

```

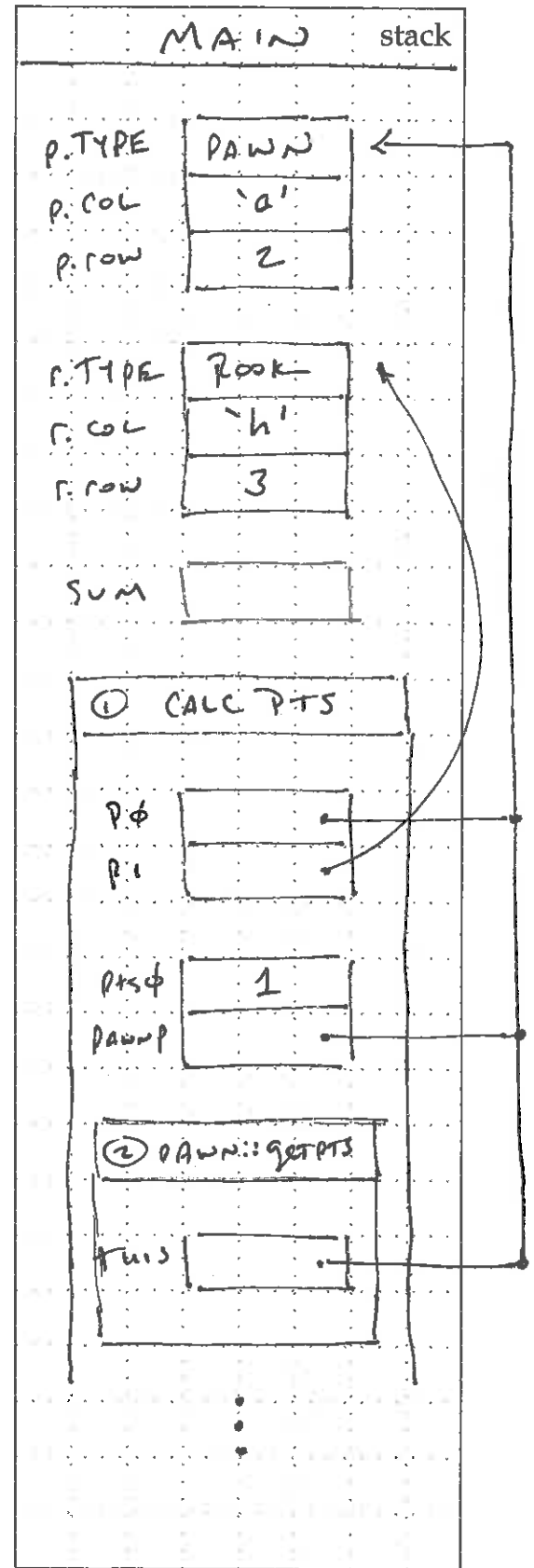
class Pawn : public Piece
{
    int get_pts() const { return 1; }
    ...
};

int calc_pts( Piece* p0, Piece* p1 )
{
    int pts0;
    if ( p0->get_type() == PAWN ) {
        Pawn* pawn_p = (Pawn*) p0;
        pts0 = pawn_p->get_pts();
    }
    else if ( p0->get_type() == ROOK ) {
        Rook* rook_p = (Rook*) p0;
        pts0 = rook_p->get_pts();
    }

    int pts1;
    if ( p1->get_type() == PAWN ) {
        Pawn* pawn_p = (Pawn*) p1;
        pts1 = pawn_p->get_pts();
    }
    else if ( p1->get_type() == ROOK ) {
        Rook* rook_p = (Rook*) p1;
        pts1 = rook_p->get_pts();
    }

    return pts0 + pts1;
}

int main( void )
{
    Pawn p('a',2);
    Rook r('h',3);
    int sum = calc_pts( &p, &r );
    return 0;
}
    
```



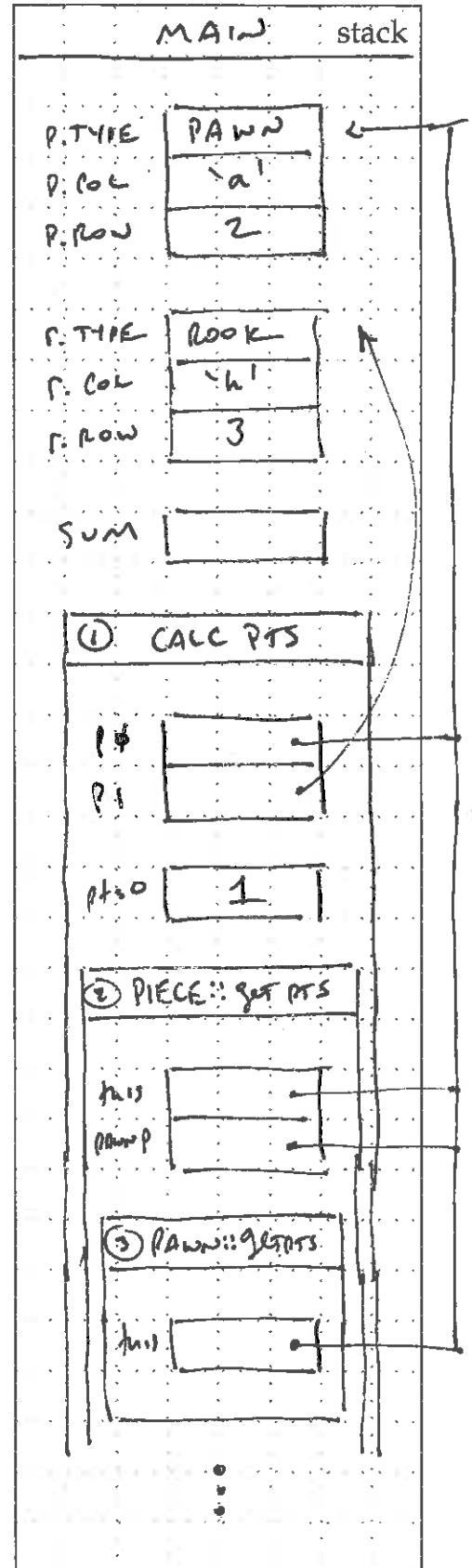
```

class Piece
{
    ...
    int get_pts() const
    {
        if ( m_type == PAWN ) {
            Pawn* pawn_p = (Pawn*) this;
            return pawn_p->get_pts();
        }
        else if ( m_type == ROOK ) {
            Rook* rook_p = (Rook*) this;
            return rook_p->get_pts();
        }
    }
};

class Pawn : public Piece
{
    int get_pts() const { return 1; }
    ...
};

int calc_pts( Piece* p0, Piece* p1 )
{
    int pts0 = p0->get_pts();
    int pts1 = p1->get_pts();
    return pts0 + pts1;
}

int main( void )
{
    Pawn p('a',2);
    Rook r('h',3);
    int sum = calc_pts( &p, &r );
    return 0;
}
    
```



```

class Piece
{
    virtual int get_pts() const = 0;
    ...
};

class Pawn : public Piece
{
    int get_pts() const { return 1; }
    ...
};

class Rook : public Piece
{
    int get_pts() const { return 5; }
    ...
};

int calc_pts( Piece* p0, Piece* p1 )
{
    int pts0 = p0->get_pts();
    int pts1 = p1->get_pts();
    return pts0 + pts1;
}

int main( void )
{
    Pawn p('a',2);
    Rook r('h',3);
    int sum = calc_pts( &p, &r );
    return 0;
}
    
```

