**Лабораторная работа №13. Часть 1**

*«Классы. Интерфейсы»*

**Рекомендации по программированию**

* Поля предпочтительнее делать private.
* Интерфейс должен быть полным, т.е. предоставлять возможность выполнять любые разумные действия с классом. И одновременно минимально необходимым, т.е. без дублирования и пересечения возможностей методов.
* Каждый метод класса должен решать только одну задачу. Не стоит объединять два коротких независимых фрагмента кода в один метод.
* Если действия встречаются хотя бы дважды, стоит оформить их в отдельные функции.
* Все заданные методы, поля и экземпляры классов должны иметь осмысленные имена.
* Каждое имя интерфейса должно начинаться с буквы I.

**Требования к лабораторной**

* Каждый разрабатываемый класс должен, как правило, содержать следующие элементы: поля с различными спецификаторами, конструкторы с параметрами и без параметров, методы, свойства.
* Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса.
* При возникновении ошибок должны выбрасываться исключения
* К каждому отчету, помимо формулировки задания и скрина консоли, нужно прикрепить еще диаграмму классов

**Постановка задачи**

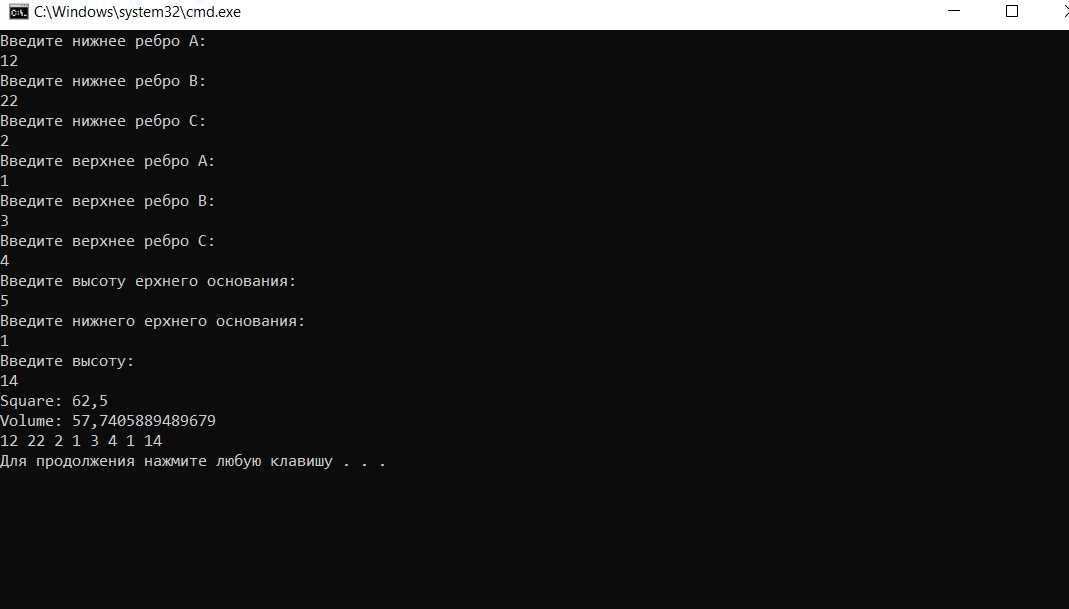
**Задача 1:** Задан некоторый класс согласно индивидуальному заданию. Реализовать для данного класса конструктор с параметрами и без. Предусмотреть ввод данных пользователем с клавиатуры. Создать для полученного класса интерфейс «Площадь», «Периметр» (длину окружности считать за периметр или можно создать отдельный интерфейс) и дополнительные («Объем», если ваша фигура трехмерная, и для поиска сторон, диагоналей и др.), внутри которого определены соответствующие методы.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Класс** | **Методы** |
| 1. | Усеченная треугольная пирамида | Площади и периметры оснований, объем, площадь боковой поверхности |
| 2. | Треугольная пирамида | Площадь и периметр основания, площадь боковой поверхности, объем |
| 3. | Четырёхугольная пирамида | Площадь и периметр основания, площадь боковой поверхности, объем |
| 4. | Конус | Площадь и длину окружности основания, площадь боковой поверхности, объем |
| 5. | Ромб | Периметр, площадь, длины диагоналей |
| 6. | Усеченный конус | Площадь, длины окружности оснований, площадь боковой поверхности, объем |
| 7. | Цилиндр | Площадь, длины окружности оснований, площадь боковой поверхности, объем |
| 8. | Круг | Длину окружности, площадь, длину хорды через центральный угол и радиус |
| 9. | Квадрат | Периметр, площадь, диагональ, радиус окружности вписанной в квадрат, радиус окружности описанной около квадрата |
| 10. | Прямоугольник | Периметр, площадь, длину диагонали, окружность, описанная вокруг прямоугольника, радиус окружности, вписанная в прямоугольник |
| 11. | Разносторонний треугольник | Периметр площадь, радиус описанной вокруг треугольника окружности, радиус окружности, вписанной в треугольник, длины средних линий для всех сторон |
| 12. | Параллелепипед | Площадь, объем, диагональ, площадь поверхности |
| 13. | Куб | Площадь, периметр, площадь поверхности, диагональ |
| 14. | Четырехугольник | Площадь, периметр, радиус вписанной и описанной окружности |
| 15. | Трапеция | Площадь, периметр, среднюю линию, длины диагоналей |
| 16. | Равнобедренный треугольник | Площадь, периметр, радиус описанной вокруг треугольника окружности, радиус окружности, вписанной в треугольник, длины средних линий для всех сторон |
| 17. | Прямоугольный треугольник | Площадь, периметр, гипотенузу, радиус описанной вокруг треугольника окружности, радиус окружности, вписанной в треугольник, длины средних линий для всех сторон |
| 18. | Прямоугольная пирамида | Площадь и периметр основания, площадь боковой поверхности, объем |
| 19. | Равносторонний треугольник | Площадь, периметр, радиус описанной вокруг треугольника окружности, радиус окружности, вписанной в треугольник, длины средних линий для всех сторон |
| 20. | Параллелограмм | Периметр, площадь, длины диагоналей, окружность, описанная вокруг параллелограмма, радиус окружности, вписанной в параллелограмм |

