**Лабораторная работа №2: Классы. Объектное моделирование**

**Контрольные вопросы**

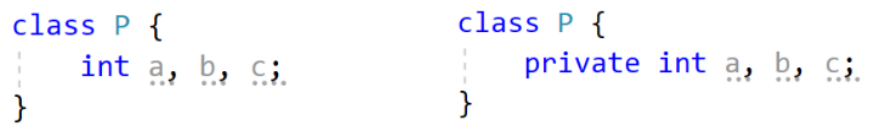
*1. Приведите определение понятия «класс».*

Класс – это абстрактное описание некоторого объекта, которое определяет характеристики и поведение этого самого объекта.

*2. Что такое структура? Чем структура отличается от класса?*

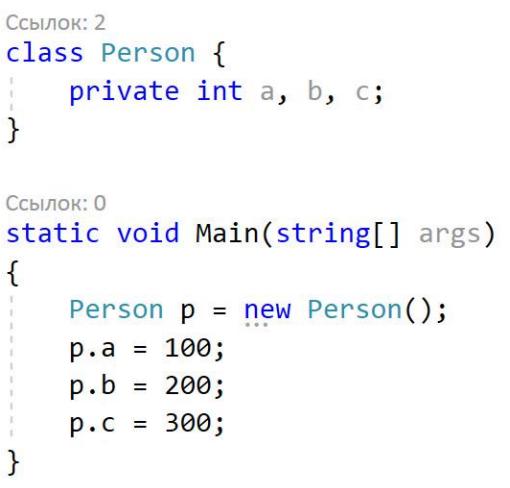
Структура – это тип данных, который представляет собой объединение различных типов данных. Структура отличается от класса меньшим функционалом в программе (отсутствуют некоторые свойства атрибутов, которые есть в классах).

*3. Чем отличаются определения классов?*



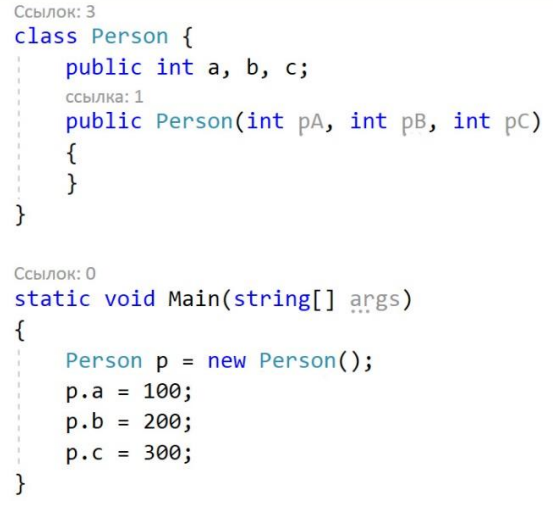
Отличия заключаются в доступности параметров: в первом случае параметры a, b и c доступны вне объявления класса (то есть их можно использовать в других классах или в основной части программы), а во втором случае не предусмотрена возможность работы этих параметров с другими классами, так как они объявлены как private параметры.

*4. Опишите ошибки компиляции (если есть) в представленном фрагменте:*



Данный код не будет скомпилирован, так как происходит попытка обращения к приватным параметрам класса Person вне самого класса, что противоречит логике работы приватных параметров.

*5. Опишите ошибки компиляции (если есть) в представленном фрагменте:*



В функции Main происходит попытка создания экземпляра класса Person без передачи параметров функции-конструктору (функция внутри класса, которая называется также как и сам класс), хотя в классе Person определен конструктор с параметрами.

*6. Как называется переменная типа класс?*

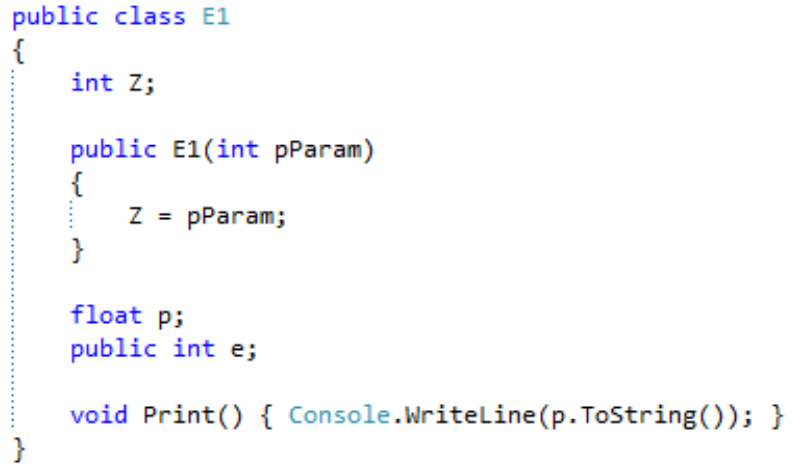
Переменная типа класса называется объектом. То есть, когда мы создаем переменную, которая ссылается на созданный ранее класс, то программа распознаёт эту переменную как объект класса.

