SNAKE AND LADDER

UC 1: SNAKE AND LADDER GAME PLAYED WITH SINGLE PLAYER AT START POSITION 0

package com.snakeLadder;

import java.util.Scanner;

public class snakeLadderUc1 {

public static void main(String[] args) {

int START\_POSITION = 0;

System.out.println("Player start at position " +START\_POSITION );

}

}

UC2: To Get Number Between 1 To 6

package com.snakeLadder;

import java.util.Scanner;

public class snakeLadderUc2 {

public static void main(String[] args) {

int START\_POSITION = 0;

System.out.println("Player start at position " +START\_POSITION );

int ROLL\_THE\_DICE = (int) (Math.floor(Math.random() \*10) %6 ) +1;

System.out.println("After rolling the dice we get " +ROLL\_THE\_DICE);

}

}

UC 3: To Check Option They Are No Play Ladder And Snake

package com.snakeLadder;

import java.util.Scanner;

public class snakeLadderUc3 {

public static final int NO\_PlAY = 1;

public static final int LADDER = 2;

public static final int SNAKE = 3;

public static final int START\_POSITION = 0;

public static void main(String[] args) {

int playerPosition = 0;

int ROLL\_THE\_DICE = (int) (Math.floor(Math.random() \*10) %6 ) +1;

System.out.println("After rolling the dice we get:" + ROLL\_THE\_DICE);

int playerCheck = (int)( Math.floor(Math.random() \* 10) % 3) + 1;

System.out.println("playerCheck:" +playerCheck);

switch(playerCheck){

case NO\_PlAY:

System.out.println("The player stay in the same position that is:" + playerPosition );

break;

case LADDER:

System.out.format("The player moves ahead by the %d ",ROLL\_THE\_DICE);

break;

case SNAKE:

System.out.format("The player got bitten by the snake moves behind by the %d :",ROLL\_THE\_DICE);

break;

default:

System.out.println("Something went wrong");

}

}

}

UC4: To Reaches The Winning Position 100

package com.snakeLadder;

import java.util.Scanner;

public class snakeLadderUc4 {

public static final int NO\_PlAY = 1;

public static final int LADDER = 2;

public static final int SNAKE = 3;

public static final int START\_POSITION = 0;

public static final int WIN = 100;

public static void main(String[] args) {

int playerPosition = 0;

while (playerPosition < WIN) {

int ROLL\_THE\_DICE = (int) (Math.floor(Math.random() \* 10) % 6) + 1;

System.out.println("After rolling the dice we get:" + ROLL\_THE\_DICE);

int playerCheck = (int) (Math.floor(Math.random() \* 10) % 3) + 1;

System.out.println("playerCheck:" + playerCheck);

switch (playerCheck) {

case NO\_PlAY:

System.out.println("The player stay in the same position that is:" + playerPosition);

break;

case LADDER:

playerPosition = playerPosition + ROLL\_THE\_DICE;

System.out.println("The player moves ahead by the:" + ROLL\_THE\_DICE);

break;

case SNAKE:

playerPosition = playerPosition - ROLL\_THE\_DICE;

if (playerPosition < 0) {

System.out.println("The player got bitten by snake and player restart from " + START\_POSITION);

}

break;

default:

System.out.println("Something went wrong");

}

System.out.println("Player position after rolling the dice is " + playerPosition);

}

}

}

UC5: To Report The Number Of Times The Dice Was Played To Win The Game And Also The Position After Every Die Role

package com.snakeLadder;

import java.util.Scanner;

public class snakeLadderUc5 {

public static final int NO\_PlAY = 1;

public static final int LADDER = 2;

public static final int SNAKE = 3;

public static final int START\_POSITION = 0;

public static final int WIN = 100;

public static void main(String[] args) {

int playerPosition = 0;

while (playerPosition <= WIN) {

int ROLL\_THE\_DICE = (int) (Math.floor(Math.random() \* 10) % 6) + 1;

System.out.println("After rolling the dice we get:" + ROLL\_THE\_DICE);

int playerCheck = (int) (Math.floor(Math.random() \* 10) % 3) + 1;

System.out.println("playerCheck:" + playerCheck);

switch (playerCheck) {

case NO\_PlAY:

System.out.println("The player stay in the same position that is:" + playerPosition);

break;

case LADDER:

playerPosition = playerPosition + ROLL\_THE\_DICE;

if (playerPosition <= WIN) {

System.out.println("The player moves ahead by the:" + ROLL\_THE\_DICE);

}

break;

case SNAKE:

playerPosition = playerPosition - ROLL\_THE\_DICE;

if (playerPosition < 0) {

System.out.println("The player got bitten by snake and player restart from " + START\_POSITION);

}

break;

default:

