VILNIAUS MYKOLO BIRŽIŠKOS GIMNAZIJA

**MARTYNAS ANTIPENKOVAS**

**III d klasė**

**„NUMERUOTOS FIGŪROS“**

Informacinės technologijos

Kūrybinis projektinis darbas

Darbo vadovė

Rima Šiaulienė

Mokytoja ekspertė

Vilnius, 2017

[**I SKYRIUS** 3](#_Toc475379951)

[TERMINŲ ŽODYNAS 3](#_Toc475379952)

[ĮVADAS 4](#_Toc475379953)

[Temos aktualumas 4](#_Toc475379954)

[PD tikslas ir uždaviniai 4](#_Toc475379955)

[Naudoti metodai 4](#_Toc475379956)

[Darbo struktūra 4](#_Toc475379957)

[Suteikta pagalba 5](#_Toc475379958)

[**II SKYRIUS** 6](#_Toc475379959)

[DĖSTYMAS 6](#_Toc475379960)

[Pradinė versija 6](#_Toc475379961)

[Antroji versija 7](#_Toc475379962)

[Aplikacija 7](#_Toc475379963)

[Duomenų bazė 8](#_Toc475379964)

[Išvados 9](#_Toc475379965)

[**SKYRIUS III** 10](#_Toc475379966)

[INFORMACIJOS ŠALTINIAI 10](#_Toc475379967)

[**SKYRIUS IV** 11](#_Toc475379968)

[Priedai 11](#_Toc475379969)

# **I SKYRIUS**

# TERMINŲ ŽODYNAS

*Klasė* – konstrukcija, kuri leidžia sukurti specialius duomenų tipus, grupuojant kitus tipus, metodus ar įvykius. Su nestatinėmis klasėmis galima sukurti objektus.

*Objektas* – dinamiškai sukurtas klasės kintamųjų, metodų rinkinys.

*C#*  – objektinė programavimo kalba, sukurta Microsoft kompanijoje kaip dalis .NET iniciatyvos. Kalba paremta C++ bei Java kalbomis. Kalba kurta balansuojant tarp galingumo (C++ įtaka) bei greito programavimo (Java, Visual Basic įtaka).

*SQL* (*akronimas angl. k. „Structured Query Language“* ) – programavimo kalba, skirta bendravimui (duomenų tvarkymui, perkėlimui, išsaugojimui) su duomenų baze.

*IDE* (*akronimas* *angl. k. „Integrated Decelopment Environment“* ) – programa skirta programavimui, kuri turi kodo kompiliatorių, kodo tvarkyklę, įvairių įrankių, kurių galėtų prireikti programuotojui.

*WFA* (*akronimas angl. k. „Windows form application“*) – grafinė klasių biblioteka, .NET bibliotekos dalis. Ši klasių biblioteka yra dažniausiai naudojama programų, skirtų kompiuteriams su Windows operacinėmis sistemomis programų kūrimui.

*Konstruktorius* – specialus objekto metodas, naudojamas naujo objekto sukūrimui. Konstruktorius gali turėti parametrus, kurie panaudojami pradinei kuriamo objekto būsenai nustatyti.

# ĮVADAS

## Temos aktualumas

Man ši tema yra aktuali, nes norėjau patikrinti savo žinias, užpildyti spragas bei sukaupti naujų žinių apie WFA tipo aplikacijas. Taip pat norėjau sukurti programą, kuri savo forma būtų panaši į kitas komercines programas.

Pasirinkta žaidimo tema yra aktuali įvairaus amžiaus žmonėms, kurie norėtų praleisti laiką žaisdami loginį žaidimą ir bandydami surinkti kuo daugiau taškų.

## PD tikslas ir uždaviniai

Sukurti programą-žaidimą bei duomenų bazę, kurioje bus išsaugomi taškai

Siekdamas savo tikslo, darbą suskirsčiau į uždavinius:

1. Programos architektūros planavimas.
2. Duomenų įvesties lango tipo programos kodo rašymas.
3. WFA tipo programos kodo rašymas.
4. Programos susiejimas su duomenų baze.

