Ассount

Этот класс — про банковский счёт. Там есть номер, баланс и PIN-код. Всё скрыто от прямого доступа.

Чтобы пополнить или снять деньги, нужно ввести правильный PIN. Если он неправильный — ничего не получится. Это защита от несанкционированного доступа. Также можно посмотреть баланс, но только при верном PIN.

Product

Класс Product — это про товар с ценой. Я сделал так, чтобы цена могла уменьшаться через скидку, но при этом не могла стать меньше нуля. Метод set\_discount применяет скидку, а final\_price() показывает конечную цену после скидки.

Course

Это курс с ограниченным числом мест. Есть список студентов, и он приватный. Можно добавить студента, если ещё есть места, или удалить. Метод get\_students() просто показывает список, но не даёт его менять напрямую — я возвращаю кортеж.

SmartWatch

Класс умных часов с зарядом батареи. Если использовать часы, заряд падает. Если заряжать — поднимается, но максимум 100%. Всё логично и с защитой от выхода за границы.

Transport

Базовый класс Transport задаёт скорость и вместимость. Метод считает, за сколько времени транспорт проедет нужную дистанцию. У Airplane я переопределил метод, чтобы он был быстрее — на 20%. У поездов и автобусов расчёт обычный.

Классы Order, DineInOrder, TakeAwayOrder, DeliveryOrder

Есть заказ с базовой ценой. У каждого типа заказа свой способ посчитать итоговую сумму. В ресторане добавляются чаевые, при доставке — комиссия, а навынос — без изменений. Это пример полиморфизма: метод один и тот же calculate\_total(), а работает по-разному.

Классы Character, Warrior, Mage, Archer

Это герои с атакой. Базовый класс задаёт имя, здоровье и силу атаки. Каждый подкласс по-своему реализует метод attack(): воин дерётся мечом, маг — магией, лучник — луком. Опять же, один метод — разные действия.

Классы MediaFile, AudioFile, VideoFile, Podcast

Базовый класс задаёт медиафайл с названием и длительностью. Метод play() работает по-разному в зависимости от типа: аудио, видео или подкаст. Например, у видео пишется, что есть изображение.

Классы PaymentSystem, CreditCardPayment, CryptoPayment, BankTransfer

Абстрактный класс PaymentSystem требует реализовать метод process\_payment(amount). Подклассы делают это по-разному: один — для карты, другой — для крипты, третий — для перевода. У каждого своё сообщение.

Классы Animal, Lion, Elephant, Snake

Каждое животное по-своему ест и спит. Метод eat() и sleep() определён в каждом классе. Лев спит долго и ест зебру, обезьяна-бананы, змея — мышей. Всё это демонстрирует, как работает наследование и переопределение.

Классы Document, WordDocument, PdfDocument, SpreadsheetDocument

Тут всё про работу с документами. Word можно редактировать, PDF — почти нет. Таблица открывается и редактируется по-своему. Всё оформлено через методы open(), edit(), save().

Классы Lesson, VideoLesson, QuizLesson, TextLesson

Уроки бывают разные: видео, тест, текст. Метод start() запускает их по-разному. У видео — плеер, у теста — первый вопрос, у текста — просто показывает содержание. Это ещё один пример, где один метод делает разное.

Уведомления: EmailNotification, SMSNotification, PushNotification

Каждое уведомление отправляется по-разному. Метод send(message) у всех есть, но реализован иначе: email, SMS или push. В цикле можно вызвать send() у всех, и сообщение уйдёт разными путями.

Фигуры: Square, Circle, Triangle

Каждая фигура умеет считать периметр. У квадрата — 4 стороны, у круга — по формуле с π, у треугольника — сумма всех сторон. Если сделать список фигур, можно пройтись по нему и вызвать perimeter() — всё посчитается правильно без проверки типа.

Сотрудники: Manager, Developer, Designer

Каждый сотрудник делает свою работу. Менеджер управляет, разработчик кодит, дизайнер рисует. Метод work() у всех есть, и вызывается одинаково — это и есть полиморфизм в действии.

Заклинания: FireSpell, IceSpell, HealingSpell

Есть три вида заклинаний. Они применяются к цели через метод cast(target). Огонь наносит урон, лёд замораживает, лечение восстанавливает здоровье. Можно сделать список заклинаний и применить их к разным целям.