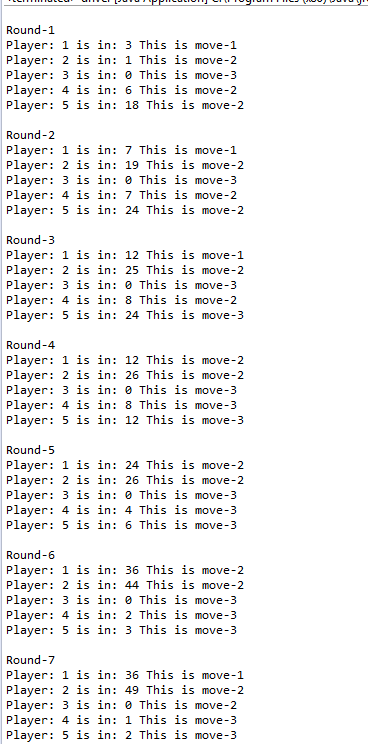
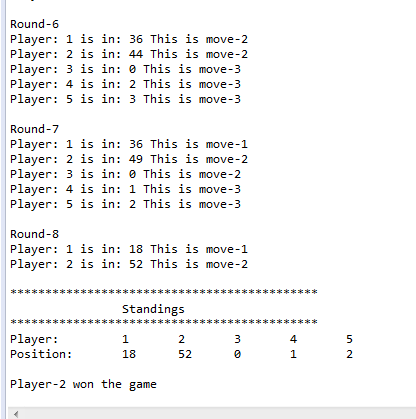
Prince Ikede

Lab Work

I completed the Labs. I used both and interface and abstract class. I also used arrays to hold the players.





**public** **class** driver {

**public** **static** **void** main(String[] args) {

handler gamer = **new** handler(5);

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

gamer.addPlayer();

gamer.playGame();

gamer.standings();

}

}

**import** java.util.\*;

**public** **class** Player {

**int** position;

**int** playerNum;

Random rand;

MoveType movetype;

**int** response;

Positioning play;

**public** Player(**int** playerNum, Positioning play){

position = 0;

**this**.playerNum = playerNum;

rand = **new** Random();

**this**.play = play;

resetResponse();

}

**public** **int** getPosition(){

**return** position;

}

**public** **void** resetResponse(){

movetype = Resposnes.*getResponse*();

response = 1 + rand.nextInt(4);

}

**public** **void** makeMove(){

**int** dieThrow = 1 + rand.nextInt(6);

**if**( response > 0 ){

position += movetype.moveBy(dieThrow, position, play);

response--;

}

**else** resetResponse();

}

**public** String toString(){

**return** "Player: " + (playerNum+1) + " is in: " + position + " "+ movetype.toString();

}

}

**import** java.util.\*;

**abstract** **class** MoveType{

**abstract** **int** moveBy(**int** dieThrow, **int** playerPos, Positioning handler);

**public** String toString(){

**return** " ";

}

**public** String toString(**int** move){

**return** ("This is move-"+ move);

}

}

**class** MoveOne **extends** MoveType{

**int** changePosition;

**int** move =1;

**public** **int** moveBy(**int** dieThrow, **int** playerPos, Positioning handler) {

changePosition = (dieThrow + (playerPos - handler.getTrailer()))/2;

**if**(dieThrow<2){

**return** (-1 \* changePosition);

}

**return** changePosition;

}

**public** String toString(){

**return** **super**.toString(move);

}

}

**class** MoveTwo **extends** MoveType{

**int** changePosition;

**int** move = 2;

**public** **int** moveBy(**int** dieThrow, **int** playerPos, Positioning handler) {

**if**(dieThrow%2 ==0){

changePosition = (3\*dieThrow);

**return** changePosition;

}

**return** dieThrow;

}

**public** String toString(){

**return** **super**.toString(move);

}

}

**class** MoveThree **extends** MoveType{

**int** changePosition;

**int** move = 3;

**public** **int** moveBy(**int** dieThrow, **int** playerPos, Positioning handler) {

changePosition = (playerPos - handler.getTrailer())/2;

**if**(dieThrow<3){

**return** changePosition;

}

**return** (-1\*changePosition);

}

**public** String toString(){

**return** **super**.toString(move);

}

}

**class** Resposnes{

**public** **static** MoveType getResponse(){

Random rand = **new** Random();

**int** respondWith = 1 + rand.nextInt( 3 );

MoveType respond = **null**;

**switch**( respondWith ){

**case** 1: respond = **new** MoveOne();

**break**;

**case** 2: respond = **new** MoveTwo();

**break**;

**case** 3: respond = **new** MoveThree();

}

**return** respond;

}

}

**interface** Positioning {

**int** getLeader();

**int** getTrailer();

**void** playGame();

}

**class** handler **implements** Positioning{

Player[] players;

**int** numPlayer;

**final** **int** WIN = 50;

**int** won = 1; //assumes at least one player playing

**public** handler(**int** numPlaying){

players = **new** Player[numPlaying];

numPlayer = 0;

}

**public** **void** addPlayer(){

players[numPlayer] = **new** Player(numPlayer, **this**);

numPlayer++;

}

@Override

**public** **int** getLeader() {

**int** pos = players[0].position;

**for**(**int** i = 1; i < numPlayer; i++){

**if**(players[i].position > pos )

pos = players[i].position;

}

**return** pos;

}

@Override

**public** **int** getTrailer() {

**int** pos = players[0].position;

**for**(**int** i = 1; i < numPlayer; i++){

**if**(players[i].position < pos)

pos = players[i].position;

}

**return** pos ;

}

**public** **void** playGame(){

**int** round = 1;

**int** playerToPlay = 0;

System.***out***.println("\nRound-" + round);

**while**( !winner() ){

players[playerToPlay].makeMove();

System.***out***.println( players[playerToPlay].toString() );

playerToPlay++;

**if**( playerToPlay > numPlayer - 1 ){

round++;

System.***out***.println("\nRound-" + round);

playerToPlay = 0;

}

}

}

**private** **boolean** winner(){

**for**(**int** i = 1; i < numPlayer; i++){

**if**(players[i].position >= WIN){

won = i+1;

**return** **true**;

}

}

**return** **false**;

}

**public** **int** whoWon(){

**return** won;

}

**public** **void** standings(){

System.***out***.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("\t\tStandings");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.print("Player: \t");

**for**(**int** i=0; i<players.length; i++){

System.***out***.print((i+1)+"\t");

}

System.***out***.print("\nPosition: \t");

**for**(**int** i=0; i<players.length; i++){

System.***out***.print(players[i].getPosition()+"\t");

}

System.***out***.println("\n\nPlayer-"+ whoWon()+ " won the game");

}

}