

## Список 03 – Лабораторная практика

### Наследование и отношения между классы

---

*Предмет: Алгоритмизация и программирование*

*Преподаватель: Хольгер Эспинола Ривера*

**1. Простая иерархия банковских счетов.** В финансовом учреждении ведутся два типа банковских счетов: 1- сберегательный счет; 2- бизнес-счет, где существует иерархическая связь между классом **Account** (базовый класс) и классами **SavingsAccount** и **BusinessAccount** (производные классы). Проанализируйте классы:

#### А) Класс **Account**

- id: строка длиной максимум 5 символов
- имя: строка длиной не более 50 символов
- ставка: процент от установленного законом налога, вычитаемого со счета
- баланс: ежемесячный доход в долларах, который поступает на карту
- реализовать методы-конструкторы, сеттеры, геттеры и виртуальную функцию для вывода данных учетной записи

#### В) Класс **SavingsAccount**

- одиночное наследование всех атрибутов и методов класса Account
- плата за обслуживание: фиксированная долларовая стоимость обслуживания карты
- Реализуйте конструкторы с отношениями наследования, сеттеры, геттеры и переопределенные функции для печати данных сберегательных счетов.

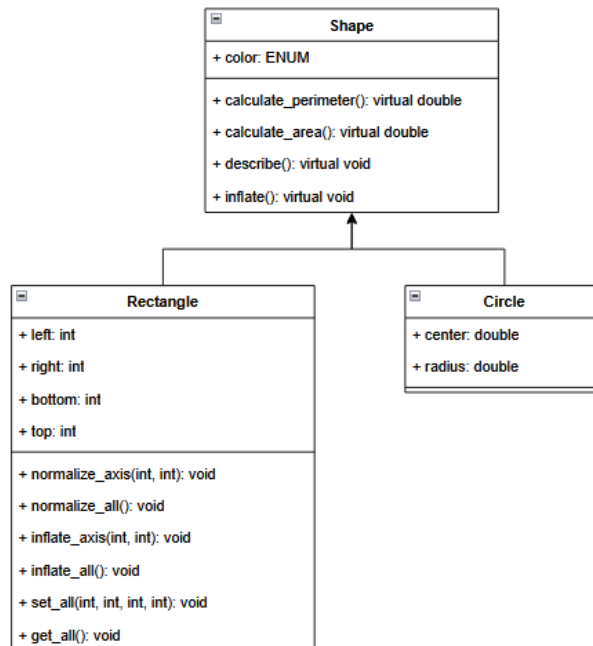
#### С) Класс **BusinessAccount**

- одиночное наследование всех атрибутов и методов класса Account
- Коэффициент дивидендов: процент от доходности карты (от чистой стоимости после уплаты налогов)
- Реализуйте конструкторы с отношениями наследования, сеттеры, геттеры и переопределенные функции для печати данных бизнес-счетов.

[1]. Создайте диаграмму UML, установив связи между классами, атрибутами и методами.

[2]. Реализуйте все функции, определенные в каждом из классов. Определите функцию чистой суммы, которая вычисляет чистую стоимость счета после расчета ставок и дивидендов в соответствии с типом счета.

**2. Геометрические фигуры. Простое наследование.** Рассматриваются три класса: Shape, Rectangle и Circle, где класс Shape является базовым классом, а классы Rectangle и Circle являются производными классами (имеется отношение наследования). Проверьте диаграмму UML:



[1]. Создайте иерархию классов с классом **Shape** в качестве базового класса и классами **Rectangle** и **Circle** в качестве производных классов. Определите атрибуты, методы и конструкторы для каждого класса.

[2]. Класс **Shape** имеет атрибут цвета, который определяется типом данных ENUM, содержащим 3 возможные категории: красный, зеленый и синий. В базовом классе объявлены четыре виртуальные функции: **calculate\_perimeter()**, **calculate\_area()**, **describe()** и **inflate()**, допускающие полиморфизм. Кроме того, класс имеет виртуальный деструктор. Кроме того, класс имеет виртуальный деструктор. Функция **describe()** должна вывести имя базового класса и значение атрибутов. В случае функции **calculate\_perimeter()**, **calculate\_area()** и **inflate()** она не должна выполнять никаких операций, являясь фактически чистой функцией.

