###以gradle方式导入项目

xjtu-courses

###目录如下：

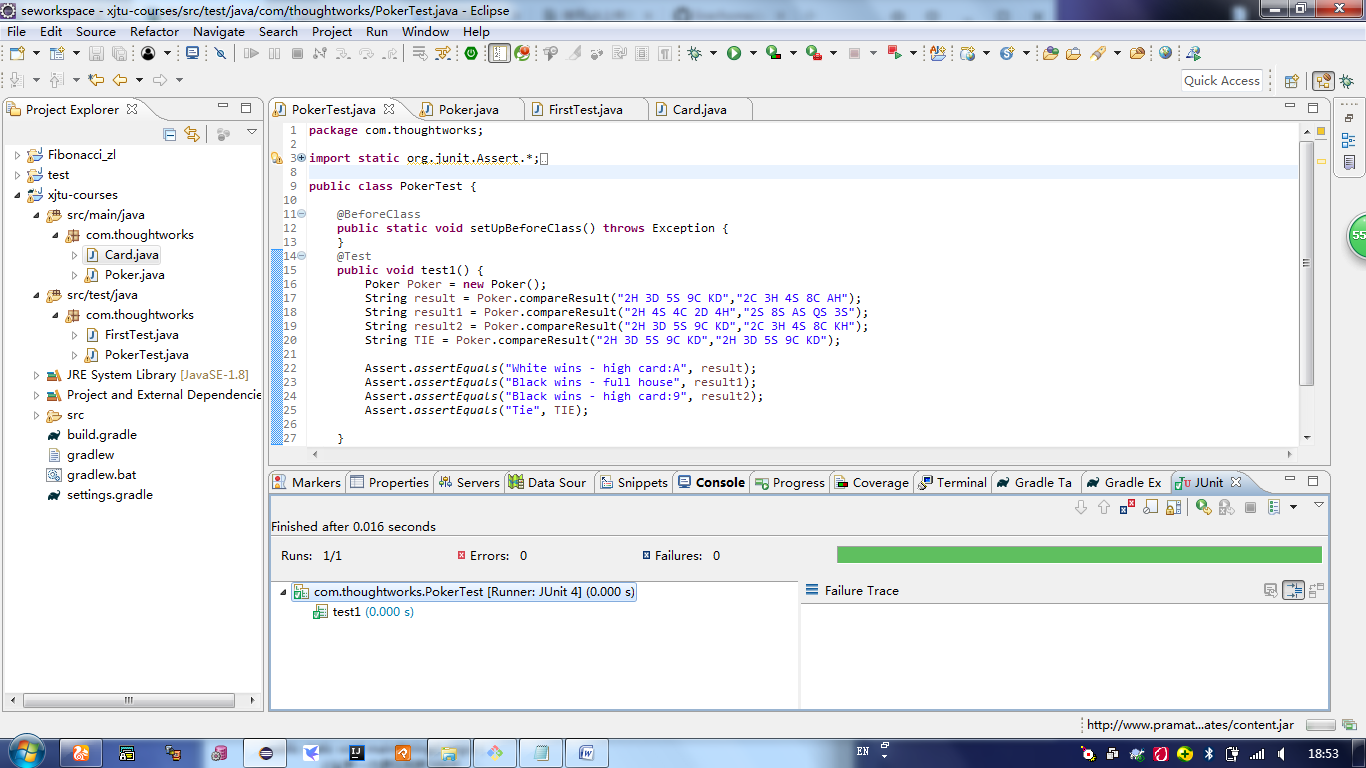
---- /xjtu-courses/src/main/java/com/thoughtworks/Card.java 是一个实体类

---- /xjtu-courses/src/main/java/com/thoughtworks/Poker.java 扑克具体实现函数

----/xjtu-courses/src/test/java/com/thoughtworks/FirstTest.java 测试测试环境正确

----/xjtu-courses/src/test/java/com/thoughtworks/PokerTest.java 测试扑克程序的类

实验结果截图：



具体核心代码实现如下：

public class Poker {

//初始化

Card card = new Card('0', 0, '0');