**Задача 2:** Реализовать иерархию классов Х->Y. В классе Х присутствуют данные x1, x2. В классе Y – данное у. Определить конструкторы с параметрами и без параметров для обоих классов. Создать интерфейс для ввода-вывода данных. В производном классе присутствует метод, определяющий действия из индивидуального задания.

|  |  |
| --- | --- |
| **Вариант** | **Метод** |
| 1. | Сумму цифр всех данных класса |
| 2. | Произведение цифр всех данных класса |
| 3. | Сумму квадратов всех данных класса |
| 4. | Значение х1+х2-у |
| 5. | Значение (х1+х2)/у |
| 6. | Значение (х1+х2)\*у |
| 7. | Значение х1\*у+х2 |
| 8. | Значение х1+х2\*у |
| 9. | Значение х1\*х2+у |
| 10. | Значение х1\*х2/у |
| 11. | Значение х1\*х2-у |
| 12. | Значение х1/у-х2\*у |
| 13. | Значение х1-х2\*(cos(у)) |
| 14. | Значение х2+tg(у\*х1) |
| 15. | Значение х1+х2+ctg(у) |
| 16. | Значение y\*5\*sin(x1\*х2) |
| 17. | Значение х1+cos(х2\*у) |
| 18. | Значение х1-х2/sin(y) |
| 19. | Значение x1+3+ tg(у\*х2) |
| 20. | Значение х1\*х2-sin(y) |

**Задача 3:** Для задания номер 2 реализовать множественное наследование интерфейсов, т.е. для выражения (а+b)\*с написать интерфейс ISum и IMult, которые будут наследовать друг друга, а реализовываться уже в одном классе Х с полями х1, х2, у. Здесь же присутствует метод для расчета данных. Создать интерфейс для ввода-вывода данных.



