Варианты заданий индивидуальных проектов (Трек 1)

В своих проектах вы должны продемонстрировать различные концепций объектно-ориентированного программирования, такие как инкапсуляция, наследование, полиморфизм, (композиция или абстракция приветствуются) с созданием от базового класса 2-3 производных классов и нескольких экземпляров (объектов) этих классов.

Вариант задания № 1

Описание задачи:

Создать базовый класс Person в C#, который будет представлять информацию о человеке. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Person:

- **Атрибуты:** Имя (Name), Возраст (Age), Пол (Gender).
- **Методы:**
    - GetInfo(): метод для получения информации о человеке в виде строки.
    - SayHello(): метод для вывода приветствия от человека.

Требования к производным классам:

1. Студент (Student): Должен содержать дополнительные атрибуты, такие как Университет (University) и Курс (Course). Метод SayHello() должен быть переопределен для добавления информации о курсе при приветствии.
2. Работник (Employee): Должен содержать дополнительные атрибуты, такие как Компания (Company) и Зарплата (Salary). Метод GetInfo() должен быть переопределен для включения информации о компании и зарплате.
3. Преподаватель (Teacher) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Предмет (Subject) и Стаж работы (Experience). Метод SayHello() должен быть переопределен для добавления информации о преподаваемом предмете при приветствии.

Вариант задания № 2

Описание задачи:

Создать базовый класс Student в C#, который будет представлять информацию о студентах. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Student:

- **Атрибуты:** Имя (Name), Возраст (Age), Курс (Course).
- **Методы:**
    - GetInfo(): метод для получения информации о студенте в виде строки.
    - Study(): метод для вывода сообщения о том, что студент учится.
    - TakeExam(): метод для вывода сообщения о сдаче экзамена.

Требования к производным классам:

1. Студент бакалавриата (BachelorStudent): Должен содержать дополнительные атрибуты, такие как Специальность (Specialty). Метод Study() должен быть переопределен для добавления информации о специальности при изучении предмета.
2. Студент магистратуры (MasterStudent): Должен содержать дополнительные атрибуты, такие как Научный руководитель (ScientificAdvisor). Метод TakeExam() должен быть переопределен для добавления информации о научном руководителе при сдаче экзамена.
3. Аспирант (PhDStudent) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Тема диссертации (DissertationTopic). Метод Study() должен быть переопределен для добавления информации о теме диссертации при изучении предмета.

Вариант задания № 3

Описание задачи:

Создать базовый класс Employee в C#, который будет представлять информацию о сотрудниках компании. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Employee:

- **Атрибуты:** Имя (Name), Возраст (Age), Зарплата (Salary).
- **Методы:**
    - GetInfo(): метод для получения информации о сотруднике в виде строки.
    - Work(): метод для вывода сообщения о выполнении рабочих обязанностей.
    - TakeVacation(): метод для вывода сообщения о взятии отпуска.

Требования к производным классам:

1. Менеджер (Manager): Должен содержать дополнительные атрибуты, такие как Количество подчиненных (SubordinatesCount). Метод Work() должен быть переопределен для добавления информации о управлении командой.
2. Разработчик (Developer): Должен содержать дополнительные атрибуты, такие как Специализация (Specialization). Метод TakeVacation() должен быть переопределен для добавления информации о необходимости согласования отпуска с проектными сроками.
3. Дизайнер (Designer) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Стиль дизайна (DesignStyle). Метод Work() должен быть переопределен для добавления информации о работе над дизайн-проектами.

Вариант задания № 4

Описание задачи:

Создать базовый класс Product в C#, который будет представлять информацию о продуктах. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Product:

- **Атрибуты:** Название (Name), Цена (Price), Производитель (Manufacturer).
- **Методы:**
    - GetInfo(): метод для получения информации о продукте в виде строки.
    - Discount(): метод для применения скидки к цене продукта.
    - Display(): метод для отображения информации о продукте на экране.

Требования к производным классам:

1. Электроника (Electronics): Должен содержать дополнительные атрибуты, такие как Гарантийный срок (WarrantyPeriod). Метод Discount() должен быть переопределен для добавления логики учета гарантийного срока при применении скидки.
2. Одежда (Clothing): Должен содержать дополнительные атрибуты, такие как Размер (Size). Метод Display() должен быть переопределен для добавления информации о размере при отображении информации о продукте.
3. Книги (Books) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Автор (Author). Метод GetInfo() должен быть переопределен для включения информации об авторе в описании продукта.