System.out.println("Something went wrong");

}

System.out.println("The player position after rolling the dice is " + playerPosition);

}

}

}

UC6: To Win The Game And Also The Position After Every Die Role

package com.snakeLadder;

import java.util.Scanner;

public class snakeLadderUc6 {

public static final int NO\_PlAY = 1;

public static final int LADDER = 2;

public static final int SNAKE = 3;

public static final int START\_POSITION = 0;

public static final int WIN = 100;

public static void main(String[] args) {

int diceRollCount = 0;

int playerPosition = 0;

while (playerPosition <= WIN) {

int ROLL\_THE\_DICE = (int) (Math.floor(Math.random() \* 10) % 6) + 1;

System.out.println("After rolling the dice we get:" + ROLL\_THE\_DICE);

int playerCheck = (int) (Math.floor(Math.random() \* 10) % 3) + 1;

System.out.println("playerCheck:" + playerCheck);

diceRollCount++;

switch (playerCheck) {

case NO\_PlAY:

System.out.println("The player stay in the same position that is:" + playerPosition);

break;

case LADDER:

playerPosition = playerPosition + ROLL\_THE\_DICE;

if (playerPosition <= WIN) {

playerPosition = playerPosition + ROLL\_THE\_DICE;

System.out.println("The player moves ahead by the:" + ROLL\_THE\_DICE);

}

break;

case SNAKE:

playerPosition = playerPosition - ROLL\_THE\_DICE;

if (playerPosition < 0) {

playerPosition = 0;

System.out.println("The player got bitten by snake and player restart from " + START\_POSITION);

}

break;

default:

System.out.println("Something went wrong");

}

System.out.println( "Player position after roll the dice is " + playerPosition);

}

System.out.println("Total number of time dice was played to win the game is "+ diceRollCount);

}

}

UC7: To Finally Report Which Player Won The Game

package com.snakeLadder;

import java.util.Scanner;

public class snakeLadderUc7 {

public static final int NO\_PlAY = 1;

public static final int LADDER = 2;

public static final int SNAKE = 3;

public static final int START\_POSITION = 0;

public static final int WIN = 100;

public static void main(String[] args) {

int diceRollCount = 0;

int playerPosition1 = 0;

int playerPosition2 = 0;

while ((playerPosition1 <= 100) && (playerPosition2 <= 100)){

int ROLL\_THE\_DICE1 = (int) (Math.floor(Math.random() \* 10) % 6) + 1;

int ROLL\_THE\_DICE2 = (int) (Math.floor(Math.random() \* 10) % 6) + 1;

System.out.println("After rolling the dice we get:" + ROLL\_THE\_DICE1);

int playerCheck1 = (int) (Math.floor(Math.random() \* 10) % 3) + 1;

int playerCheck2 = (int) (Math.floor(Math.random() \* 10) % 3) + 1;

System.out.println("playerCheck:" + playerCheck1);

diceRollCount++;

switch (playerCheck1) {

case NO\_PlAY:

System.out.println("The player stay in the same position that is:" + playerPosition1);

break;

case LADDER:

playerPosition1 = playerPosition1 + ROLL\_THE\_DICE1;

if (playerPosition1 <= WIN) {

playerPosition1 = playerPosition1 + ROLL\_THE\_DICE1;

System.out.println("The player moves ahead by the:" + ROLL\_THE\_DICE1);

}

break;

case SNAKE:

playerPosition1 = playerPosition1 - ROLL\_THE\_DICE1;

if (playerPosition1 < 0) {

System.out.println("The player got bitten by snake and player restart from " + START\_POSITION);

}

break;

default:

System.out.println("Something went wrong");

}

switch (playerCheck2) {

case NO\_PlAY:

System.out.println("The player stay in the same position that is:" + playerPosition1);

break;

case LADDER:

playerPosition2 = playerPosition2 + ROLL\_THE\_DICE2;

if (playerPosition2 <= WIN) {

playerPosition2 = playerPosition2 + ROLL\_THE\_DICE2;

System.out.println("The player moves ahead by the:" + ROLL\_THE\_DICE2);

}

break;

case SNAKE:

playerPosition2 = playerPosition2 - ROLL\_THE\_DICE1;

if (playerPosition2 < 0) {

System.out.println("The player got bitten by snake and player restart from " + START\_POSITION);

}

break;

default:

System.out.println("Something went wrong");

}

}

System.out.println("Final position of player1 is " + playerPosition1);

System.out.println("Final position of player2 is " + playerPosition2);

System.out.println( "Total number of time dice was played to win the game is " +diceRollCount);

if(playerPosition1 > playerPosition2){

System.out.println("Player1 win the game");

}else{

System.out.println("Player2 win the game");

}

}

}