog

    def display_order(self):
        return (f"Заказ ID: {self.order_id}, Дата: {self.date}, Время: {self.time}, "
                f"Вес: {self.weight} кг, Диаметр: {self.diameter} см, "
                f"Форма: {self.shape}, Цвет: {self.color}, Вкус: {self.flavor.name}")

```

2 usages

```

class Catalog:
    def __init__(self, name, image_url, ingredients):
        self.name = name
        self.image_url = image_url
        self.ingredients = ingredients

    def display_info(self):
        return (f"Торт: {self.name}, Фото: {self.image_url}, "
                f"Ингредиенты: {'', '.join(self.ingredients)}")

```

```

class Confectioner:
    def __init__(self, name, schedule):
        self.name = name
        self.schedule = schedule

    def display_info(self):
        return f"Кондитер: {self.name}, Расписание: {'', '.join(self.schedule)}"

```

1 usage

```

class Booking:
    def __init__(self, available_times):
        self.available_times = available_times
        self.client = None
        self.order = None

    def book(self, client, order):
        if order.time in self.available_times:
            self.client = client
            self.order = order
            self.available_times.remove(order.time)
            return True
        return False

```

```

def cancel(self):
    if self.order:
        self.available_times.append(self.order.time)
        self.client = None
        self.order = None
        return True
    return False

def main():
    clients = []

    catalog = [
        Catalog(name="Шоколадный торт", image_url="url_to_image1", ingredients=["мука", "сахар", "какао", "яйца"]),
        Catalog(name="Торт Сникерс", image_url="url_to_image2", ingredients=["мука", "сахар", "карамель", "арахис"]),
    ]

    confectioner = Confectioner(name="Анна", schedule=["Пн-Вс: 10:00-16:00"])

    number_of_clients = int(input("Введите количество клиентов: "))
    for i in range(number_of_clients):
        print(f"\nКлиент {i + 1}:")
        name, phone_number, address = Client.from_input()
        client = Client(i + 1, name, phone_number, address)
        clients.append(client)

    print("\nСписок клиентов:")

    number_of_clients = int(input("Введите количество клиентов: "))
    for i in range(number_of_clients):
        print(f"\nКлиент {i + 1}:")
        name, phone_number, address = Client.from_input()
        client = Client(i + 1, name, phone_number, address)
        clients.append(client)

    print("\nСписок клиентов:")
    for client in clients:
        print(f"{client.name}, Телефон: {client.phone_number}, Адрес: {client.address}")

    booking = Booking(["10:00", "11:00", "12:00", "13:00", "14:00"])

    for client in clients:
        order_id = int(input(f"\nВведите ID заказа для клиента {client.name}: "))
        date = input("Введите дату заказа: ")
        time = input("Введите время заказа: ")
        weight = float(input("Введите вес торта (кг): "))
        diameter = float(input("Введите диаметр торта (см): "))
        shape = input("Введите форму торта (круг, квадрат, сердце): ")
        color = input("Введите цвет торта: ")

```

https://github.com/shnelanna29/OOP/blob/d058217fb7c7bcd37b91f23bd8fa270f40a25868/TGbot_Conditer.py