Вариант задания № 5

Описание задачи:

Создать базовый класс Book в C#, который будет представлять информацию о книгах. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Book:

- Атрибуты: Название (Title), Автор (Author), Год издания (YearOfPublication).
- Методы:
    - GetInfo(): метод для получения информации о книге в виде строки.
    - Read(): метод для вывода сообщения о чтении книги.
    - Borrow(): метод для вывода сообщения о выдаче книги на чтение.

Требования к производным классам:

1. Учебник (Textbook): Должен содержать дополнительные атрибуты, такие как Предмет (Subject). Метод Read() должен быть переопределен для добавления информации о предмете при чтении учебника.
2. Художественная литература (Fiction): Должен содержать дополнительные атрибуты, такие как Жанр (Genre). Метод Borrow() должен быть переопределен для добавления информации о жанре при выдаче книги на чтение.
3. Научная литература (ScientificLiterature) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Область науки (FieldOfScience). Метод GetInfo() должен быть переопределен для включения информации об области науки в описании книги.

Вариант задания № 6

Описание задачи:

Создать базовый класс Movie в C#, который будет представлять информацию о фильмах. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Movie:

- **Атрибуты:** Название (Title), Режиссер (Director), Год выпуска (ReleaseYear).
- **Методы:**
  - GetInfo(): метод для получения информации о фильме в виде строки.
  - Watch(): метод для вывода сообщения о просмотре фильма.
  - Rate(): метод для оценки фильма.

Требования к производным классам:

1. Документальный фильм (Documentary): Должен содержать дополнительные атрибуты, такие как Тематика (Theme). Метод Watch() должен быть переопределен для добавления информации о тематике при просмотре документального фильма.
2. Игровой фильм (FeatureFilm): Должен содержать дополнительные атрибуты, такие как Жанр (Genre). Метод Rate() должен быть переопределен для добавления логики оценки в зависимости от жанра.
3. Анимационный фильм (AnimatedMovie) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Студия анимации (AnimationStudio). Метод GetInfo() должен быть переопределен для включения информации о студии анимации в описании фильма.

Вариант задания № 7

Описание задачи:

Создать базовый класс BankAccount в C#, который будет представлять информацию об учетных записях в банке. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации **полиморфизма.**

Требования к базовому классу BankAccount:

- **Атрибуты:** Номер счета (AccountNumber), Баланс (Balance), Тип счета (AccountType).
- **Методы:**
    - GetInfo(): метод для получения информации о счете в виде строки.
    - Deposit(decimal amount): метод для внесения денег на счет.
    - Withdraw(decimal amount): метод для снятия денег со счета.

Требования к производным классам:

1. Сберегательный счет (SavingsAccount): Должен содержать дополнительные атрибуты, такие как Процентная ставка (InterestRate). Метод Deposit() должен быть переопределен для добавления процентов к сумме вклада при внесении денег на счет.
2. Текущий счет (CheckingAccount): Должен содержать дополнительные атрибуты, такие как Лимит овердрафта (OverdraftLimit). Метод Withdraw() должен быть переопределен для проверки и применения лимита овердрафта при снятии денег со счета.
3. Инвестиционный счет (InvestmentAccount) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Список активов (AssetsList). Метод GetInfo() должен быть переопределен для включения информации о списках активов в описании счета.

Вариант задания № 8

Описание задачи:

Создать базовый класс CreditCard в C#, который будет представлять информацию о кредитных картах. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу CreditCard:

- **Атрибуты:** Номер карты (CardNumber), Холдера (HolderName), Срок действия (ExpiryDate).
- **Методы:**
    - GetInfo(): метод для получения информации о кредитной карте в виде строки.
    - Pay(): метод для оплаты покупки с использованием карты.
    - CheckBalance(): метод для проверки баланса на карте.

Требования к производным классам:

1. GoldCard (GoldCreditCard): Должен содержать дополнительные атрибуты, такие как Бесплатные бонусные мили (BonusMiles). Метод Pay() должен быть переопределен для добавления информации о получении бонусных миль при оплате покупки.
2. PremiumCard (PremiumCreditCard): Должен содержать дополнительные атрибуты, такие как Ассистент поддержки (SupportAssistant). Метод CheckBalance() должен быть переопределен для предоставления возможности обратиться за помощью к ассистенту поддержки в случае проблем с балансом.
3. CorporateCard (CorporateCreditCard) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Компания (Company). Метод GetInfo() должен быть переопределен для включения информации о компании в описании карты.

