eccezioni

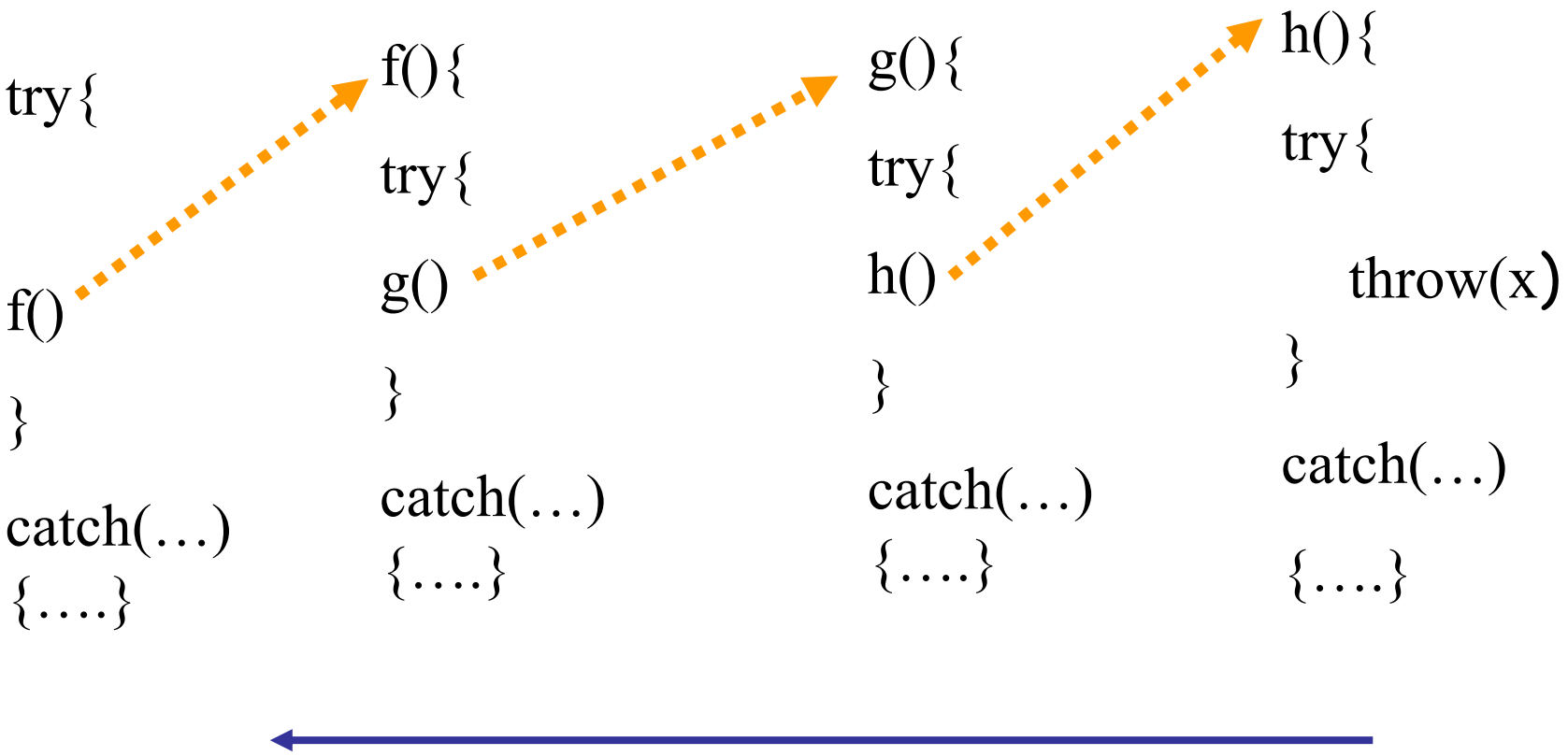
3 nuove istruzioni del C++:

- try : evidenzia un blocco in cui possono essere sollevate eccezioni
- throw(exp) : solleva l'eccezione, exp ha tipo T
- catch(T x): la "afferra" e la gestisce se ha il tipo T

```
main()
{
try
 {int x;
  cin >> x;
  f(x);
 }
 catch(int x)
 {switch(x)
     {
     case 1: cout << "errore 1"<<endl; break;
     case 2: cout << "errore 2"<<endl; break;
     case 3: cout << "nessun errore"<< endl;
     }
 }}
```

```
void f(int x)
{
  if(x==1)
    throw(1);
  if(x==2)
    throw(2);
  throw(3);
}
```

concordanza di tipo



i catch vengono esaminati in ordine
viene preso il primo con parametro di
tipo = a quello sollevato dal throw

try e catch possono essere anche dentro un ciclo

quindi dopo il catch il ciclo continua

```
main()
{
 int c=0; bool controllato=false;
 char A[]="pip4oplu5srom1wst3a12er";
 int s=sizeof(A);
 while(c<10 && !controllato)
  {
   try {
         F(A,s-1);
         cout<<"stringa ok="<<A<<endl;
         controllato=true;
       }
   catch(int i)
   {
        c++;
         cout<<"trovato carattere strano in posizione="<<i <<endl;
         cout<<"errore n."<<c<<endl;
         A[i]='x';
         cout<<A<<endl;
         }}}
```

```
void F(char *A, int dim)
{
  for(int i=0; i<dim; i++)
    if(A[i]<'a' || A[i]>'z')
      throw(i);
}
```

una catch può contenere una throw

che verrà gestita da un’altra catch

```
main()
{try
  {….
   while(!controllato)
     {try {  F(A, s-1);
              cout<<"stringa ok="<<A<<endl;
              controllato=true;
             }
             catch(int i)
             {
              cout<<"trovato carattere strano in posizione="<<i<<endl;
              c++;
              cout<<"errore n."<<c<<endl;
              A[i]='x';
              cout<<A<<endl;
              if(c==4)
                throw(c);
             } }
  }
  catch(int i)  {cout<<“trovati "<< c <<" errori: termino"<<endl;}
}
```

```
void F(char *A, int dim)
{
  for(int i=0; i<dim; i++)
    if(A[i]<'a' || A[i]>'z')
      throw(i);
}
```

catch che lancia una throw

tipi struttura lanciati dalle eccezioni

```
struct err{ string mess; int pos;
  err(string S, int i)  // costruttore
  {mess= S; pos=i;}
  err(){}
};

main()
try {
  char A[]="pip4o";
  F(A,5);
  cout<<A<<endl;

}
catch(err  x)
{cout<<“trovato "<<x.mess<< " in pos=" <<x.pos<<endl;}
```

```
void F(char *A, int dim)
{
  for(int i=0; i<dim; i++)
    if(A[i]<'a' || A[i]>'z')
      {       err E(“errore tal dei tali”,i);
              throw(E);
       }
}
```