

КОНСТРУКТОРЫ

Конструктор – это специальный метод, который всегда присутствует в классе и вызывается при создании объекта данного класса.

Если конструктор не переопределяется в классе явным образом, он все равно присутствует в классе. Синтаксис конструктора по умолчанию следующий:

```
class Tree {  
  
    // Конструктор по умолчанию  
    public Tree() {}  
  
    ...  
}
```

Обратим внимание, что в отличие от обычного метода, конструктор не имеет указания типа возвращаемого значения (т.к. он всегда возвращает новый объект данного класса) и его имя всегда совпадает с именем класса.

Вызывается конструктор при создании объекта данного класса, т.е. в нашем примере при вызове операции:

```
new Tree();
```

Программист может переопределить конструктор данного класса, используя его для осуществления некоторых действий при создании объектов, например:

```
class Tree {  
  
    // Конструктор по умолчанию  
    public Tree() {  
        System.out.println("Создано новое дерево");  
    }  
    ...  
}
```

Теперь при каждом вызове операции `new Tree()` будет происходить вывод на экран сообщения «Создано новое дерево».

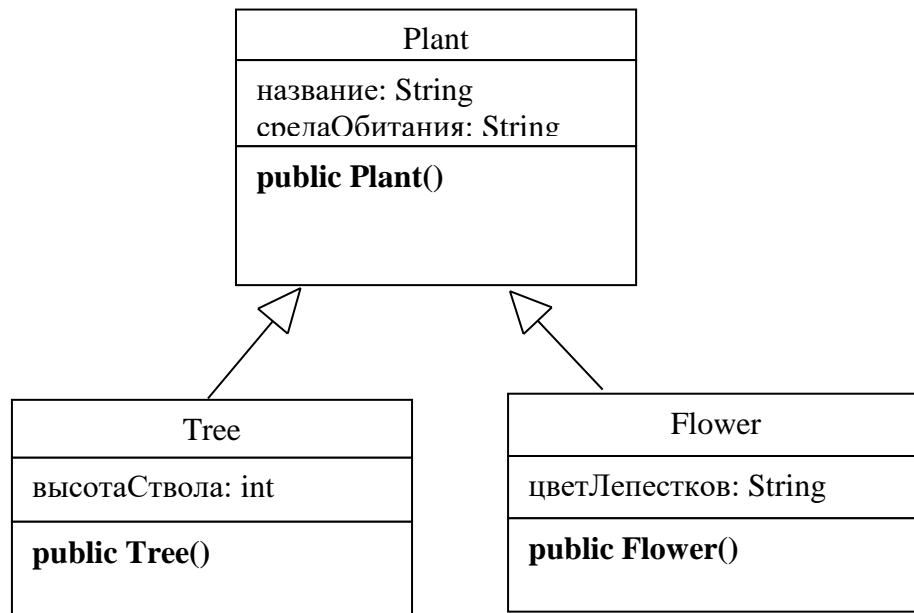
Конструкторы могут содержать параметры, и, благодаря этому, часто используются для задания начальных значений свойств при создании объекта, например:

```
class Tree {  
  
    String nazvanie;  
  
    // Конструктор по умолчанию  
    public Tree(String nazvanie) {  
        this.nazvanie = nazvanie;  
    }  
  
    ...  
}
```

Теперь при создании объекта класса `Tree` необходимо указывать значение параметра `nazvanie`.

```
new Tree("Сосна");
```