Вариант задания № 9

Описание задачи:

Создать базовый класс Order в C#, который будет представлять информацию о заказах товаров или услуг. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Order:

- **Атрибуты:** ID заказа (OrderId), Дата создания (CreationDate), Сумма заказа (TotalAmount).
- **Методы:**
  - CalculateTotal(): метод для расчета общей суммы заказа.
  - AddItem(Item item): метод для добавления элемента в заказ.
  - RemoveItem(Item item): метод для удаления элемента из заказа.

Требования к производным классам:

1. ОнлайнЗаказ (OnlineOrder): Должен содержать дополнительные атрибуты, такие как Email клиента (CustomerEmail). Метод AddItem() должен быть переопределен для добавления информации о способе доставки при добавлении элемента.
2. ФизическийЗаказ (PhysicalOrder): Должен содержать дополнительные атрибуты, такие как Адрес доставки (DeliveryAddress). Метод RemoveItem() должен быть переопределен для добавления информации о возврате товара при удалении элемента.
3. СпециализированныйЗаказ (SpecializedOrder) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Специальные условия (SpecialConditions). Метод CalculateTotal() должен быть переопределен для учета специальных условий при расчете общей суммы заказа.

Вариант задания № 10

Описание задачи:

Создать базовый класс Invoice в C#, который будет представлять информацию о фактурах за поставленные товары или оказанные услуги. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Invoice:

- **Атрибуты:** Номер фактуры (InvoiceNumber), Дата выдачи (IssueDate), Общая сумма (TotalAmount).
- **Методы:**
  -
  - CalculateTotal(): метод для расчета общей суммы по фактуре.
  - AddLine(LineItem lineItem): метод для добавления позиции в фактуру.
  - RemoveLine(LineItem lineItem): метод для удаления позиции из фактуры.

Требования к производным классам:

1. ТоварнаяФактура (GoodsInvoice): Должна содержать дополнительные атрибуты, такие как Дата поставки (SupplyDate). Метод AddLine() должен быть переопределен для добавления информации о дате поставки товара при добавлении позиции.
2. УслуговаяФактура (ServiceInvoice): Должна содержать дополнительные атрибуты, такие как Дата оказания услуги (ServiceDate). Метод RemoveLine() должен быть переопределен для добавления информации о причине аннулирования услуги при удалении позиции.
3. КомбинированнаяФактура (CombinedInvoice) (если требуется третий класс): Должна содержать дополнительные атрибуты, такие как Наличие возврата (ReturnAllowed). Метод CalculateTotal() должен быть переопределен для учета возможного возврата товара или услуги при расчете общей суммы.

Вариант задания № 11

Описание задачи:

Создать базовый класс Customer в C#, который будет представлять информацию о клиентах или покупателях. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Customer:

- **Атрибуты:** Идентификатор клиента (CustomerId), Имя (Name), Электронная почта (Email).
- **Методы:**
  -
  - GetFullName(): метод для получения полного имени клиента.
  - UpdateEmail(string newEmail): метод для обновления электронной почты клиента.
  - ViewProfile(): метод для просмотра профиля клиента.

Требования к производным классам:

1. VIPКлиент (VipCustomer): Должен содержать дополнительные атрибуты, такие как Баланс лояльности (LoyaltyPoints). Метод ViewProfile() должен быть переопределен для отображения дополнительной информации о VIP-клиенте.
2. ОбычныйКлиент (RegularCustomer): Должен содержать дополнительные атрибуты, такие как Дата регистрации (RegistrationDate). Метод UpdateEmail() должен быть переопределен для добавления информации о дате последнего обновления электронной почты.
3. ГрупповойКлиент (GroupCustomer) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Название группы (GroupName). Метод GetFullName() должен быть переопределен для отображения названия группы вместо имени клиента.

Вариант задания № 12

Описание задачи:

Создать базовый класс Item в C#, который будет представлять информацию о товарах, которые могут быть заказаны или возвращены. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Item:

- **Атрибуты:** ID товара (ItemId), Название (Name), Цена (Price).
- **Методы:**
    -
    - GetDetails(): метод для получения детальной информации о товаре.
    - CalculateDiscount(): метод для расчета скидки на товар.
    - ApplyDiscount(decimal discount): метод для применения скидки к цене товара.

