

## Наследование

- Во всех последних классах иерархии следует переопределить метод *Equals*, чтобы обеспечить сравнение значений, а не ссылок, и переопределить метод *ToString*.
- Функция Main должна содержать *массив из элементов базового класса, заполненный ссылками на производные классы*.

### Условия задач по вариантам

#### Вариант 1.

##### 1. Смоделировать структуру предприятия:

Классы	Свойства
Фирма	название (get, set)
Отдел	название (get, set) количество сотрудников (get, set)
Сотрудник	фио (get, set) должность (get, set) оклад (get, set)

Предположим, что все Сотрудники делятся на два типа:

- Штатный сотрудник
- Сотрудник по контракту

Требуется производить расчет заработной платы:

- Для Штатного сотрудника может быть установлена премия, тогда зар.плата = оклад + премия
- Для Сотрудника по контракту зар.плата = оклад

для этого выполнить следующие действия:

- сделать класс Сотрудник абстрактным (abstract)
- добавить в класс Сотрудник абстрактный метод «рассчитать зарплату»
- создать классы «Штатный сотрудник» и «Контрактник», унаследовать их от класса Сотрудник
- в класс Штатный сотрудник добавить свойство «премия» (set)
- перегрузить метод «рассчитать зарплату» в классах «Штатный сотрудник» и «Контрактник»

В один из отделов добавить двух сотрудников по контракту и двух штатных сотрудников. Назначить премии штатным сотрудникам. Используя цикл for рассчитать и вывести на экран зарплату по всем сотрудникам данного отдела.

2. Создать открытые (Public) конструкторы для классов Фирма, Отдел, Сотрудник, Штатный сотрудник, Сотрудник по контракту. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе. Конструкторы дочерних классов (Штатный сотрудник и Сотрудник по контракту) должны вызывать конструктор базового класса (Сотрудник). В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

#### Вариант 2.

##### 1. Смоделировать структуру банка:

Классы	Свойства
Банк	название (get, set)
Филиал	название (get, set) общая сумма вкладов (get, set)
Вклад	фиовкладчика (get, set)

	сумма вклада (get, set)
--	-------------------------

Предположим, что все Вклады делятся на два типа:

- Долгосрочный - начисления процентов по вкладу зависят от суммы (до 10000 – 12%, больше 10000 – 18% годовых)
- До востребования (6% годовых независимо от суммы вклада)

Предполагаем, что проценты по вкладу начисляются каждый месяц. Требуется рассчитывать сумму вклада, по прошествии указанного количества месяцев.

для этого выполнить следующие действия:

- сделать класс Вклад абстрактным (abstract)
- добавить в класс Вклад абстрактный метод «рассчитать сумму вклада» с параметром «количество месяцев»
- создать классы «Долгосрочный вклад» и «Вклад до востребования», унаследовать их от класса Вклад
- перегрузить метод «рассчитать сумму вклада (количество месяцев)» в классах «Долгосрочный вклад» и «Вклад до востребования».

В определенный филиал добавить два долгосрочных вклада и три вклада до востребования. Используя цикл for рассчитать и вывести на экран первоначальную сумму и сумму по прошествии трех месяцев каждого вклада данного филиала.

2. Создать открытые (Public) конструкторы для классов Банк, Филиал, Вклад, Долгосрочный вклад, Вклад до востребования. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе. Конструкторы дочерних классов (Долгосрочный вклад и Вклад до востребования) должны вызывать конструктор базового класса (Вклад)

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

### Вариант 3.

#### 1. Смоделировать структуру аэропорта:

Классы	Свойства
Аэропорт	название (get, set)
Летательный аппарат	название (get, set) макс. количество пассажиров (get, set)
Пассажир	фио (get, set) № посадочного места (get, set)

Предположим, что все Летательные аппараты делятся на два типа:

- Самолет
- Вертолет

Требуется реализовать процедуру взлета летательного аппарата, реализованную по разному для Самолета и Вертолета. При этом Самолет также отличается от Вертолета наличием свойства «длина полосы разгона».

Процедура выводит в текстовом виде примерно следующее:

- для Самолета: «<название л/а>. Разогреваю двигатели. Прохожу полосу разгона длиной <длина полосы разгона>. Взлетаю. Убираю шасси.»
- для Вертолета «<название л/а>. Разогреваю двигатели. Взлетаю.»

для этого выполнить следующие действия:

- сделать класс Летательный аппарат абстрактным (abstract)
- добавить в класс Летательный аппарат абстрактный метод «взлет»
- создать классы «Самолет» и «Вертолет», унаследовать их от класса Летательный аппарат
- добавить в класс Самолет свойство «длина полосы разгона» (get, set)
- перегрузить метод «взлет» в классах «Самолет» и «Вертолет».

