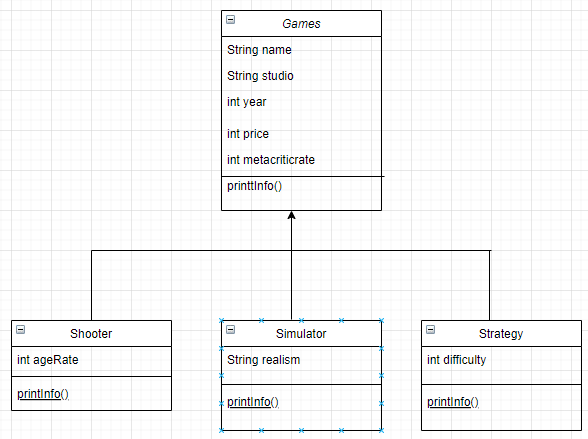
Uzdevuma apraksts

Uzņēmums “Kostyan Games” vēlas automatizēt videospēļu izvēli un pārdošanu savā tīmekļa vietnē.   Norādīt konkrētas spēles vispārīgo informāciju. Kā arī klasificēt spēles pēc žanriem.

Specifikācija



* Klase **Games** pieprasa videospēles nosaukumu, studiju, kas izdeva videospēli, izdošanas gadu, cenu un vērtējumu vietnē Metacritic.
* Klase **Shooter** manto visus klases Games laukus un metodes un pieprasa vecuma rekomendāciju.
* Klase **Simulator** manto visus klases Games laukus un metodes un pieprasa reālisma vērtējumu.
* Klase **Strategy** manto visus klases Games laukus un metodes un pieprasa sarežģītības vērtējumu.

Lauki:

Klase **Games**:

* String name (piemēram, STALKER)
* String studio (piemēram, GSC)
* Int year (piemēram, 2007)
* Int price (piemēram, 10)
* Int metacriticrate (piemēram, 87)

Klase **Shooter**:

* Int ageRate (piemēram, 16)

Klase **Simulator**:

* String realism (piemēram, regular)

Klase Strategy:

* Int difficulty (piemēram, 9)

Konstruktori:

* public Games(String name, String studio, int year, int price, int metacriticrate){

this.name = name;

this.studio = studio;

this.year = year;

this.price = price;

this.metacriticrate = metacriticrate;

}

* public Shooter(String name, String studio, int year, int price, int metacriticrate, int ageRate) {

super(name, studio, year, price, metacriticrate);

this.ageRate = ageRate;

}

* public Simulator (String name, String studio, int year, int price, int metacriticrate, String realism){

super(name, studio, year, price, metacriticrate);

this.realism = realism;

}

Metodes:

1. public String printInfo () {

return("Name of the shooter: " + getName() + "\nStudio: " + getStudio() + "\nYear: " + getYear() + "\nPrice: " + getPrice() + "\nRate on Metacritic:" + getMetacriticrate() + "\n \n");

}

//metode izvada vispārējo informāciju par spēli

1. public void writeFile() {

try {

FileWriter fw = new FileWriter("games.txt", true);

String str = printInfo();

fw.write(str);

fw.close();

} catch (IOException var3) {

throw new RuntimeException(var3);

}

}

//metode ieraksta failā informāciju par spēlēm, kas ir izvēlēti

1. public String printInfo () {

return("Name of the shooter: " + getName() + "\nStudio: " + getStudio() + "\nYear: " + getYear() + "\nPrice: " + getPrice() + "\nRate on Metacritic:" + getMetacriticrate() + "\nAge rate: " + this.ageRate + "\n \n");

}

//metode izvada informāciju par spēli “Shooter” žanrā

1. public String printInfo() {

return("Name of the Simulator: " + getName() + "\nStudio: " + getStudio() + "\nYear: " + getYear() + "\nPrice: " + getPrice() + "\nRate on Metacritic:" + getMetacriticrate() + "\nRealism rate: " + this.realism + "\n \n");

}

//metode izvada informāciju par spēli “Simulator” žanrā

1. public String printInfo() {

return("Name of the strategy: " + getName() + "\nStudio: " + getStudio() + "\nYear: " + getYear() + "\nPrice: " + getPrice() + "\nRate on Metacritic:" + getMetacriticrate() + "\nDifficulty: " + this.difficulty + "\n \n");

}

//metode izvada informāciju par spēli “Strategy” žanrā

Programma piedāvā:

Programma piedāvā izvēlēties videospēles žanru, kurā lietotājs grib nopirkt spēli, pēc žanra izvēles, programma piedāvā izvēlēties videospēli izvelētā žanrā un pēc izvēles izvada informāciju konsolē un pieraksta to failā (videospēle ir veiksmīgi nopirkta).

Padziļināti:

Datu saglabāšana games.txt failā.

Secinājumi:

Ir izdevusies izveidot programmu ar objektorientētu programmēšanu, kas ļauj izpildīt visus projekta mērķus ar informācijas par nopirkto videospēli pieraksti failā. Klients var izvēlēties videospēles žanru un pēc tam pašu videospēli no piedāvātiem. Tomēr šo projektu var uzlabot, pievienojot vairāk žanru un videospēļu variantus katram žanram un oievienojot iespēju apskatīt recenzijus par videospēlēm vai ievietojot savu atziņu.

Pielikums: https://github.com/Patina-89/oop