Требования к производным классам:

1. ЕдиничныйТовар (SingleItem): Должен содержать дополнительные атрибуты, такие как Единица измерения (UnitMeasure). Метод GetDetails() должен быть переопределен для добавления информации о единице измерения товара.
2. ПакетныйТовар (PackageItem): Должен содержать дополнительные атрибуты, такие как Количество единиц в пакете (QuantityPerPackage). Метод CalculateDiscount() должен быть переопределен для учета количества единиц в пакете при расчете скидки.
3. СпециальныйТовар (SpecialItem) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Дата истечения скидки (DiscountExpirationDate). Метод ApplyDiscount() должен быть переопределен для добавления информации о сроке действия скидки.

Вариант задания № 13

Описание задачи:

Создать базовый класс Inventory в C#, который будет представлять информацию о наличии товаров на складе. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Inventory:

- **Атрибуты:** ID склада (WarehouseId), Название склада (WarehouseName), Общий объем хранения (StorageCapacity).
- **Методы:**
  -
  - GetStorageStatus(): метод для получения статуса доступного пространства на складе.
  - AddItem(Item item): метод для добавления товара на склад.
  - RemoveItem(Item item): метод для удаления товара со склада.

Требования к производным классам:

1. ПерсональныйСклад (PersonalInventory): Должен содержать дополнительные атрибуты, такие как Владелец склада (OwnerName). Метод GetStorageStatus() должен быть переопределен для отображения информации о владельце склада вместе с статусом хранения.
2. ГрупповойСклад (GroupInventory): Должен содержать дополнительные атрибуты, такие как Группа товаров (ProductGroup). Метод AddItem() должен быть переопределен для добавления информации о группе товаров при добавлении нового товара.
3. АвтоматизированныйСклад (AutomatedInventory) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Автоматизация уровня (AutomationLevel). Метод RemoveItem() должен быть переопределен для добавления информации о уровне автоматизации при удалении товара.

Вариант задания № 14

Описание задачи:

Создать базовый класс Supplier в C#, который будет представлять информацию о поставщиках товаров или услуг. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Supplier:

- **Атрибуты:** ID поставщика (SupplierId), Название компании (CompanyName), Тип продукции (ProductType).
- **Методы:**
  - GetCompanyInfo(): метод для получения информации о компании.
  - ProvideQuote(): метод для предоставления котировки на товары или услуги.
  - SubmitOrder(): метод для отправки заказа поставщику.

Требования к производным классам:

1. Производитель (Manufacturer): Должен содержать дополнительные атрибуты, такие как Год основания (FoundedYear). Метод ProvideQuote() должен быть переопределен для включения информации о годе основания компании в котировку.
2. Ритейлер (Retailer): Должен содержать дополнительные атрибуты, такие как Расположение магазина (StoreLocation). Метод SubmitOrder() должен быть переопределен для добавления информации о расположении магазина при отправке заказа.
3. Импортер (Importer) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Страна происхождения товара (OriginCountry). Метод GetCompanyInfo() должен быть переопределен для отображения страны происхождения товара вместе с остальной информацией о компании.

Вариант задания № 15

Описание задачи:

Создать базовый класс OrderLine в C#, который будет представлять информацию о строке заказа, содержащей детали одного товара в заказе. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу OrderLine:

- **Атрибуты:** ID товара (ProductId), Название товара (ProductName), Цена товара (Price).
- **Методы:**
  -
  - CalculateTotal(): метод для расчета общей стоимости строки заказа.
  - UpdatePrice(decimal newPrice): метод для обновления цены товара в строке заказа.
  - GetProductDetails(): метод для получения деталей товара.

Требования к производным классам:

1. СтандартнаяСтрока (StandardLine): Должна содержать дополнительные атрибуты, такие как Количество единиц (Units). Метод CalculateTotal() должен быть переопределен для учета количества единиц при расчете общей стоимости.
2. СпециальнаяСтрока (SpecialLine): Должна содержать дополнительные атрибуты, такие как Скидка (Discount). Метод UpdatePrice() должен быть переопределен для применения скидки к цене товара.
3. БесплатнаяСтрока (FreeLine) (если требуется третий класс): Должна содержать дополнительные атрибуты, такие как Предварительный платеж (Prepayment). Метод CalculateTotal() должен быть переопределен для учета предварительного плата при расчете общей стоимости.

