**Задание 1: Система управления пользователями**

**Описание:** Разработайте систему управления пользователями с различными типами пользователей, такими как Admin, Moderator, и RegularUser. Все пользователи должны реализовывать интерфейс User, который содержит методы login() и logout().

**Требования:**

1. Интерфейс User должен содержать методы login() и logout().
2. Класс Admin должен иметь дополнительные методы manageUsers() и viewReports().
3. Класс Moderator должен иметь метод moderateContent().
4. Класс RegularUser должен иметь метод postContent().

**Задание 2: Электронная коммерция**

**Описание:** Создайте систему для управления продуктами в интернет-магазине. Определите абстрактный класс Product с общими свойствами и методами, такими как name, price, и метод getDiscountedPrice(). Создайте конкретные классы Electronics, Clothing, и Food.

**Требования:**

1. Абстрактный класс Product должен иметь метод getDiscountedPrice().
2. Класс Electronics должен иметь поле warrantyPeriod и метод applyDiscount() для вычисления скидок.
3. Класс Clothing должен иметь поле size и метод applyDiscount().
4. Класс Food должен иметь поле expirationDate и метод applyDiscount().

**Задание 3: Библиотечная система**

**Описание:** Разработайте систему для управления книгами в библиотеке. Определите интерфейс Borrowable с методами borrow() и returnItem(). Создайте абстрактный класс LibraryItem с полями title и author и конкретные классы Book и Magazine.

**Требования:**

1. Интерфейс Borrowable должен содержать методы borrow() и returnItem().
2. Абстрактный класс LibraryItem должен иметь поля title и author.
3. Класс Book должен реализовывать интерфейс Borrowable и иметь дополнительные поля isbn и publicationYear.
4. Класс Magazine должен реализовывать интерфейс Borrowable и иметь поле issueNumber.

**Задание 4: Система мониторинга оборудования**

**Описание:** Создайте систему мониторинга оборудования на производстве. Определите абстрактный класс Equipment с общими свойствами и методами, такими как status и метод performMaintenance(). Создайте конкретные классы Machine, Sensor, и ConveyorBelt.

**Требования:**

1. Абстрактный класс Equipment должен иметь метод performMaintenance().
2. Класс Machine должен иметь поля serialNumber и lastMaintenanceDate, а также метод performMaintenance().
3. Класс Sensor должен иметь поле sensitivity и метод performMaintenance().
4. Класс ConveyorBelt должен иметь поле beltLength и метод performMaintenance().

**Задание 5: Финансовая система**

**Описание:** Разработайте систему для управления финансовыми транзакциями. Определите интерфейс Transaction с методами process() и cancel(). Создайте абстрактный класс Account с полями balance и методами deposit() и withdraw(). Реализуйте конкретные классы SavingsAccount и CheckingAccount.

**Требования:**

1. Интерфейс Transaction должен содержать методы process() и cancel().
2. Абстрактный класс Account должен иметь поля balance и методы deposit() и withdraw().
3. Класс SavingsAccount должен добавлять метод applyInterest().
4. Класс CheckingAccount должен добавлять метод applyFees().

**Задание 6: Система заказов ресторана**

**Описание:** Создайте систему для управления заказами в ресторане. Определите интерфейс Orderable с методами prepareOrder() и deliverOrder(). Создайте абстрактный класс MenuItem с полем price и конкретные классы Pizza, Pasta, и Salad.

**Требования:**

1. Интерфейс Orderable должен содержать методы prepareOrder() и deliverOrder().
2. Абстрактный класс MenuItem должен иметь поле price.
3. Класс Pizza должен реализовывать интерфейс Orderable и иметь поле size.
4. Класс Pasta должен реализовывать интерфейс Orderable и иметь поле type.
5. Класс Salad должен реализовывать интерфейс Orderable и иметь поле ingredients.

**Задание 7: Система управления проектами**

**Описание:** Разработайте систему для управления проектами, которая включает задачи и подзадачи. Определите абстрактный класс Task с полем description и методами start() и complete(). Создайте конкретные классы SimpleTask и SubTask.

**Требования:**

1. Абстрактный класс Task должен иметь поле description и методы start() и complete().
2. Класс SimpleTask должен реализовывать все методы абстрактного класса.
3. Класс SubTask должен наследовать Task и добавлять метод addSubTask(), который позволяет добавлять подзадачи.

**Задание 8: Игровая система**

**Описание:** Создайте игровую систему для разных типов персонажей в игре. Определите интерфейс Playable с методами attack() и defend(). Создайте абстрактный класс Character с полями name и health. Реализуйте конкретные классы Warrior, Mage, и Archer.

**Требования:**

1. Интерфейс Playable должен содержать методы attack() и defend().
2. Абстрактный класс Character должен иметь поля name и health.
3. Класс Warrior должен реализовывать интерфейс Playable и добавлять метод useShield().
4. Класс Mage должен реализовывать интерфейс Playable и добавлять метод castSpell().
5. Класс Archer должен реализовывать интерфейс Playable и добавлять метод shootArrow().

**Задание 9: Система курсов и студентов**

**Описание:** Создайте систему для управления курсами и студентами. Определите интерфейс Registrable с методами register() и unregister(). Создайте абстрактный класс Person с полями name и email. Реализуйте конкретные классы Student и Instructor.

**Требования:**

1. Интерфейс Registrable должен содержать методы register() и unregister().
2. Абстрактный класс Person должен иметь поля name и email.
3. Класс Student должен реализовывать интерфейс Registrable и добавлять поле studentID.
4. Класс Instructor должен реализовывать интерфейс Registrable и добавлять поле employeeID.