using lab13.interfaces;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13

{

class Piramide : IParams, ISumNum, IShowInfo

{

int rebroNijA;

int rebroNijB;

int rebroNijC;

int rebroVerhA;

int rebroVerhC;

int rebroVerhD;

int hVerhOsn;

int hNijOsn;

int h;

public Piramide(int rebroNijA, int rebroNijB, int rebroNijC, int rebroVerhA, int rebroVerhC, int rebroVerhD, int hVerhOsn, int hNijOsn, int h)

{

this.rebroNijA = rebroNijA;

this.rebroNijB = rebroNijB;

this.rebroNijC = rebroNijC;

this.rebroVerhA = rebroVerhA;

this.rebroVerhC = rebroVerhC;

this.rebroVerhD = rebroVerhD;

this.hVerhOsn = hVerhOsn;

this.hNijOsn = hNijOsn;

this.h = h;

}

public int RebroNijA { get => rebroNijA; set => rebroNijA = value; }

public int RebroNijB { get => rebroNijB; set => rebroNijB = value; }

public int RebroNijC { get => rebroNijC; set => rebroNijC = value; }

public int RebroVerhA { get => rebroVerhA; set => rebroVerhA = value; }

public int RebroVerhC { get => rebroVerhC; set => rebroVerhC = value; }

public int RebroVerhD { get => rebroVerhD; set => rebroVerhD = value; }

public int HVerhOsn { get => hVerhOsn; set => hVerhOsn = value; }

public int HNijOsn { get => hNijOsn; set => hNijOsn = value; }

public int H { get => h; set => h = value; }

public int PerimetrOsn(int rebroA, int rebroB, int rebroC)

{

return rebroA + rebroB + rebroC;

}

public void ShowInfo()

{

Console.WriteLine(RebroNijA + " " + rebroNijB + " " + RebroNijC + " " + RebroVerhA + " " + RebroVerhC + " " + rebroVerhD + " " + HNijOsn + " " + H);

}

public double Square(int hOsn,int rebro)

{

return 1 / 2.0 \* hOsn \* rebro;

}

public int SumNum(int num)

{

int sum = 0;

for (; num!=0; num/=10)

{

sum += num % 10;

}

return sum;

}

public double Volume()

{

return 1 / 3.0 \* h \* (Square(hNijOsn, rebroNijA) + Math.Sqrt(Square(hNijOsn, rebroNijA) \* Square(hVerhOsn, rebroVerhA)) + Square(hVerhOsn, rebroVerhA));

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите нижнее ребро А: ");

int rebroNijA = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите нижнее ребро B: ");

int rebroNijB = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите нижнее ребро C: ");

int rebroNijC = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите верхнее ребро A: ");

int rebroVerhA = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите верхнее ребро B: ");

int rebroVerhB = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите верхнее ребро C: ");

int rebroVerhC = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите высоту ерхнего основания: ");

int hVerhOsn = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите нижнего ерхнего основания: ");

int hNijOsn = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите высоту: ");

int h = Convert.ToInt32(Console.ReadLine());

Piramide p = new Piramide(rebroNijA, rebroNijB, rebroNijC, rebroVerhA, rebroVerhB, rebroVerhC, hVerhOsn, hNijOsn, h);

Console.WriteLine("Square: " + p.Square(p.HNijOsn, p.RebroNijA) + p.Square(hVerhOsn, rebroVerhA));

Console.WriteLine("Volume: " + p.Volume());

p.ShowInfo();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13.interfaces

{

interface IParams

{

double Square(int hOsn, int rebro);

int PerimetrOsn(int rebroA, int rebroB, int rebroC);

double Volume();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13.interfaces

{

interface IShowInfo

{

void ShowInfo();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13.interfaces

{

interface ISumNum

{

int SumNum(int num);

}

}