Вариант задания № 16

Описание задачи:

Создать базовый класс PaymentMethod в C#, который будет представлять различные способы оплаты. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу PaymentMethod:

- **Атрибуты:** ID способа оплаты (PaymentMethodId), Название способа оплаты (MethodName), Минимальная сумма (MinAmount).
- **Методы:**
  - ProcessPayment(decimal amount): метод для обработки платежа указанной суммы.
  - CheckMinimumAmount(decimal amount): метод для проверки минимальной суммы платежа.
  - GetPaymentDetails(): метод для получения деталей способа оплаты.

Требования к производным классам:

1. ОнлайнОплата (OnlinePayment): Должен содержать дополнительные атрибуты, такие как URL платежной системы (PaymentUrl). Метод ProcessPayment() должен быть переопределен для включения URL платежной системы в процесс оплаты.
2. БанковскийПеревод (BankTransfer): Должен содержать дополнительные атрибуты, такие как Банковские данные (BankData). Метод CheckMinimumAmount() должен быть переопределен для проверки минимальной суммы платежа с учетом банковских комиссий.
3. Наличные (CashPayment) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Место выдачи наличных (CashPickupPoint). Метод GetPaymentDetails() должен быть переопределен для отображения места выдачи наличных.

Вариант задания № 17

Описание задачи:

Создать базовый класс ShippingOption в C#, который будет представлять различные опции доставки. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу ShippingOption:

- **Атрибуты:** ID опции доставки (DeliveryOptionId), Название опции доставки (DeliveryOptionName), Стоимость доставки (Cost).
- **Методы:**
  - CalculateCost(): метод для расчета стоимости доставки.
  - EstimateDeliveryTime(): метод для оценки времени доставки.
  - GetDeliveryDetails(): метод для получения деталей опции доставки.

Требования к производным классам:

1. СтандартнаяДоставка (StandardDelivery): Должна содержать дополнительные атрибуты, такие как Среднее время доставки (AverageDeliveryTime). Метод EstimateDeliveryTime() должен быть переопределен для предоставления среднего времени доставки.
2. ЭкспрессДоставка (ExpressDelivery): Должна содержать дополнительные атрибуты, такие как Минимальное время доставки (MinDeliveryTime). Метод CalculateCost() должен быть переопределен для увеличения стоимости доставки в случае необходимости ускорения доставки.
3. Самовывоз (Pickup) (если требуется третий класс): Должна содержать дополнительные атрибуты, такие как Адрес пункта самовывоза (PickupAddress). Метод GetDeliveryDetails() должен быть переопределен для отображения адреса пункта самовывоза.

Вариант задания № 18

Описание задачи:

Создать базовый класс Review в C#, который будет представлять отзывы о продуктах или услугах. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Review:

- **Атрибуты:** ID отзыва (ReviewId), Текст отзыва (Text), Рейтинг (Rating).
- **Методы:**
    - DisplayReview(): метод для отображения отзыва.
    - RateProduct(): метод для присвоения рейтинга продукту.
    - GetReviewDetails(): метод для получения деталей отзыва.

Требования к производным классам:

1. ОтзывОбслуживания (ServiceReview): Должен содержать дополнительные атрибуты, такие как Дата посещения (VisitDate). Метод DisplayReview() должен быть переопределен для включения даты посещения в отображение отзыва.
2. ОтзывТовара (ProductReview): Должен содержать дополнительные атрибуты, такие как Идентификатор продукта (ProductId). Метод RateProduct() должен быть переопределен для связывания рейтинга с конкретным продуктом.
3. ОтзывУслуги (ServiceReview) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Время начала услуги (StartTime). Метод GetReviewDetails() должен быть переопределен для отображения времени начала услуги вместе с другими деталями отзыва.

Вариант задания № 19

Описание задачи:

Создать базовый класс Subscription в C#, который будет представлять подписки на различные услуги. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Subscription:

- **Атрибуты:** ID подписки (SubscriptionId), Название услуги (ServiceName), Стоимость подписки (Cost).
- **Методы:**
    - CalculateMonthlyCost(): метод для расчета ежемесячной стоимости подписки.
    - ExtendSubscription(): метод для продления подписки на дополнительный период.
    - GetSubscriptionDetails(): метод для получения деталей подписки.