*7. С использованием какого зарезервированного слова происходит объявление класса? Какое зарезервированное слово используется для объявления структуры?*

Для объявления класса в C# используется зарезервированное слово class.

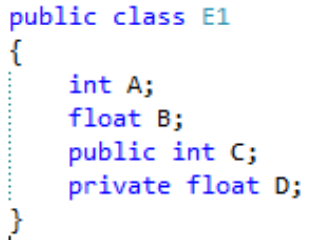
Для объявления структуры используется слово struct.

*8. Опишите ошибки компиляции (если есть) в представленном фрагменте:*



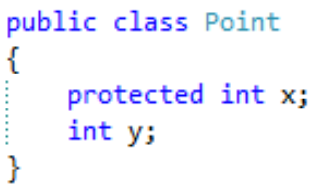
Данный код будет скомпилирован без ошибок.

*9. Какие члены класса являются закрытыми?*



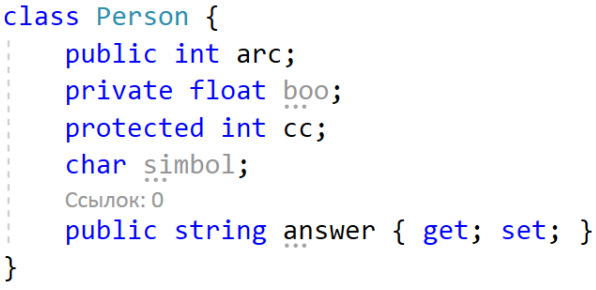
Следующие члены класса закрыты: D (так как объявлен как private float D).

*10. Какие поля объявленного класса доступны из объектной переменной?*



В данном примере доступны все поля из объектной переменной. Переменная будет недоступна только в том случае, если она объявлена через private.

*11. К каким полям нельзя обратиться из объектной переменной?*



Нельзя обратиться к: boo, cc и simbol.

**Индивидуальное задание**

*Задание 1.*

Разработать класс для представления объекта «Прямоугольный параллелепипед». Реализуйте все необходимые поля данных (закрытые) и методы позволяющие:

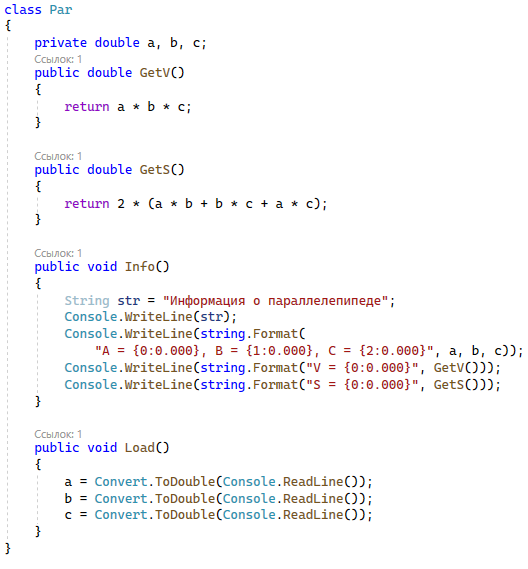
– считывать из файла состояние полей данных объекта;

– вычислять объем прямоугольного параллелепипеда;

– вычислять площадь поверхности прямоугольного параллелепипеда;

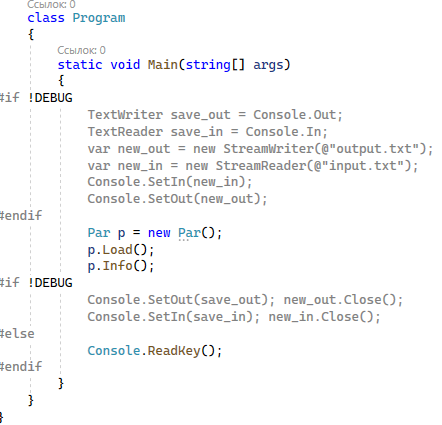
– выводить полную информацию об объекте в выходной файл.

Ниже представлен код класса «Прямоугольный параллелепипед»:



В основной части программы присутствуют различия работы с данными в Release и Debug версиях: в Debug запись и вывод данных производится в консоли, а в Release версии данные на входе и выходе программы поступают через файлы «input.txt» и «output.txt», соответственно.

Ниже представлен код Main функции программы:



*Задание 2.*

Спроектируйте класс, наполните его требуемой функциональностью, продемонстрируйте работоспособность класса. Реализуйте две версии: debug и release, продемонстрируйте различное поведение программы для различных конфигураций сборки.

