|  |
| --- |
| Erik Urban |
| Semestrálna práca  Hra Dáma |
| Skupina 5ZI1B |

|  |
| --- |
| 10.1.2021 |

**Dáma**

Tento projekt predstavuje stolovú hru Dáma. Hra obsahuje šachovnicu, ktorá sa skladá z červených a čiernych políčok. Figúrky sa môžu pohybovať iba po čiernych políčkach. Keď hráč príde o všetky figúrky, hra sa končí a vyhráva hráč, ktorého figúrky ostali na hracej ploche. Pri spustení hry, sa objavia obyčajné figúrky na v prvých troch radoch na oboch stranách, na čiernych políčkach. Každý hráč vlastní 20 figúrok jeho farby.

V Dáme existujú dva typy figúrok, obyčajná figúrka (nazývaná aj pešiak, kameň, panáčik) a dáma (niekedy nazývana aj kráľ). Pešiak je schopný sa posúvať o jedno políčko dopredu po diagonále vpred – tzv. ťah. V hre Dáma vždy prvý začína červený hráč, ktorý vykoná ťah figúrkou. Následne je na ťahu čierny hráč.

Jedinou výnimkou je skok. Hráč skáče figúrkou vtedy, ak je na ťahu, na diagonále pred figúrkou alebo za figúrkou je nepriateľská figúrka a za nepriateľskou figúrkou je voľné políčko tiež diagonálne, na ktoré figúrka, ktorá skáče, následne skočí. Po skoku sa preskočená figúrka odstráni z plochy, ďalej s ňou nie je možné hrať.  
Hráč nemôže preskočiť figúrku vlastnej farby. Figúrky tiež vedia skákať v jednom ťahu viac, ako jeden krát, ak sú splnené hore uvedené podmienky.

Na začiatku hry sa nachádzajú na šachovnici iba figúrky typu obyčajná figúrka.

Dáma je špeciálny typ figúrky, ktorý sa vie hýbať ľubovoľný počet políčok po diagonále. Pešiak sa stane dámou, ak sa dostane na posledný rad políčok protihráča. Ostatné pravidlá a schopnosti pre dámu sú rovnaké ako pre pešiaka.

V mojej Dáme je implementovaný Force Jump. To znamená že, ak je možné skákať figúrkou, hráč je POVINNÝ skočiť. Ak to nevykoná príde o figúrku. Toto pravidlo sa vzťahuje aj na viacnásobne skoky tou istou figúrkou.

Ak môžu skákať naraz dve figúrky a hráč nevykoná skok figúrkou, príde o obe figúrky. Ak však vykoná skok aspoň s jednou z nich, nepríde ani o jednu z nich.

**Hra sa ovláda nasledovne :**

Hráč klikne myškou na figúrku jeho farby, ak je na rade . Následne druhým kliknutím klikne na políčko kde sa chce presunúť.

**Základné Objekty**

Hlavným objektom, na ktorej sa Dáma odohráva je šachovnica. Šachovnica sa skladá z čiernych a červených políčok. Na šachovnici, na čiernych políčkach sa nachádzajú obyčajné figúrky, červené a čierne . Tieto figúrky môžu nadobudnúť podobu dámy, ak splnia pravidlá hry.

**Enum FarbaHraca**

Enum, ktorý reprezentuje možný typ hráča.  
Obsahuje tiež konštanty Cerveny a Cierny.  
**Konštruktor :**FarbaHraca(String farbaHraca) priraďuje hodnotu atribútu farbaHraca podľa prijatého parametra typu String.

**Metódy :**toString() – jedná sa o prekrytie metódy toString, ktorú dedí každý objekt. Vráti tam String.  
ohlasVyhercu() – táto metoda ohlási výhercu vo forme String.

**Trieda Figurka**

Táto trieda reprezentuje figurku.

**Konštruktor :**

public Figurka​(Suradnice suradnice, int velkost, TypFigurky typ, Policko policko, FarbaHraca hrac) - Konstruktor, ktory inicializuje atributy podla danych parametrov.

Parametre -   
suradnice - inicialiuje atribut suradnice podla tohto parametra.   
velkost - inicializuje atribut velkost podla tohto parametra.   
typ - inicialiuje atribut typ podla tohto parametra.   
policko - inicialiuje atribut policko podla tohto parametra.   
hrac - inicialiuje atribut hrac podla tohto parametra.

**Metódy :**

getHrac() - Getter pre atribut hrac.

getTypFigurky() – Getter pre atribút typ, vráti typ figurky , ktorú daná figúrka reprezentuje

getHrac() - Getter pre atribút hráč, vráti akému hráčovi daná figúrka patri.

getPolicko() - Getter pre atribút políčko, vráti na akom políčku stoji figúrka.

prekresliFigurku(Policko policko) – Nastaví figúrku na dane políčko,

zároveň upraví jeho podobu na požadovaný tvar.

zmazFigurku() – Táto metóda zmaže figúrku.

