

Лаб.8 по ООП.

1) Я создала класс Account с приватными полями номера счёта, баланса и PIN-кода. Методы deposit, withdraw и get_balance проверяют правильность PIN перед выполнением операций, чтобы обеспечить безопасность доступа к счёту.

2)В классе Product цена сделана приватной, чтобы защитить её от прямого изменения. Метод set_discount уменьшает цену, но не допускает отрицательных значений, а метод final_price возвращает итоговую цену с учётом скидки.

3)Класс Course содержит приватные поля: название, список студентов и максимальное количество мест. Методы добавляют или удаляют студентов с учётом лимита, а get_students возвращает копию списка, чтобы предотвратить его изменение извне.

4)SmartWatch хранит уровень заряда в приватном поле. Метод use уменьшает заряд в зависимости от времени использования, charge увеличивает заряд, не позволяя превышать 100%, а get_battery возвращает текущий уровень заряда.

5В базовом классе Transport я реализовала атрибуты скорости и вместимости, а метод travel_time считает время поездки. У подкласса Airplane этот метод переопределён, чтобы уменьшить время на 20%, показывая наследование и полиморфизм.

6)Класс Order и его подклассы DineInOrder, TakeAwayOrder и DeliveryOrder имеют метод calculate_total с разной логикой расчёта итоговой суммы — с учётом чаевых, скидок и доставки соответственно, что демонстрирует полиморфизм.

7)В классе Character и его наследниках Warrior, Mage и Archer каждый реализует метод attack по-разному, что показывает, как работает наследование и полиморфизм.

8)Классы MediaFile и его подклассы AudioFile, VideoFile и Podcast имеют метод play, который выводит различную информацию в зависимости от типа медиафайла, демонстрируя полиморфизм.

9)Абстрактный класс PaymentSystem задаёт метод process_payment, который по-разному реализуют подклассы CreditCardPayment, CryptoPayment и BankTransfer, что иллюстрирует абстракцию и переопределение методов.

10)В абстрактном классе Animal методы eat и sleep реализуются в подклассах Lion, Elephant и Snake по-разному, показывая наследование и индивидуальное поведение животных.

11)Абстрактный класс Document с методами open, edit и save имеет подклассы WordDocument, PdfDocument и SpreadsheetDocument, которые реализуют эти методы по-своему, демонстрируя полиморфизм.

12)Класс Lesson и его подклассы VideoLesson, QuizLesson и TextLesson имеют метод start, который начинается по-разному в зависимости от типа урока, показывая полиморфизм.

13)Классы EmailNotification, SMSNotification и PushNotification имеют одинаковый метод send, который отправляет сообщение разными способами, что демонстрирует единый интерфейс с разной реализацией.

14)В классах фигур Square, Circle и Triangle метод perimeter считает периметр, и в цикле можно вызвать этот метод у объектов разных классов без проверки типа, что показывает полиморфизм.

15)Классы Manager, Developer и Designer имеют метод work, который выводит описание работы каждого сотрудника. В списке сотрудников вызывается одинаковый метод у разных объектов, что тоже пример полиморфизма.

1Классы заклинаний FireSpell, IceSpell и HealingSpell реализуют метод cast, который выводит эффект заклинания для цели. В цикле вызывается этот метод у каждого объекта, демонстрируя одинаковый интерфейс и разное поведение.