Особенности использования конструкторов при наследовании



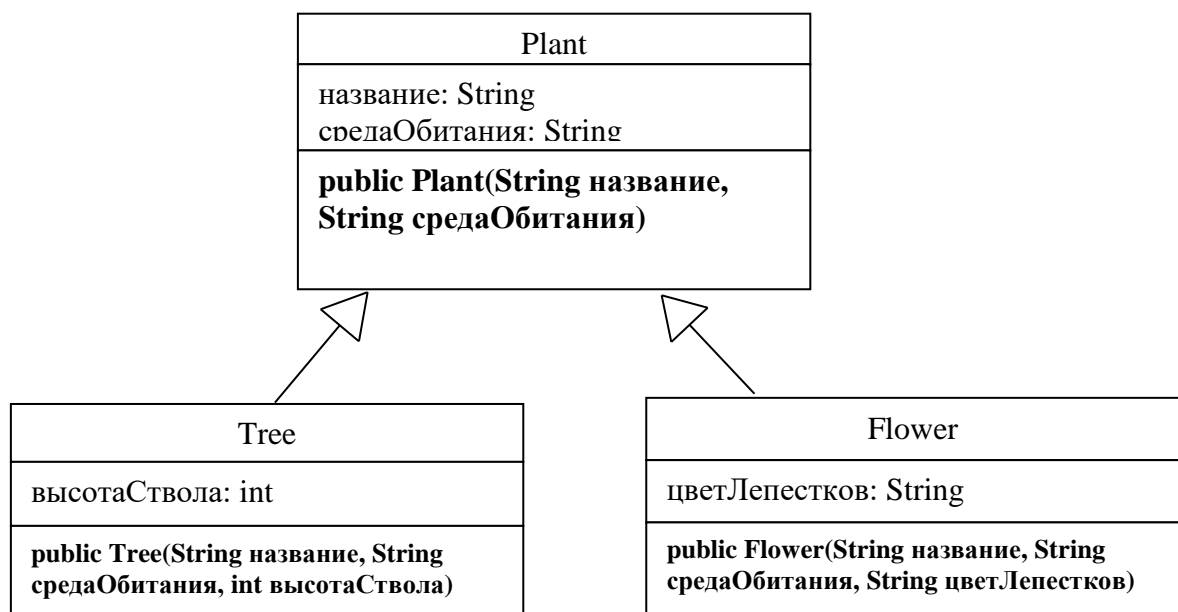
На рисунке приведен пример иерархии классов, где классы **Tree** и **Flower** являются наследниками класса **Plant** (некоторые из свойств и методов, приведенные в предыдущих лб., не указаны для упрощения). Жирным шрифтом выделены конструкторы по умолчанию, присутствующие в каждом из приведенных классов. Рассмотрим ситуацию создания объекта класса **Tree**:

```
new Tree();
```

Казалось бы, при создании данного объекта сразу должен быть вызван конструктор `public Tree()` класса **Tree**. На самом деле, ситуа: **при создании объекта определенного класса, сначала вызывается конструктор базового класса. Этот шаг повторяется рекурсивно, начиная с «самого базового» класса (корень иерархии), затем следующий за ним класс и т.д. пока не будет выполнен конструктор данного класса.**

В нашем примере, согласно приведенной схеме, при создании объекта класса **Tree**, сначала будет вызван конструктор `public Plant()` класса **Plant**, и лишь затем конструктор `public Tree()` класса **Tree**.

Далее, рассмотрим ситуацию, если конструкторы классов принимают какие-либо параметры:



Как указывалось выше, при создании объекта класса Tree происходит вызов конструктора класса Plant. Однако в данном примере, конструктор public Plant (String название, String средаОбитания) требует указать значения параметров «название» и «средаОбитания» и, следовательно, не может быть вызван неявно, автоматически. В данном случае программист должен сам позаботиться о вызове конструктора базового класса в конструкторе дочернего класса. Вызов конструктора базового класса выполняется с помощью команды **super(...)**, которая должна быть указана **первой** командой конструктора.

