|  |  |
| --- | --- |
| Без властивостей | З властивостями |
| #include<iostream>  #include <string>  using namespace std;  class Client  {  private:  double sum;  public:  double getSum()  {  return sum;  }  void setSum(double value)  {  if (value >= 0)  {  sum = value;  }  else  throw "error";  }  };  void main()  {  Client cl;  cl.setSum(23);  cout << "sum=" << cl.getSum() << endl;  system("pause");  } | #include<iostream>  #include <string>  using namespace std;  ref class Client  {  private:  double sum;  public:  property double Sum  {  double get()  {  return sum;  }  void set(double value)  {  if (value >= 0)  {  sum = value;  }  else  throw "error";  }  }  };  void main()  {  Client cl;  cl.Sum=23; //виклик метода set(23)  cout << "sum=" << cl.Sum << endl; //виклик метода get()  system("pause");  } |

#include<iostream>

#include <string>

using namespace std;

ref class Client

{

private:

double sum;

int id;

public:

property double Sum

{

double get()

{

return sum;

}

void set(double value)

{

if (value >= 0)

{

sum = value;

}

else

throw "error";

}

}

property int ID

{

int get()

{

return id;

}

void set(int value)

{

if (value >= 0)

{

id = value;

}

else

throw "error";

}

}

};

void main()

{

Client cl;

cl.Sum=23; //виклик метода set(23)

cout << "sum=" << cl.Sum << endl; //виклик метода get()

cl.ID = 235623; //виклик метода set(235623)

cout << "ID=" << cl.ID << endl; //виклик метода get()

system("pause");

}

|  |  |
| --- | --- |
| Без індексованих властивостей | З індексованими властивостями |
| #include<iostream>  #include <string>  using namespace std;  ref class Truk  {  private:  double a,b,c;    public:  property double A  {  double get()  {  return a;  }  void set(double value)  {  if (value >= 0)  {  a = value;  }  else  throw "error";  }  }  property double B  {  double get()  {  return b;  }  void set(double value)  {  if (value >= 0)  {  b = value;  }  else  throw "error";  }  }  property double C  {  double get()  {  return c;  }  void set(double value)  {  if (value >= 0)  {  c = value;  }  else  throw "error";  }  }  };  void main()  {  Truk tr;  tr.A = 12; //виклик метода set(12)  tr.B = 5;  tr.C = 11;  cout << "A=" << tr.A << endl; //виклик метода get()  system("pause");  } | #include<iostream>  #include <string>  using namespace std;  ref class Truk  {  private:  double a,b,c;    public:  property double default[int] //Індексатор (все одно що індексована властивість, але не треба вказувати імені властивості)  {  double get(int i)  {  switch (i)  {  case 1: return a;  case 2: return b;  case 3: return c;  default: throw "bad index";  }  }  void set(int i,double value)  {  if (value >= 0)  {  switch (i)  {  case 1: a = value;  break;  case 2: b = value;  break;  case 3: c = value;  break;  default: throw "bad index";  }  }  else  throw "error";  }  }  property double Sides[int] //Індексована властивість для доступу до сторін трикутника за номером сторони  {  double get(int i)  {  switch (i)  {  case 1: return a;  case 2: return b;  case 3: return c;  default: throw "bad index";  }  }  void set(int i,double value)  {  if (value >= 0)  {  switch (i)  {  case 1: a=value;  break;  case 2: b=value;  break;  case 3: c=value;  break;  default: throw "bad index";  }  }  else  throw "error";  }  }  //------------------------------------------------  property double A //Властивість для доступу до сторони а  {  double get()  {  return a;  }  void set(double value)  {  if (value >= 0)  {  a = value;  }  else  throw "error";  }  }  property double B //Властивість для доступу до сторони b  {  double get()  {  return b;  }  void set(double value)  {  if (value >= 0)  {  b = value;  }  else  throw "error";  }  }  property double C //Властивість для доступу до сторони с  {  double get()  {  return c;  }  void set(double value)  {  if (value >= 0)  {  c = value;  }  else  throw "error";  }  }  //----------------------------------------  property double Perimeter //Властивість периметр  {  double get()  {  return a + b + c;  }  }  double getPerimeter() //Метод знаходження перметру  {  return a + b + c;  }  };  void main()  {  Truk tr;  tr.A = 12; //виклик метода set(12)  tr.Sides[1] = 12;  tr[1] = 12;  tr.B = 5;  tr.Sides[2] = 5;  tr[2] = 5;  tr.C = 11;  tr.Sides[3] = 11;  tr[3] = 11;    cout << tr.getPerimeter() << endl; //Виклик функції  cout << tr.Perimeter << endl; //Звертання до властивості  for (int i = 1; i <= 3; i++)  {  cout << tr.Sides[i] << endl; //Звертання до індексованої властивості  }  for (int i = 1; i <= 3; i++)  {  cout << tr[i] << endl; //Звертання до індексатора  }  cout << "A=" << tr.A << endl; //виклик метода get()  system("pause");  } |