**Trieda Hra** :

Táto trieda reprezentuje logiku hry. Deklarujem atribúty, ktoré reprezentujú manažéra hry, farbu hráča, ktorý je na ťahu, figúrky, pozície x a y, boolean, ktorý zisťuje, ci sa hra ma vypnúť a šachovnicu

**Konštruktor :**

Public Hra() – Konštruktor hry ktory inicializuje atributy

**Metódy:**

skok​(Policko obsadenePolicko, int dy, int dx, Policko polickoPoSkonceni) - Metoda, ktora reprezentuje skok figurky dopredu / dozadu po diagonale.

skokDamou​(int dy, int dx, Policko polickoPoSkonceni) - Metoda, ktora reprezentuje pohyb damy .

spusti() - Tato metoda nastavi, aby program reagoval na input mysi a nastavi aby bol cerveny hrac na rade.

ukonci() - Tato metoda ukonci hru a vyhlasi vitaza, tiez nastavy atribut vypniSa na true.

volnoDama​(Policko zaciatocne, Policko konecne) - Tato metoda kontroluje ci sa dama moze presunut na urcite policko.

vyberSuradnice​(int x, int y) - Tato metoda umozuje presun figuriek po sachovnici, zaroven sa stara aj o striedanie hracov, kontrolovanie podmienok pri skakani a posune, vypise vyhercov, zmeny figuriek a ukoncenie hry.

vymazVdiagonale​(Policko zaciatocne, Policko konecne) - Tato metoda vymaze nepriatelske figurky, ktorymi presla dama, pri svojom pohybe.

**Trieda Main :**

Tato trieda obsahuje metodu main. Vytvori sa nam objekt hry a nasledne spusti.

**Trieda Policko :**

Trieda policko reprezentuje policko v sachovnici.

**Konštruktor :**

Policko​(Suradnice suradnice, int velkost) - Konstruktor inicializuje policka, jeho zakladny tvar, polohu, typ, farbu (cierna) a zaroven ho zobrazi na platne.

**Metódy :**

getObsadenost() - Getter, na zistenie obsadenosti policka.

getPoziciaX() - Getter, na zistenie pozicie x policka.

getPoziciaY() - Getter, na zistenie pozicie y policka.

obsahujeSuradnice​(int x, int y) - Metoda overuje ci su dane suradnice platne pre dane policko.

obsahujeSuradniceSachovnica​(int x, int y) - Metoda overuje ci su dane suradnice platne pre dane policko.

poziciaX() - Getter, na zistenie pozicie x policka.

poziciaY() - Getter, na zistenie pozicie y policka.

setJeObsadeny​(boolean obsadeny) - Setter, nastavuje obsadenost policka

zmenFarbuPolickaNaCervenu() - Tato metoda meni farbu policka na cervenu

**Trieda Sachovnica :**

Tato trieda reprezentuje sachovnicu. Deklarujem tu dva ArrayListy pre figurky a pole poli, ktore bude reprezentovat sachovnicu a finalne atributy, ktore reprezentuju vlastnosti sachovnice.

**Konštruktor :**

Sachovnica() - Konstruktor sachovnice, tu inicializujem ArrayListy, pole poli a volam metodu, ktora mi nastavy a vykresli sachovnicu a jej zakladny stav.

**Metódy :**

dajPolicko​(int x, int y) - Tato metoda sluzi na zistenie policka na danych suradniciach na sachovnici.

dajPolickoNaSuradniciach​(int x, int y) - Tato metoda sluzi na zistenie policka na danych suradniciach v pixeloch.

getCerveneFigurky() - Getter pre atribut, ktory obsahuje cervene figurky.

getCierneFigurky() - Getter pre atribut, ktory obsahuje cierne figurky.

getVelkostPolicka() - Getter, vdaka ktoremu sa ziska velkost policka.

jeObsadeneVOkoliNepriatelom​(Policko policko) - Tato metoda kontroluje volne a obsadene pozicie v okoli daneho policka, ktore je parametrom metody.

prehladajPoleFigurok​(String farba, Policko policko) - Metoda, ktora vrati figurku, ktora splna zadane parametre.

prehladajPoleFigurok​(Policko policko) - Metoda, ktora vrati figurku, ktora splna zadane parametre.

zmazFigurku​(ArrayList zoznamFiguriek, Figurka figurka) - Tato metoda zmaze figurku z konkretneho ArrayListu

zmazFigurku​(ArrayList zoznamFiguriek, Policko policko) - Tato metoda zmaze figurku z konkretneho ArrayListu, na danom policku.

zmenNaDamu() - Tato metoda prehladava oba ArrayListy a zistuje ci sa obycajna figurka nenachadza na poslednom riadku sachovnice protivnikovej farby.

**Trieda Suradnice :**

Trieda, ktora modeluje funkciu suradnic v priestore.

**Konštruktor :**

Suradnice​(int poziciaX, int poziciaY) -   
Kontstruktor novych suradnic, inicializuje atributy podla zadanych parametrov.

**Metody :**

getPoziciaX() - Getter pre atribut poziciaX.

getPoziciaY() - Getter pre atribut poziciaY.

**Enum TypFigurky :**

Enum, ktory reprezentuje typy figuriek a ich prislusne farby. Obsahuje 4 konštanty : CERVENA, CERVENA\_DAMA, CIERNA, CIERNA\_DAMA.

**Metody :**

getSubor() - Getter atributu subor.

toString() - Prekrytie metody toString, ktoru dedi kazdy objekt.