Требования к производным классам:

1. ПодпискаНаОнлайнСервис (OnlineServiceSubscription): Должна содержать дополнительные атрибуты, такие как Количество доступных пользователей (MaxUsers). Метод CalculateMonthlyCost() должен быть переопределен для учета количества пользователей при расчете стоимости.
2. ПодпискаНаСтреаминг (StreamingSubscription): Должна содержать дополнительные атрибуты, такие как Количество одновременных потоков (MaxStreams). Метод ExtendSubscription() должен быть переопределен для добавления специальных предложений для продления подписки.
3. ПодпискаНаВидео(VideoSubscription) (если требуется третий класс): Должна содержать дополнительные атрибуты, такие как Качество видео (VideoQuality). Метод GetSubscriptionDetails() должен быть переопределен для отображения качества видео вместе с другими деталями подписки.

Вариант задания № 20

Описание задачи:

Создать базовый класс Event в C#, который будет представлять организованные мероприятия. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Event:

- **Атрибуты:** ID мероприятия (EventId), Название мероприятия (EventName), Дата проведения (Date).
- **Методы:**
  - RegisterParticipant(): метод для регистрации участника.
  - NotifyParticipants(): метод для уведомления участников о мероприятии.
  - GetEventDetails(): метод для получения деталей мероприятия.

Требования к производным классам:

1. Конференция (Conference): Должна содержать дополнительные атрибуты, такие как Количество спикеров (SpeakersCount). Метод NotifyParticipants() должен быть переопределен для отправки уведомлений с программой конференции.
2. Фестиваль (Festival): Должна содержать дополнительные атрибуты, такие как Количество музыкальных групп (BandsCount). Метод RegisterParticipant() должен быть переопределен для регистрации участников с выбором интересующих музыкальных жанров.
3. Семинар (Seminar) (если требуется третий класс): Должна содержать дополнительные атрибуты, такие как Тематические области (Topics). Метод GetEventDetails() должен быть переопределен для отображения тематических областей семинара вместе с другими деталями мероприятия.

Вариант задания № 21

Описание задачи:

Создать базовый класс Ticket в C#, который будет представлять билеты на различные мероприятия. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Ticket:

- **Атрибуты:** ID билета (TicketId), Номер ряда (RowNumber), Номер места (SeatNumber).
- **Методы:**
  - PrintTicketInfo(): метод для печати информации о билете.
  - ReserveSeat(): метод для резервирования места.
  - GetTicketDetails(): метод для получения деталей билета.

Требования к производным классам:

1. БилетНаКонцерт (ConcertTicket): Должен содержать дополнительные атрибуты, такие как Имя исполнителя (ArtistName). Метод PrintTicketInfo() должен быть переопределен для включения имени исполнителя в информацию о билете.
2. БилетНаСпектакль (PlayTicket): Должен содержать дополнительные атрибуты, такие как Название спектакля (PlayTitle). Метод GetTicketDetails() должен быть переопределен для отображения названия спектакля вместе с другими деталями билета.
3. БилетНаКонференцию (ConferenceTicket) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Название конференции (ConferenceName). Метод ReserveSeat() должен быть переопределен для добавления информации о предпочтениях участника относительно мест.

Вариант задания № 22

Описание задачи:

Создать базовый класс Project в C#, который будет представлять проекты в рамках организации. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Project:

- **Атрибуты:** ID проекта (ProjectId), Название проекта (ProjectName), Статус проекта (Status).
- **Методы:**
    - UpdateStatus(): метод для обновления статуса проекта.
    - GetProjectDetails(): метод для получения деталей проекта.
    - AssignTeamMember(): метод для назначения командного члена проекту.

Требования к производным классам:

1. ПродуктовыйПроект (ProductProject): Должен содержать дополнительные атрибуты, такие как Ожидаемая дата завершения (ExpectedCompletionDate). Метод UpdateStatus() должен быть переопределен для включения ожидаемой даты завершения в статус проекта.
2. ИсследовательскийПроект (ResearchProject): Должен содержать дополнительные атрибуты, такие как Финансирование (FundingAmount). Метод AssignTeamMember() должен быть переопределен для указания специфических требований к навыкам командного члена для исследовательского проекта.
3. ИнфраструктурныйПроект (InfrastructureProject) (если требуется третий класс): Должен содержать дополнительные атрибуты, такие как Срок выполнения (ExecutionPeriod). Метод GetProjectDetails() должен быть переопределен для отображения срока выполнения проекта вместе с другими деталями проекта.