[3]. Класс **Rectangle** сохраняет атрибуты и методы, реализованные в упражнении 03 (лабораторная работа 01). Внесите изменения в класс, чтобы унаследовать методы, определенные базовым классом Shape, но переопределить эти методы.

- Функция **calculate\_perimeter()** должна использовать формулу  $2 \times (\text{длина} + \text{ширина})$ ;

- Функция **calculate\_area()** должна использовать формулу  $\text{длина} \cdot \text{ширина}$ ;
- Функция **describe()** должна вывести класс и значения каждого из 4 атрибутов (длина, ширина, вниз, верх);
- Функция **inflate()** должна использовать входной параметр и отправлять аргумент в каждую из координат прямоугольника.

[4]. Класс **Circle** имеет атрибуты центра и радиуса. Необходимо реализовать конструктор на основе параметров для класса **Circle**, который копирует объекты по умолчанию. В случае, когда конструктор является объектом класса **Rectangle**, то он должен построить окружность, описанную вокруг указанного прямоугольника, используя ее измерения для вычисления координат центра и радиуса. Для этого центр описанной около прямоугольника окружности помещается в центр прямоугольника, а диагональ прямоугольника определяется как диаметр описанной около прямоугольника окружности. Необходимы адаптации, чтобы класс **Circle** наследовал методы, определенные базовым классом **Shape**, но переопределял эти методы.

- Функция **calculate\_perimeter()** должна использовать формулу:

$$perimeter = 2 \cdot \pi \cdot r$$

- Функция **calculate\_area()** должна использовать формулу:

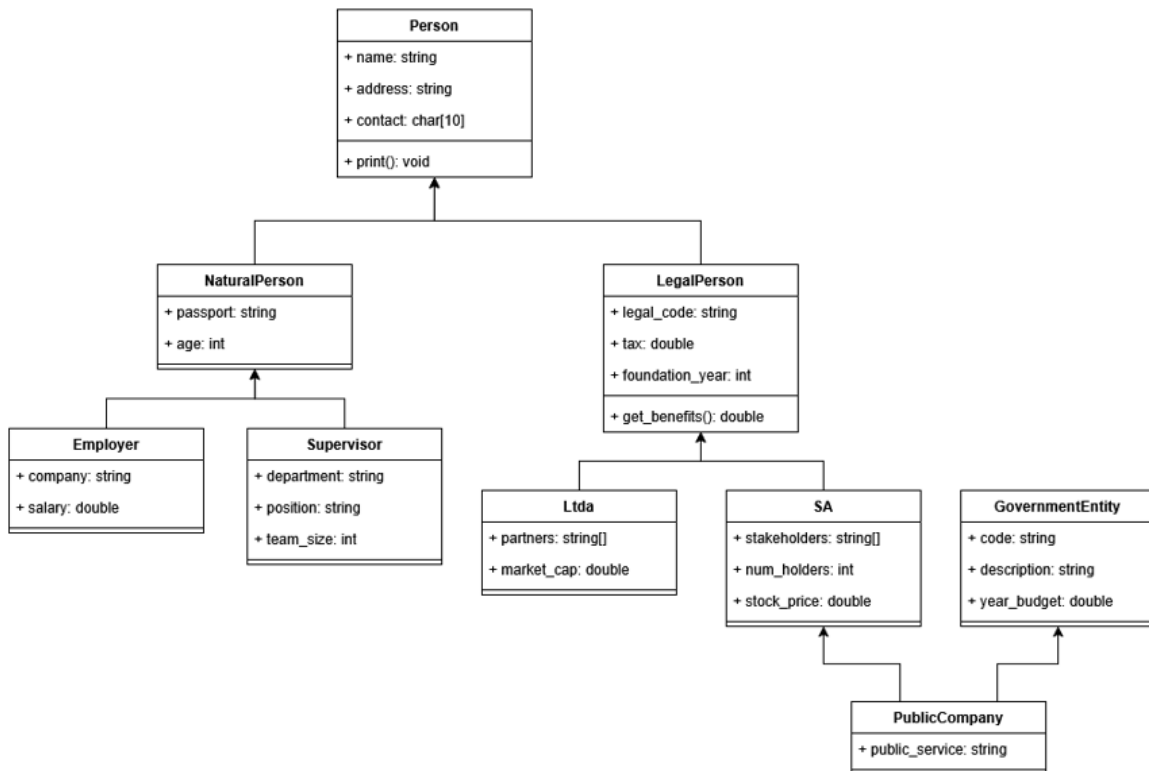
$$area = \pi \cdot r^2$$

- Функция **describe()** должна вывести класс и значения двух атрибутов круга.
- Функция **inflate()** должна увеличивать количество единиц, указанных в качестве входного параметра для радиуса.

[5]. Поэкспериментируйте с виртуальными методами, инициализируя объекты базового класса и производного класса. Получить описание каждого инициализированного объекта.

[6]. Создает объекты базового класса и передает объекты производных классов как ссылки на эти объекты. Смотрите описания, периметр и площадь этих объектов. Что случилось? Изменили ли эти объекты природу своего класса?

**3. Многоуровневое и многократное наследование.** Компания хочет создать систему переписи для управления информацией о лицах и организациях. В данном исследовании рассматриваются следующие типы сущностей, определяемые диаграммой UML:



1) Class **Person** (Человек): определяется именем, адресом и телефонным номером;

2) class **Natural Person** (Физическое лицо): представитель физического лица. Определяется паспортом и возрастом. Физическое лицо может быть двух типов:

- **Employer** (Сотрудников): лицо, доходы которого зависят от компании и которое находится в подчинении. Определяется компанией и зарплатой.

- **Supervisor** (Руководитель): лицо, которое также подчиняется какой-либо компании, но имеет более высокую иерархию в компании. Определяется отделом, должностью и размером команды, которой руководит руководитель.

3) class **Legal Person** (Юридическое лицо): представляет юридическое лицо. Определяется некоторым правовым кодексом, налоговой ставкой и годом основания. Юридическое лицо может быть двух типов:

- **Ltda**: представляет общество с ограниченной ответственностью. Определяется вектором имен партнеров и рыночной капитализацией (в миллионах долларов)

- **SA**: представляет анонимное общество - корпорацию. Определяется вектором имен заинтересованных лиц, количеством держателей и ценой на фондовом рынке.

4) class **Government Entity** (Государственная организация): представляет собой организацию, связанную с правительством. Определяется

государственным кодом, описанием государственной организации и объемом годового бюджета.

5) class **Public Company** (Публичная компания): представляет собой компании, которые имеют публичное предложение для продажи. Определяется названием государственной услуги, предлагаемой компанией.

Учитывая описание системы переписи и диаграмму UML, выполните следующие процессы:

[1]. Реализуйте конструкторы по умолчанию, на основе параметров и конструкторы копирования объектов, учитывая следующие отношения наследования:

- **Person** (базовый класс) ==> **Natural Person** и **Legal Person** (производные классы);

- **Natural Person** (базовый класс) ==> **Employer** - Работодатель и **Supervisor** -Руководитель (производные классы);

- **Legal Person** - Юридическое лицо (базовый класс) ==> **Ltda** и **SA** (производные классы);

- **SA** и **Government Entity** (базовые классы) => **Public Company** (производный класс)

[2]. Реализовать функции **setters**, **getters** и **print()**.

[3]. Используя виртуальные методы, рассчитайте прибыль класса **Legal Person** (юридического лица) в зависимости от типа компании и ее атрибутов.

[4]. Проанализируйте разницу между многоуровневым наследованием и множественным наследованием. Определите, какие классы связаны с каким типом наследования.