## Naudoti metodai

Programos kodas parašytas C# kalba, pasitelkus *„Visual Studio 2015“* IDE platformą, bei SQL kalba, panaudojus *„SQL Server 2014 Managment Studio“* IDE platformą.

## Darbo struktūra

Pradinę žaidimo-programos versiją sudaro tik duomenų įvesties langas, kuriame yra pateikiama informacija bei žaidėjo galimybės. Tam, kad įvyktų veiksmas, kurio pageidauja naudotojas, jam reikia įvesti skaičių, kuris atitinka vieną iš veiksmų ir paspausti „*Enter*“ mygtuką.

Antrąją žaidimo-programos versiją sudaro daug WFA langų, kuriuose galima atlikti pasirinktinus veiksmus, pavyzdžiui: žaidimo lange galima pasirinkti iš kelių pasirinkimų tokių, kaip: traukti naują figūrą, didinti statomų taškų kiekį, užbaigti raundą ir kt. Žinoma, yra ir tokių langų, kuriose nėra veiksmo pasirinkimo ir iš žaidėjo yra reikalaujama atlikti tik tam tikrą veiksmą, pavyzdžiui: lange, kuriame reikia įvesti žaidėjo vardą galima tiktai tai ir padaryti, ko reikalaujama (įvesti vardą) ir nieko daugiau. Taip pat antrojo darbo versijoje sukurta sąsaja su duomenų baze, kurioje yra įrašyti žaidėjų rezultatai.

## Suteikta pagalba

Pagalbą suteikė programuotojas, duomenų bazių ekspertas Dionizas Antipenkovas. Taip pat mano bendraklasis Julius Baranauskas pasidalino savo mintimis. Jos paskatino atlikti kelis pakeitimus.

# **II SKYRIUS**

# DĖSTYMAS

Kaip ir nurodyta įžangoje visas mano *PD* buvo padalintas į dvi dalis – pradinę ir galutinę. Taigi šios dvi dalys bus nagrinėjamos atskirai, dėl savo skirtingo funkcionalumo.

## Pradinė versija

Pradinės versijos sukūrimo priežastis – kuo greičiau sukurti veikiantį programos kodą, jį pratestuoti bei, jei reikia, koreguoti. Duomenų įvesties lango tipo programa buvo pasirinkta, nes šis tipas turi metodus, kuriuos panaudojus yra kur kas lengviau rasti ir pataisyti iškilusias problemas.

Prieš rašant pačią programą buvo sunku numatyti informacijos šaltinius, kuriais teks naudotis, nes informacijos šaltinių poreikis atsiranda susidūrus su problema. Taip pat daug informacijos buvo ieškota ir rasta asmeniniuose failuose bei projektuose.

Žaidimas prasideda nuo paprasto klausimo „Ar norite pradėti žaidimą? t/n“. Atsakius į klausimą „t“ arba „T“ buvo aktyvuotas (*if* sąlygos sakiniu) *while* ciklas (*žiūrėti priedą 1 Pav.*). Vėliau buvo sukurti 3 atskiri klasių failai *Figura.cs*, *Bankas.cs* ir *Player.cs* (*žiūrėti priedus 3 Pav., 2 Pav., 4 Pav.*). Šiuose klasės failuose buvo atskirai aprašyti figūros, banko ir žaidėjo objektai. Pagrindiniame faile (*Program.cs*) buvo pridėtos papildomos kodo eilutės, kurios paprašė įvesti žaidėjo vardą ir gautą atsakymą priskyrė prie *zaidejas* objekto, kuris buvo inicializuotas su *Player* klase. Tada buvo sukurtas *bank* objektas inicializuojant *Bankas* klasę. Objektams *bank* ir *zaidejas* suteikiami taškai per parametrą *BendriTaskai.* Šie taškai panaudoti ne tik kaip netikra valiuta statymams, bet kaip ir vienintelis rodiklis, kuris parodo žaidėjo progresą žaidime, t. y. kuo daugiau bendrų taškų žaidėjas turi, tuo jis yra arčiau absoliučios pergalės. Po to buvo sukurtas naujas klasės failas *FigurosKomandos.cs,* kuriame buvo aprašytas statinis metodas atsitiktinai sugeneruojantis figūrą. Vėliau programa buvo paildyta *while* ciklu, kuriame buvo objektams *zaidejas* ir *bank* priskiriama po dvi figūras, kurios sukurtos panaudojus statinį metodą *Generuoti* iš klasės *NaujaFigura.* Tada sukurtas *UserInterface.cs* klasės failas, kuriame buvo aprašytas naudotojo bendravimas su programa, t. y. programa sugeba reaguoti į vartotojo įvestus skaičius, kurie atitinka tam tikrą komandą. Vėliau padėta kurti bendrų taškų palyginimo sistema, kuri patikrina ar žaidėjas gali tęsti žaidimą. Bet ši sistema nebuvo pabaigta, nes buvo pereita į antrąją darbo fazę.

