// Facon Guillaume

/\*@Override

public boolean deplacementValide(Echiquier echec, Case depart, Case arrive) {

Case dpt = echec.getCase(depart.getLigne(), depart.getColonne());

Case arr = echec.getCase(arrive.getLigne(), arrive.getColonne());

if (!arr.caseOccupe() || !arr.caseOccupe(this.couleur)) { // case arrivée vide ou pièce couleur opposée

if(isDeplacementDroite(dpt, arr)) {

return deplacerHorizontalement(echec, dpt, arr);

}

if(isDeplacementGauche(dpt, arr)) {

return deplacerHorizontalement(echec, arr, dpt);

}

if(isDeplacementHaut(dpt, arr)) {

return deplacerVerticalement(echec, dpt, arr);

}

if(isDeplacementBas(dpt, arr)) {

return deplacerVerticalement(echec, arr, dpt);

}

System.out.println("Déplacement impossible.");

return false;

} else {

System.out.println("Déplacement impossible : un copain est sur la case demandée.");

return false;

}

}

private boolean isDeplacementBas(Case dpt, Case arr) {

return arr.getLigne() > dpt.getLigne() && arr.getColonne() == dpt.getColonne();

}

private boolean isDeplacementHaut(Case dpt, Case arr) {

return arr.getLigne() < dpt.getLigne() && arr.getColonne() == dpt.getColonne();

}

private boolean isDeplacementGauche(Case dpt, Case arr) {

return arr.getLigne() == dpt.getLigne() && arr.getColonne() < dpt.getColonne();

}

private boolean isDeplacementDroite(Case dpt, Case arr) {

return arr.getLigne() == dpt.getLigne() && arr.getColonne() > dpt.getColonne();

}

// private boolean deplacerBas(Echiquier echec, Case dpt, Case arr) {

// int compteur = 0;

// for(int i = dpt.getLigne()+1; i <= arr.getLigne()-1; i++) {

// if(!echec.getCase(i, dpt.getColonne()).caseOccupe()) {

// compteur++;

// }

// }

// if(compteur == Math.abs(arr.getLigne() - dpt.getLigne() )-1 ) {

// return true;

// } else {

// System.out.println("Il y a une pièce sur le chemin en bas.");

// return false;

// }

// }

public static boolean deplacerVerticalement(Echiquier echec, Case dpt, Case arr) {

int compteur = 0;

for(int i = dpt.getLigne()-1; i >= arr.getLigne()+1; i--) {

if(!echec.getCase(i, dpt.getColonne()).caseOccupe() ) {

compteur++;

}

}

if(compteur == Math.abs(arr.getLigne() - dpt.getLigne() )-1 ) {

return true;

} else {

System.out.println("Il y a une pièce sur le chemin en haut.");

return false;

}

}

// private boolean deplacerGauche(Echiquier echec, Case dpt, Case arr) {

// int compteur = 0;

// for(int i = dpt.getColonne()-1; i >= arr.getColonne()+1; i--) {

// if(!echec.getCase(dpt.getLigne(), i).caseOccupe()) {

// compteur++;

// }

// }

// if(compteur == Math.abs(arr.getColonne() - dpt.getColonne())-1 ) {

// return true;

// } else {

// System.out.println("Il y a une pièce sur le chemin à gauche.");

// return false;

// }

// }

public static boolean deplacerHorizontalement(Echiquier echec, Case dpt, Case arr) {

int compteur = 0;

for(int i = dpt.getColonne()+1; i <= arr.getColonne() -1; i++) {

if(!echec.getCase(dpt.getLigne(), i).caseOccupe()) {

compteur++;

}

}

if(compteur == Math.abs(arr.getColonne() - dpt.getColonne())-1) {

return true;

} else {

System.out.println("Il y a une pièce sur le chemin à droite.");

return false;

}

}

\*/