```
#ifndef _YAHTZEEMODEL_H
#define _YAHTZEEMODEL_H

#include "tvector.h"
#include "scorecard.h"
#include "cupofdice.h"

class YahtzeeView;

class IYahtzeeModel
{
  public:

    virtual int getScore(int scoreIndex) = 0;
    virtual bool doScore(int scoreIndex) = 0;
    virtual void rollDice(tvector<bool>& frozen) = 0;
    virtual void newGame() = 0;
};

class YahtzeeModelImpl : public IYahtzeeModel
{
  public:
    YahtzeeModelImpl(YahtzeeView * view);
    virtual int getScore(int scoreIndex);
    virtual bool doScore(int scoreIndex);
    virtual void newGame();
    virtual void rollDice(tvector<bool>& frozen);

  private:
    CupOfDice myDice;
    ScoreCard myScoreCard;
    tvector<ScoreEntry *> myEntries;
    int       myRollCount;
    YahtzeeView * myView;
    int       myScoredRolls;
};

#endif
```

```cpp
#include "yahtzeemodel.h"
#include "yahtzeeview.h"

YahtzeeModelImpl::YahtzeeModelImpl(YahtzeeView * view)
    : myDice(5),
      myRollCount(0),
      myView(view),
      myScoredRolls(0)
{
    myEntries = myScoreCard.getEntries();
}

void YahtzeeModelImpl::newGame()
{
    myScoredRolls = myRollCount = 0;
    myScoreCard = ScoreCard();
    myEntries = myScoreCard.getEntries();
    myView->showAbove(0);
    myView->showBonus(0);
    myView->showBelow(0);
    myView->showTotal(0);
}


int YahtzeeModelImpl::getScore(int scoreIndex)
{
    return myEntries[scoreIndex]->calcScore(myDice);
}

bool YahtzeeModelImpl::doScore(int scoreIndex)
{
    ScoreEntry * entry = myEntries[scoreIndex];
    if (entry->getScore() != -1){
        return false;
    }
    entry->scoreIt(myDice);
    myRollCount = 0;
    myDice.unfreezeAll();
    myView->enableRoll(true);

    int low,up;
    myView->showBelow(low = myScoreCard.getBelowScore());
    myView->showAbove(up = myScoreCard.getAboveScore());
    int bonus = 0;
    if (up >= 63) bonus = 35;
    myView->showBonus(bonus);
    myView->showTotal(low+up+bonus);

    myView->setScore(scoreIndex,entry->getScore());
    myScoredRolls++;
    if (myScoredRolls >= 13){
        myView->gameOver();
    }

    return true;
}

void YahtzeeModelImpl::rollDice(tvector<bool>& frozen)
{
    for(int k=0; k < frozen.size(); k++){
        if (frozen[k]){
            myDice.freeze(k);
        }
        else {
            myDice.unfreeze(k);
        }
    }
    myDice.roll();
    myView->showRolls(myDice);
```



```cpp
    myRollCount++;
    if (myRollCount == 3){
        myView->enableRoll(false);
    }
}
```

```
#ifndef _YAHTZEEVIEW_H
#define _YAHTZEEVIEW_H

#include "dicegroup.h"

class YahtzeeView
{
  public:

    virtual void showRolls(DiceGroup& dg) = 0;
    virtual void showScore(int scoreIndex, int score) = 0;
    virtual void setScore(int scoreIndex, int score) = 0;
    virtual void showAbove(int total) = 0;
    virtual void showBelow(int total) = 0;
    virtual void showBonus(int total) = 0;
    virtual void showTotal(int total) = 0;

    virtual void enableRoll(bool value) = 0;
    virtual void gameOver() = 0;
};

#endif
```