Таким образом, конструктор класса Tree, будет выглядеть следующим образом:

```
class Tree {  
  
    int высотаСтвола;  
  
    public Tree(String название, String средаОбитания, int высотаСтвола) {  
  
        // выполняем вызов конструктора базового класса Plant с передачей  
        // значений параметров  
        super (название, средаОбитания) ;  
  
        this. высотаСтвола = высотаСтвола;  
        ...  
    }  
  
    ...  
}
```

ВАРИАНТЫ ЗАДАНИЙ

Вариант № 1

Смоделировать структуру предприятия:

Классы	Свойства и методы
Фирма	название (get, set)
Отдел	название (get, set) количество сотрудников (get, set)
Сотрудник	фио (get, set) должность (get, set) оклад (get, set) рассчитать зарплату()
Штатный сотрудник	премия (set) рассчитать зарплату()
Сотрудник по контракту	рассчитать зарплату()

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

Смоделировать структуру банка:

Классы	Свойства
Банк	название (get, set)
Филиал	название (get, set) общая сумма вкладов (get, set)
Вклад	фио вкладчика (get, set) сумма вклада (get, set) рассчитать сумму вклада (количество месяцев)
Долгосрчный вклад	рассчитать сумму вклада (количество месяцев)
Вклад до востребования	рассчитать сумму вклада (количество месяцев)

Создать открытые (Public) конструкторы для классов Банк, Филиал, Вклад, Долгосрчный вклад, Вклад до востребования. Конструкторы должны принимать в качестве параметров значения простых (не множественных) свойств, описанных в соответствующем классе.

Конструкторы дочерних классов (Долгосрчный 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

Смоделировать структуру аэропорта:

Классы	Свойства
Аэропорт	название (get, set)
Летательный аппарат	название (get, set) макс. количество пассажиров (get, set) взлет()
Самолет	длина полосы разгона (get, set) взлет()
Вертолет	взлет()
Пассажир	фио (get, set) №посадочного места (get, set)

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

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) сформировать описание()
Книга	резюме (get, set) сформировать описание()
Журнал	статьи (get, add) сформировать описание()

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

Смоделировать структуру компании сотовой связи:

Классы	Свойства
Компания	Название (get, set)
Тариф	название (get, set) количество абонентов (get, set) рассчитать стоимость разговора (количество секунд)
Посекундный тариф	стоимость секунды разговора» (get, set) рассчитать стоимость разговора (количество секунд)
Поминутный тариф	стоимость минуты разговора» (get, set) рассчитать стоимость разговора (количество секунд)
Абонент	фio (get, set) номер телефона (get, set) остаток на счете (get, set)

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

Смоделировать структуру автосалона:

Классы	Свойства
Автосалон	название (get, set)
Автомобиль	марка (get, set) макс. количество пассажиров (get, set) стоимость (get, set) количество на складе (get, set) boolean наличие (get, set)
Заявка на покупку	фio покупателя (get, set) номер телефона (get, set) автомобиль (get, set) рассчитать стоимость заказа()
Заявка на приобретение со стенда	рассчитать стоимость заказа()
Заявка на отложенную поставку	процент скидки (get, set) рассчитать стоимость заказа()

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.

Смоделировать структуру музыкальной коллекции:

Классы	Свойства
Коллекция	название (get, set) фио владельца (get, set)
Альбом	автор/группа (get, set) жанр (get, set) год выпуска (get, set) общая продолжительность звучания (get, set)
Музыкальное произведение	название (get, set) продолжительность (get, set) сформировать описание()
Песня	текст (get, set) автор текста (get, set) сформировать описание()
Инструментальное произведение	инструменты (get, set) сформировать описание()

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

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) рассчитать ежемес. оплату()
Квартира	ФИО жильцов (get, add, remove) рассчитать ежемес. оплату()
Офис	название фирмы-владельца (get, set) вид деятельности (get, set) рассчитать ежемес. оплату()

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

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) сформировать описание()
Рыба	глубоководная (get, set) сформировать описание()
Птица	скорость полета (get, set) сформировать описание()
Зверь	среда обитания (get, set) сформировать описание()

а) 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

Смоделировать структуру автоматизированного банкомата:

Классы	Свойства
Банк	название (get, set)
Счет	номер (get, set) PIN-код (get, set) остаток (get, set) снять со счета (сумма)
Обычный счет	снять со счета (сумма)
Льготный счет	снять со счета (сумма)
Банкомат	идентификационный номер (get, set) адрес (get, set)

а) 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.