В определенный аэропорт добавить два самолета и один вертолет. Указать длину полосы разгона для самолетов. Используя цикл for вызвать метод «взлет» у каждого летательного аппарата аэропорта и вывести результаты на экран.

**2.** Создать открытые (Public) конструкторы для классов Аэропорт, Летательный аппарат, Самолет, Вертолет, Пассажир. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.

Конструкторы дочерних классов (Самолет и Вертолет) должны вызывать конструктор базового класса (Летательный аппарат).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

#### Вариант 4.

##### 1. Смоделировать структуру библиотеки:

Классы	Свойства
Библиотека	название (get, set)
Отдел (по жанрам)	название жанра (get, set) количество изданий (get, set)
Издание	название (get, set) автор (get, set) год издания (get, set)

Предположим, что все Издания делятся на два типа:

- Книга – отличается наличием резюме (краткого описания)
- Журнал – отличается наличием множества статей

Требуется формировать текстовое описание Издания в виде:

- для Книги: «Автор: <автор>,  
год издания: <год издания>  
название: <название>  
резюме: <резюме>»
- для Журнала: «Автор: <автор>,  
год издания: <год издания>  
название: <название>  
статьи: <название статьи 1>, <название статьи 2> ... »

для этого выполнить следующие действия:

- сделать класс Издание абстрактным (abstract)
- добавить в класс Издание абстрактный метод «сформировать описание»
- создать классы «Книга» и «Журнал», унаследовать их от класса Издание
- добавить в класс Журнал множественное свойство «названия статей» (get, add)
- перегрузить метод «сформировать описание» в классах «Книга» и «Журнал».

В определенный отдел библиотеки добавить две книги и один журнал. В журнал добавить две статьи. Используя цикл `for`, сформировать и вывести на экран описание каждого издания данного отдела.

2. Создать открытые (Public) конструкторы для классов Библиотека, Отдел, Издание, Книга, Журнал. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.  
Конструкторы дочерних классов (Книга и Журнал) должны вызывать конструктор базового класса (Издание).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

#### Вариант 5.

1. Смоделировать структуру компании сотовой связи:

Классы	Свойства
Компания	Название (get, set)
Тариф	название (get, set) количество абонентов (get, set)
Абонент	фио (get, set) номертелефона (get, set) остаток на счете (get, set)

Предположим, что все Тарифы делятся на два типа:

- Посекундный. Отличается наличием свойства «стоимость секунды разговора» (get, set). Оплата производится за каждую секунду разговора.
- Поминутный. Отличается наличием свойства «стоимость минуты разговора» (get, set). Оплата производится за каждую минуту разговора, т.е., например, 20 сек. приравняется 1 мин.

Требуется производить подсчет стоимости разговора указанной длительности (в секундах).

для этого выполнить следующие действия:

- сделать класс Тариф абстрактным (abstract)
- добавить в класс Тариф абстрактный метод «рассчитать стоимость разговора» с параметром «количество секунд»
- создать классы «Посекундный тариф» и «Поминутный тариф», унаследовать их от класса Тариф
- в класс Посекундный тариф добавить свойство «стоимость секунды разговора» (get, set)
- в класс Поминутный тариф добавить свойство «стоимость минуты разговора» (get, set)
- перегрузить метод «рассчитать стоимость разговора (количество секунд)» в классах «Посекундный тариф» и «Поминутный тариф».

В определенную компанию добавить один поминутный тариф и один посекундный тариф. Указать стоимость минуты и секунды разговора соответственно. Используя цикл `for`, рассчитать и вывести на экран стоимость разговора продолжительностью 80 секунд для каждого тарифа данной компании.

2. Создать открытые (Public) конструкторы для классов Компания, Тариф, Посекундный тариф, Поминутный тариф, Абонент. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.  
Конструкторы дочерних классов (Посекундный тариф и Поминутный тариф) должны вызывать конструктор базового класса (Тариф).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

#### Вариант 6.

1. Смоделировать структуру автосалона:

Классы	Свойства
Автосалон	название (get, set)
Автомобиль (марка)	название марки (get, set) макс. количество пассажиров (get, set) стоимость (get, set) количество на складе (get, set) boolean наличие (get, set)
Заявка на покупку	фио покупателя (get, set) номер телефона (get, set)

Предположим, что все Заявки делятся на два типа:

- Заявки на приобретение со стенда.
- Заявки на отложенную поставку. Отличается наличием свойства «процент скидки» (get, set), действующей при приобретении автомобиля с отложенной поставкой.

