Список 03 – Лабораторная практика

Наследование и отношения между классы

Предмед: 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:

A) 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

B) 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.

C) Класс **BusinessAccount**

- одиночное наследование всех атрибутов и методов класса Account

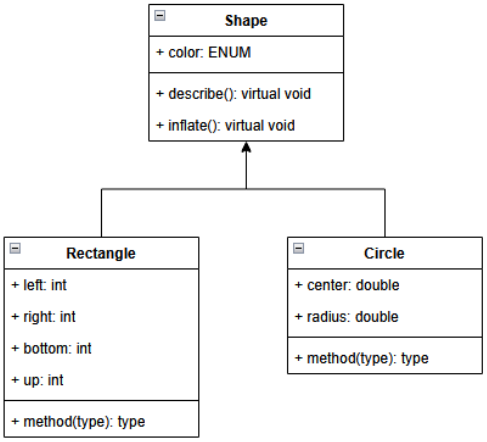
- Коэффициент дивидендов: процент от доходности карты (от чистой стоимости после уплаты налогов)

- 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 две виртуальные функции: **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 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 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 center и radius. Необходимо реализовать конструктор класса Circle, основанный на параметрах и копирующий объекты по умолчанию, при этом разрешая использование конструктора класса Rectangle, использующего аргументы объектов. Адаптации необходимы для того, чтобы класс наследовал методы, определенные базовым классом Shape, но переопределял эти методы. Функция describe()

должна вывести класс и значения двух атрибутов круга. Функция inflate() должна увеличивать количество единиц, определенное как входной параметр для радиуса.. На основе прямоугольника рассчитываются атрибуты круга: $radio = \frac{x_2 - x_1}{2}$; $center = x_1 + radio$

[5]. Поэкспериментируйте с виртуальными классами, инициализируя объекты базового класса и производных классов. Получить описание для каждого инициализированного объекта.

[6]. Создайте объекты базового класса и передайте объекты производных классов в качестве ссылок на эти объекты. Проверьте описания этих объектов. Что случилось? Изменили ли эти объекты природу своего класса?

[7]. Чисто виртуальные функции. Ввести метод void inflate() в базовый класс. Подумайте: возможно ли реализовать такой метод для базового класса? => как его объявить. Реализовать этот метод для производных классов. Подсказка: использовать указатель базового класса и отправлять ссылки на объекты из производных классов.