Card[] Black = new Card[5];

Card[] White = new Card[5];

char []blackOrder = new char[] {'0','0','0','0','0'};

char []whiteOrder = new char[] {'0','0','0','0','0'};

//比较

public String compareResult(String str1, String str2) {

dealInputString(str1, str2);

Sort(Black,White);

int i;

int ruleBlack = getRule(Black,blackOrder);

int ruleWhite = getRule(White,whiteOrder);

String[] ruleName = new String[10];

ruleName[1] = "high card";

ruleName[2] = "pair";

ruleName[3] = "two pairs";

ruleName[4] = "three of a kind";

ruleName[5] = "straight";

ruleName[6] = "flush";

ruleName[7] = "full house";

ruleName[8] = "four of a kind";

ruleName[9] = "straight flush";

//有序两个数组比较

if (ruleBlack > ruleWhite) {

return "Black wins - "+ruleName[ruleBlack];

}else if (ruleBlack < ruleWhite) {

return "White wins - "+ruleName[ruleWhite];

}else {

for (i = 0; i < 5; i++) {

if (blackOrder[i] > whiteOrder[i]) {

if (blackOrder[i] == 'A') {

blackOrder[i] = 'T';

}else if (blackOrder[i] == 'K') {

blackOrder[i] = 'Q';

}else if (blackOrder[i] == 'Q') {

blackOrder[i] = 'K';

}else if (blackOrder[i] == 'T') {

blackOrder[i] ='A';

}

return "Black wins - "+ruleName[ruleBlack]+":"+blackOrder[i];

}else if (blackOrder[i] < whiteOrder[i]) {

//ASCII码 T>J>Q>K>A 而我们要的 A>K>Q>J>T 需要转换

if (whiteOrder[i] == 'A') {

whiteOrder[i] = 'T';

}else if (whiteOrder[i] == 'K') {

whiteOrder[i] = 'Q';

}else if (whiteOrder[i] == 'Q') {

whiteOrder[i] = 'K';

}else if (whiteOrder[i] == 'T') {

whiteOrder[i] ='A';

}

return "White wins - "+ruleName[ruleWhite]+":"+whiteOrder[i];

}

}

if(i == 5) {

return "Tie.";

}

}

return null;

}

//9种规则制定

private int getRule(Card[] card, char order[]) {

int rule;

rule = 9;

if(Is5ConsecutiveCards(card) && IsSameColor(card)) {

order[0] = card[4].getName();

return rule;

}

rule --;

if (Is4SameValue(card)) {

order[0] = card[1].getName();

return rule;

}

rule --;

if (Is3SameValue(card,0,1,2) && Is2SameValue(card,3,4) || Is2SameValue(card,0,1) && Is3SameValue(card,2,3,4) ) {

order[0] = card[2].getName();

return rule;

}

rule --;

if (IsSameColor(card)) {

for (int i = 0; i < 5; i++) {

order[i] = card[4-i].getName();

return rule;

}

}

rule --;

if (Is5ConsecutiveCards(card)) {

order[0] = card[4].getName();

return rule;

}

rule --;

if (Is3SameValue(card, 0, 1, 2) || Is3SameValue(card, 1, 2, 3) || Is3SameValue(card, 2, 3, 4)) {

order[0] = card[2].getName();

return rule;

}

rule --;

int k = 9 ;

if (Is2SameValue(card, 0, 1) || Is2SameValue(card, 2, 3)) {

k = 4;

}else if (Is2SameValue(card, 1, 2) || Is2SameValue(card, 3, 4)){

k = 0;

}else if (Is2SameValue(card, 0, 1) || Is2SameValue(card, 3, 4)) {

k = 2;

}

if (k!=9) {

order[0] = card[3].getName();

order[1] = card[1].getName();

order[2] = card[k].getName();

return rule;