Требуется производить подсчет стоимости заказа:

- Заявка на приобретение со стенда. Стоимость заказа = стоимость автомобиля, указанного в заявке.
- Заявка на отложенную поставку. Стоимость заказа = стоимость автомобиля, указанного в заявке \* (100% - процент скидки).

для этого выполнить следующие действия:

- сделать класс Заявка абстрактным (abstract)
- добавить в класс Заявка абстрактный метод «рассчитать стоимость заказа»
- создать классы «Заявка на приобретение со стенда» и «Заявка на отложенную поставку», унаследовать их от класса Заявка
- в класс Заявка на отложенную поставку добавить свойство «процент скидки» (get, set)
- перегрузить метод «рассчитать стоимость заказа» в классах «Заявка на приобретение со стенда» и «Заявка на отложенную поставку».

На определенный автомобиль оформить две заявки на приобретение со стенда, и одну заявку на отложенную поставку. Для заявки на отложенную поставку указать значение процента скидки. Используя цикл for, рассчитать и вывести на экран стоимость заказа по всем заявкам на данный автомобиль.

2. Создать открытые (Public) конструкторы для классов Автосалон, Автомобиль, Заявка на покупку, Заявка на приобретение со стенда, Заявка на отложенную поставку. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.

Конструкторы дочерних классов (Заявка на приобретение со стенда и Заявка на отложенную поставку) должны вызывать конструктор базового класса (Заявка на покупку).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

## Вариант 7.

1. Смоделировать структуру музыкальной коллекции:

Классы	Свойства
Коллекция	название (get, set) фио владельца (get, set)
Музыкальный носитель	автор/группа (get, set)

(альбом)	жанр (get, set) год выпуска (get, set) общая продолжительность звучания (get, set)
Музыкальное произведение	название (get, set) продолжительность (get, set)

Предположим, что все Музыкальные произведения делятся на два типа:

- Песни – отличаются наличием свойств «текст» и «автор текста»
- Инструментальные произведения – отличаются наличием множественного свойства строкового типа «инструменты-участники»

Требуется формировать текстовое описание Музыкального произведения в виде:

- для Песни:  
«Название: <название>,  
продолжительность: <продолжительность>  
текст: <текст>  
автор текста: <автор текста>»
- для Инструментального произведения:  
«Название: <название>,  
продолжительность: <продолжительность>  
инструменты-участники: <инструменты-участники>»

для этого выполнить следующие действия:

- сделать класс Музыкальное произведение абстрактным (abstract)
- добавить в класс Музыкальное произведение абстрактный метод «сформировать описание»
- создать классы «Песня» и «Инструментальное произведение», унаследовать их от класса Музыкальное произведение
- добавить в класс Песня свойства «текст» и «автор текста» (set)
- добавить в класс Инструментальное произведение свойство «инструменты-участники» (add, remove)
- перегрузить метод «сформировать описание» в классах «Песня» и «Инструментальное произведение».

В определенный альбом добавить две песни и одно инструментальное произведение. Для песен указать тексты песен и их авторов. Для инструментального произведения указать инструменты-участники. Используя цикл for, сформировать и вывести на экран описание каждого музыкального произведения данного альбома.

2. Создать открытые (Public) конструкторы для классов Коллекция, Альбом, Музыкальное произведение, Песня, Инструментальное произведение. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.

Конструкторы дочерних классов (Песня, Инструментальное произведение) должны вызывать конструктор базового класса (Музыкальное произведение).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

## Вариант 8.

1. Смоделировать структуру реестра городского жилья:

Классы	Свойства
--------	----------

Город	название (get, set)
Здание	название улицы (get, set) номер дома (get, set) общая площадь (get) базовая ежемесячная оплата за кв.м площади (get, set)
Помещение	номер (get, set) площадь (get, set)

Предположим, что все Помещения делятся на два типа:

- Квартира – отличается наличием множественного свойства строкового типа «ФИО жильцов»
- Офис – отличается наличием свойств «Название фирмы-владельца», «Вид деятельности»

Требуется рассчитывать ежемесячную оплату за помещение по следующим правилам:

- для Квартиры:

Ежемес. оплата = базовая ежемесячная оплата за кв.м площади \* площадь \* (1 + кол-во жильцов\*0,1)

- Для Офиса:

Ежемес. оплата = базовая ежемесячная оплата за кв.м площади \* площадь \* 2

для этого выполнить следующие действия:

- сделать класс Помещение абстрактным (abstract)
- добавить в класс Помещение абстрактный метод «рассчитатьЕжемесячнуюОплату»
- создать классы «Квартира» и «Офис», унаследовать их от класса Помещение
- добавить в класс Квартира свойство «ФИО жильцов» (add)
- добавить в класс Офис свойства «название фирмы-владельца» и «вид деятельности» (get, set)
- перегрузить метод «рассчитатьЕжемесячнуюОплату» в классах «Квартира» и «Офис».