Вариант задания № 23

Описание задачи:

Создать базовый класс Task в C#, который будет представлять задачи внутри проекта. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Task:

- **Атрибуты:** ID задачи (TaskId), Название задачи (TaskName), Приоритет задачи (Priority).
- **Методы:**
  - MarkAsComplete(): метод для отметки задачи как выполненной.
  - GetTaskDetails(): метод для получения деталей задачи.
  - ReassignTo(): метод для переназначения задачи другому члену команды.

Требования к производным классам:

1. ДелегатскаяЗадача (DelegateTask): Должна содержать дополнительные атрибуты, такие как Дата выполнения (DueDate). Метод MarkAsComplete() должен быть переопределен для включения даты выполнения в сообщение о завершении задачи.
2. КомандныйЗадача (TeamTask): Должна содержать дополнительные атрибуты, такие как Команда (TeamName). Метод ReassignTo() должен быть переопределен для указания нового члена команды, которому будет переназначена задача.
3. ИсследовательскаяЗадача (ResearchTask) (если требуется третий класс): Должна содержать дополнительные атрибуты, такие как Исходные данные (DataSource). Метод GetTaskDetails() должен быть переопределен для отображения источников данных, используемых в задаче, вместе с другими деталями задачи.

Вариант задания № 24

Описание задачи:

Создать базовый класс Notification в C#, который будет представлять уведомления пользователям. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу Notification:

- **Атрибуты:** ID уведомления (NotificationId), Текст уведомления (MessageText), Тип уведомления (Type).
- **Методы:**
    - DisplayNotification(): метод для отображения уведомления пользователю.
    - SendNotification(): метод для отправки уведомления.
    - GetNotificationDetails(): метод для получения деталей уведомления.

Требования к производным классам:

1. EmailУведомление (EmailNotification): Должно содержать дополнительные атрибуты, такие как Адрес электронной почты (EmailAddress). Метод SendNotification() должен быть переопределен для отправки уведомления по электронной почте.
2. SMSУведомление (SMSNotification): Должно содержать дополнительные атрибуты, такие как Номер телефона (PhoneNumber). Метод SendNotification() должен быть переопределен для отправки уведомления через SMS.
3. PushУведомление (PushNotification) (если требуется третий класс): Должно содержать дополнительные атрибуты, такие как Платформа (Platform, например, iOS или Android). Метод DisplayNotification() должен быть переопределен для отображения уведомления на мобильной платформе.

Вариант задания № 25

Описание задачи:

Создать базовый класс SavingsAccount в C#, который будет представлять специализированные учетные записи для сбережений в банке. На основе этого класса разработать 2-3 производных класса, демонстрирующих принципы наследования и полиморфизма. В каждом из классов должны быть реализованы новые атрибуты и методы, а также переопределены некоторые методы базового класса для демонстрации полиморфизма.

Требования к базовому классу SavingsAccount:

- **Атрибуты:** ID счета (AccountId), Баланс (Balance), Процентная ставка (InterestRate).
- **Методы:**
  - Deposit(): метод для внесения денег на счет.
  - Withdraw(): метод для снятия денег со счета.
  - CalculateInterest(): метод для расчета процентов по счету.

Требования к производным классам:

1. СтуденческаяУчетнаяЗапись (StudentAccount): Должна содержать дополнительные атрибуты, такие как Год обучения (YearOfStudy). Метод CalculateInterest() должен быть переопределен для применения сниженной процентной ставки для студентов.
2. ПремиумУчетнаяЗапись (PremiumAccount): Должна содержать дополнительные атрибуты, такие как Минимальный баланс (MinimumBalance). Метод Withdraw() должен быть переопределен для ограничения снятия средств до минимального баланса.
3. ИнвестиционныйУчет (InvestmentAccount) (если требуется третий класс): Должна содержать дополнительные атрибуты, такие как Инвестиционный портфель (PortfolioValue). Метод Deposit() должен быть переопределен для автоматического инвестирования части внесенных средств.