Вариант 1: Класс «Шар». Реализовать ввод и вывод полей данных, вычисление объема, диаметра и площади поверхности, а также вывод информации об объекте.

Код класса «Sphere» (шар) реализован следующим образом: единственным атрибутом для вычисления площади и объёма фигуры, который может вводить пользователь в программу, является радиус шара. Радиус представлен в виде переменной R типа данных double. Параметр R также является закрытым, то есть использовать его вне класса нельзя.

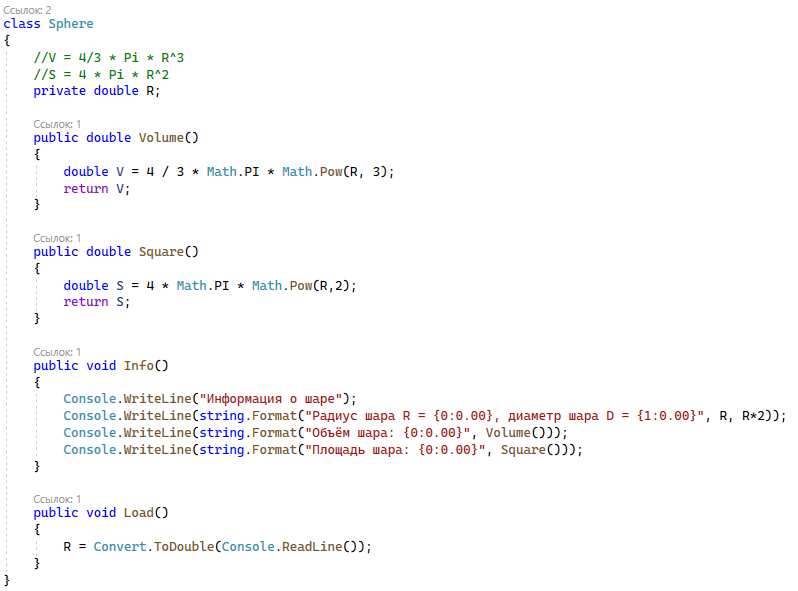
Функция Volume() – находит объём шара по формуле V=4/3\*Pi\*R^3, где Pi – математическая константа, которую можно получить из библиотеки Math (Math.PI). Для возведения в степень используется функция Math.Pow().

Функция Square() – находит площадь шара по формуле S = 4\*Pi\*R^2, где Pi также математическая константа.

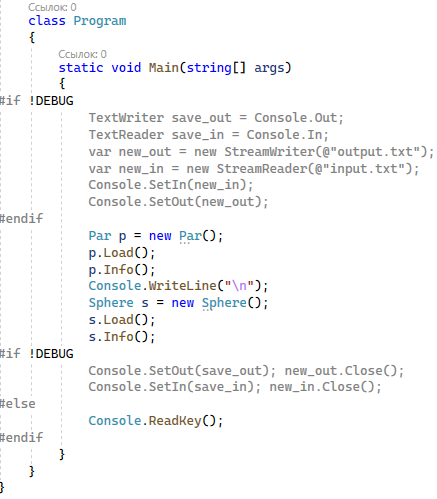
Функция Info() – выводит информацию о классе. Информация представлена в виде строк, в которых записано значение радиуса, диаметра, объёма и площади шара.

Функция Load() считывает значение для переменной R. Эта функция в основной части программы вызывается в первую очередь.

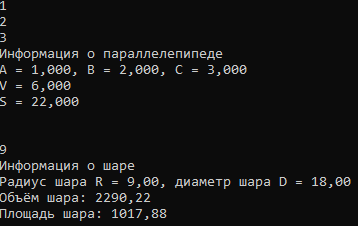
Код получившегося класса представлен ниже:



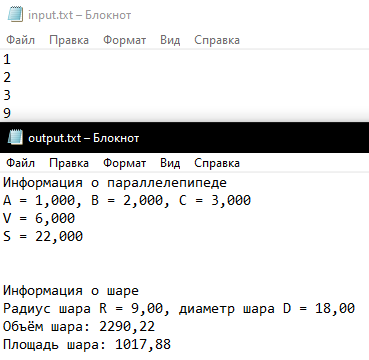
Основная часть программы была также изменена, а именно было добавлено создание объекта класса «Шар» с загрузкой в него значения радиуса и вывода основной информации о нём:



Работа программы в консоли продемонстрирована ниже:



Работа программы через файлы представлена ниже (используются такие же входные значения):



*Код лабораторной работы доступен на Github:* https://github.com/DaryaAristova/KIS\_Labs/tree/main/LabKIS2