В определенное здание добавить две квартиры и один офис. Для квартир указать ФИО жильцов. Для офиса указать название фирмы-владельца и вид деятельности. Используя цикл for, рассчитать и вывести на экран размер ежемесячной оплаты для каждого помещения в данном здании.

**2.** Создать открытые (Public) конструкторы для классов Город, Здание, Помещение, Квартира, Офис. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.

Конструкторы дочерних классов (Квартира, Офис) должны вызывать конструктор базового класса (Помещение).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

## Вариант 9.

### 1. Смоделировать структуру зоопарка:

Классы	Свойства
Зоопарк	название (get, set)
Вольер/клетка	номер (get, set) размер (get, set) макс. количество животных (get, set) текущее количество животных (get, set)
Животное	название (get, set)



	boolean хищник (get, set)
--	---------------------------

Предположим, что все Животные, живущие в зоопарке, делятся на три типа:

- Рыбы – отличаются наличием свойства boolean «глубоководная»
- Птицы – отличаются наличием свойства «скорость полета»
- Звери – отличается наличием свойства «среда обитания» (н-р, «лес», «степь», «пустыня» и т.п.)

Требуется формировать описание животных по следующему правилу:

- для Рыбы:  
«Класс: <глубоководная/не глубоководная> рыба,  
Название: <название>,  
хищник: <да/нет>»
- для Птицы:  
«Класс: птица,  
Название: <название>,  
скорость полета: <скорость полета>»
- для Зверя:  
«Класс: зверь,  
Название: <название>,  
Среда обитания: <среда обитания>»

для этого выполнить следующие действия:

- сделать класс Животное абстрактным (abstract)
- добавить в класс Животное абстрактный метод «сформировать описание»
- создать классы «Рыба», «Птица» и «Зверь», унаследовать их от класса Животное
- добавить в класс Рыба свойство «глубоководная» (set)
- добавить в класс Птица свойство «скорость полета» (set)
- добавить в класс Зверь свойство «среда обитания» (set)
- перегрузить метод «сформировать описание» в классах «Рыба», «Птица» и «Зверь».

В определенный зоопарк добавить одну рыбу, одну птицу и одного зверя. Для рыбы указать значение свойства «глубоководная», для птицы – «скорость полета», для зверя – «среда обитания». Используя цикл for, сформировать и вывести на экран описание каждого животного в зоопарке.

Создать открытые (Public) конструкторы для классов Зоопарк, Вольер/Клетка, Животное, Рыба, Птица, Зверь. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.

Конструкторы дочерних классов (Рыба, Птица, Зверь) должны вызывать конструктор базового класса (Животное).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.

#### Вариант 10.

##### 1. Смоделировать структуру автоматизированного банкомата:

Классы	Свойства
Банк	название (get, set)
Счет	номер (get, set) PIN-код (get, set) остаток (get, set)
Банкомат	идентификационный номер (get, set)



Предположим, что все Счета делятся на два типа:

- обычный
- льготный

Требуется реализовать метод «снять со счета(сумма)» так, чтобы при снятии определенной суммы с Обычного счета дополнительно взималось 5% от указанной суммы, а при снятии с Льготного счета – взимался всего 1%.

для этого выполнить следующие действия:

- сделать класс Счет абстрактным (abstract)
- в классе Счет сделать абстрактным метод «снять со счета» с параметром «сумма»
- создать классы «Обычный счет» и «Льготный счет», унаследовать их от класса Счет
- перегрузить метод «снять со счета(сумма)» в классах «Обычный счет» и «Льготный счет».

В определенный банк добавить два обычных счета и два льготных счета и задать значения свойств. Используя цикл for, вывести на экран остаток по каждому счету данного банка, затем снять с каждого счета определенную сумму и снова вывести остаток по каждому счету.

2. Создать открытые (Public) конструкторы для классов Банк, Счет, Обычный счет, Льготный счет, Банкомат. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.

Конструкторы дочерних классов (Обычный счет и Льготный счет) должны вызывать конструктор базового класса (Счет).

В основной программе (main) при создании объектов задавать значения свойств. Проверить работу программы.