Rozwiązania zadań 3 i 4:

```
(---1---)
#include <iostream>
using namespace std;
(---2---)
zespol(double r = 0., double i = 0.): re(r), im(i){}
zespol operator+(double d){return zespol(re+d, im);}
(---3---)
zespol operator+(int i, zespol c){
      zespol res;
      res.re = static_cast<double>(i) + c.re;
      res.im = c.im;
     return res;
}
ostream& operator<<(ostream& str, zespol c){
      str << "(" << c.re << ", " << c.im << ")";
      return str;
}
(---4---)
#include <iostream>
(---5---)
virtual
(---6---)
virtual
(---7---)
: public figura
(---8---)
virtual double pole() {return 0.5*podst*wys;}
virtual void wypisz() {cout << "Jestem trojkatem - moje pole :";}
(---9---)
wskf = & t2;
```

Rozwiązania zadań 3 i 4 – pełny kod programu:

```
//(---1---)->
#include <iostream>
using namespace std;
//(---1---)<-

class zespol{
public:
    double re, im;
    //(---2---)->
       zespol(double r = 0., double i = 0.): re(r), im(i){}
       zespol operator+(double d){return zespol(re+d, im);}
       //(---2---)<-
};
//(---3---)->
zespol operator+(int i, zespol c){
       zespol res;
       res.re = static_cast<double>(i) + c.re;
       res.im = c.im;
    return res;
}
ostream& operator<<(ostream& str, zespol c){
       str << "(" << c.re << ", " << c.im << ")";
       return str;
}
//(---3---)<-

//(---4---)
class figura{
public:
    double podst, wys;
    figura(double p = 0., double w = 0.) : podst(p), wys(w){}
    //(---5---) double pole() = 0;
       virtual double pole() = 0;
    //(---6---) void wypisz()=0;
       virtual void wypisz()=0;
};
// class trojkat (---7---) {
class trojkat :  public figura {
public:
    trojkat(double p = 0., double w = 0.): figura(p, w){}
       //(---8---)->
       virtual double pole() {return 0.5*podst*wys;}
       virtual void wypisz() {cout << "Jestem trojkatem - moje pole: ";}
       //(---8---)<-
};

int main()
{
    int i = -3;
    double d = 7.2;
    zespol z1, z2(1, -1);     // z1 = (0,0)
    cout << "z1 = " << z1 << endl;
    cout << "z2 = " << z2 << endl;
    z1 = z2 + d;
    z2 = i + z1;
    cout << "z1 = " << z1 << endl;
    cout << "z2 = " << z2 << endl;

       trojkat t1, t2(2, 5);
    figura * wskf;
    wskf = & t2; //(---9---)
    wskf->wypisz();
       cout << wskf->pole() << endl;

       system("PAUSE");
       return 0;
}
```