}

rule--;

int m = 9,m1 =0,m2 =0 ,m3= 0;

if (Is2SameValue(card, 0, 1)) {

m = 0;

m1 = 4;

m2 = 3;

m3 = 2;

}else if (Is2SameValue(card, 1, 2)) {

m = 1;

m1 = 4;

m2 = 3;

m3 = 0;

}else if (Is2SameValue(card, 2, 3)){

m = 2;

m1 = 4;

m2 = 1;

m3 = 0;

}else if (Is2SameValue(card, 3, 4)) {

m = 3;

m1 = 2;

m2 = 1;

m3 = 0;

}

if (m!=9) {

order[0] = card[m].getName();

order[1] = card[m1].getName();

order[2] = card[m2].getName();

order[3] = card[m3].getName();

return rule;

}

rule--;

for (int i = 0; i < 5; i++) {

order[i] = card[4-i].getName();

}

return rule;

}

private boolean Is2SameValue(Card[] card, int i, int j) {

if((card[i].getName() == card[j].getName())) {

return true;

}

return false;

}

private boolean Is3SameValue(Card[] card, int i, int j, int k) {

if((card[i].getName() == card[j].getName()) && ( card[j].getName()== card[k].getName())) {

return true;

}

return false;

}

private boolean Is4SameValue(Card[] card) {

if ((card[1].getName() == card[2].getName()) && (card[3].getName() == card[2].getName()) && (card[3].getName() == card[4].getName())){

return true;

}

if ((card[0].getName() == card[1].getName()) && (card[3].getName() == card[2].getName()) && (card[1].getName() == card[2].getName())){

return true;

}

return false;

}

private boolean IsSameColor(Card[] card) {

for (int i = 1; i < card.length; i++) {

if (card[i].getColor() != card[i-1].getColor()) {

return false;

}

}

return true;

}

private boolean Is5ConsecutiveCards(Card[] card) {

for (int i = 1; i < card.length; i++) {

if ((card[i].getName()-card[i-1].getName()) !=1) {

return false;

}

}

return true;

}

//排序

private void Sort(Card[] black, Card[] white) {

for (int j = 0; j < black.length; j++) {

for (int i = 1; i < black.length; i++) {

if (Black[i].getName() < Black[i-1].getName()) {

Card tmp = Black[i-1].card;

char tmpName = Black[i-1].getName();

int tmpValue = Black[i-1].getValue();

char tmpColor = Black[i-1].getColor();

Black[i-1].setName(Black[i].getName());

Black[i-1].setValue(Black[i].getValue());

Black[i-1].setColor(Black[i].getColor());

Black[i].setName(tmpName);

Black[i].setValue(tmpValue);

Black[i].setColor(tmpColor);

}

}

}

}

//处理输入流

private void dealInputString(String str1, String str2) {

int count = 0;

for (int i = 0; i < str1.length(); i+=3) {

Black[count] = new Card(str1.charAt(i), i, str1.charAt(i+1));

Black[count].setName(str1.charAt(i));

Black[count].setValue(i);

Black[count].setColor(str1.charAt(i+1));

count++;

}

for (int i = 0; i < 5; i++) {

if (Black[i].getName() == 'T') {

Black[i].setName('A');

}else if (Black[i].getName() == 'Q') {

Black[i].setName('K');

}else if (Black[i].getName() == 'K') {

Black[i].setName('Q');

}else if (Black[i].getName() == 'A') {

Black[i].setName('T');

}

}

count = 0;

for (int i = 0; i < str2.length(); i+=3) {

White[count] = new Card(str2.charAt(i), i, str2.charAt(i+1));

White[count].setName(str2.charAt(i));

White[count].setValue( i);

White[count].setColor(str2.charAt(i+1));

count++;

}

for (int i = 0; i < 5; i++) {

if (White[i].getName() == 'T') {

White[i].setName('A');

}else if (White[i].getName() == 'Q') {

White[i].setName('K');

}else if (White[i].getName() == 'K') {

White[i].setName('Q');

}else if (White[i].getName() == 'A') {

White[i].setName('T');

}

}

}

}