

**Лабораторная работа состоит из нескольких заданий, где показаны основные принципы объектно-ориентированного программирования (ООП):
Абстракция, Наследование, Полиморфизм и Инкапсуляция.**

1 Первая часть — Payment (оплата).

Создан абстрактный класс `Payment` с методами `pay()` и `refund()`.

От него наследуются классы `CreditCardPayment` и `CryptoPayment`.

Каждый выполняет оплату и возврат по-своему.

В конце создаётся список способов оплаты и выполняется оплата и возврат.

Это показывает **наследование и полиморфизм** — разные классы по-разному выполняют одну и ту же функцию.

2 Вторая часть — Course (курсы).

Создан абстрактный класс `Course` и два наследника — `PythonCourse` и `MathCourse`.

У каждого свои материалы и логика запуска курса.

Показывает **абстракцию и наследование**.

3 Третья часть — Delivery (доставка).

Есть абстрактный класс `Delivery` и три варианта доставки: `AirDelivery`, `GroundDelivery`, `SeaDelivery`.

Каждая рассчитывает стоимость по своей формуле.

Это пример **полиморфизма** — разные способы доставки делают одно действие по-разному.

4 Четвёртая часть — BankAccount (банк).

Используются **приватные поля** (`__owner`, `__balance`, `__pin`) для защиты данных.

Методы `deposit()`, `withdraw()` и `change_pin()` работают только при правильном PIN-коде.

Это пример **инкапсуляции** — скрывание личных данных.

5 Пятая часть — UserProfile (профиль пользователя).

Созданы методы для входа, обновления статуса и получения информации.

Пароль скрыт, а статус можно изменить на Premium.

Показывает **защиту данных** и работу с приватными полями.

6 Шестая часть — Product (товар).

У товара есть название, цена и скрытая скидка.

Только администратор может установить скидку.

Это снова **инкапсуляция** — доступ ограничен.

7 Седьмая часть — файлы (TextFile, ImageFile, AudioFile).

Каждый тип файла имеет свой метод `open()`.

Функция `open_all()` открывает все файлы из списка.

Это пример **полиморфизма**.

8) Восьмая часть — транспорт (Car, Truck, Bicycle).

Каждый вид транспорта имеет метод `move()` и свою скорость.

Функция `simulate_transport()` показывает, за сколько времени каждый проедет расстояние.

Это тоже **полиморфизм**.

9) Девятая часть — роли пользователей (Student, Teacher, Administrator).

Каждый имеет свой метод `access_portal()` и разные права доступа.

Все объекты собраны в один список, и у каждого вызывается свой метод.

Это показывает **наследование и полиморфизм**.