## Antroji versija

### Aplikacija

Antrasis programos etapas buvo pradėtas sukūrus atskirą projektą, skirtą WAF aplikacijų kūrimui. Pirmasis buvo sukurtas pradinis langas *PradinisLaukas.cs,* kuriame galima pasirinkti *Pradėti naują žaidimą, Top 10 žaidėjų, Taisyklės, Išeiti.* Taip pat sukurtas langas *NewPlayer.cs*, kuriame vartotojo prašoma įvesti žaidėjo vardą. Paspaudus mygtuką *Tęsti* sukuriami objektai *bank* bei *player*, kurio parametrui *Name* yra priskiriamas atsakymas, kuris buvo įvestas (*žiūrėti priedą 5 Pav.*). Minėtas langas užsidaro ir atsidaro kitas langas, kuriame yra išspausdinti abiejų objektų *player* ir *bank* bendri taškai ir paprašoma žaidėjo įvesti statomų taškų sumą, kuri, paspaudus mygtuką *Tęsti* yra patikrinama (patikrinama ar įvesta suma neviršija abiejų objektų turimų bendrų taškų sumos ir ar tai yra skaičius, t. y. ar nėra įvestyje raidžių, simbolių) ir priskiriama į kintamąjį, skirtą saugoti bendrus taškų statymus (*žiūrėti priedą 6 Pav.*). Vėliau buvo sukurtas pagrindinis langas, kuriame vyksta žaidimas (*žiūrėti priedą 7 Pav.*). Visų pirma šiam langui buvo aprašyti lango inicijavimo metu atliekami veiksmai. Atidarius langą ir patikrinus ar šis langas nebuvo jau atidarytas (minėtas patikrinimas buvo pridėtas vėliau, kai buvo atrasta programos klaida, kuri naudotojui priskirdavo po dvi naujas figūras, kai langas buvo sukuriamas iš naujo, tiksliau, kai žaidėjas grįždavo iš kito (*Didinti taškų statymus*) lango) žaidėjo bei banko objektams buvo priskiriama po dvi naujas figūras. Pirmojoje versijoje objektams priskiriant naujas figūras iškilo problema, apie kurią nepavyko rasti informacijos internete – visus priskyrimo kartus buvo panaudojama ta pati figūra, nors metode *Generuoti* yra aprašyta tokių klaidų prevencija. Antrojoje darbo versijoje ši problema buvo išspręsta tarp kiekvieno priskyrimo padarant 250 milisekundžių pauzę pasitelkus *System.Threading* bibliotekoje esančiu metodu *Thread.Sleep(250);*. Po figūrų priskyrimo etiketėms, kurios yra lange (*angl. k. labels*), buvo suteikto *string* tipo reikšmės, kuriose buvo įrašyti žaidėjo ir banko turimi taškai bei turimos figūros (*žiūrėti priedą 8 Pav.*). Vėliau buvo sukurtas *StatymuLangas.cs* langas, kuriame galima padidinti statomų taškų kiekį. Šiame lange buvo panaudotas toks pats įvesties patikrinimas (*žiūrėti* *priedą 6 Pav.*), kuris buvo panaudotas prieš tai. Šis langas (kaip ir kiti, pradedant žaidėjo sukūrimo langu) perėmė objektų *player* ir *bank* reikšmes per savo konstruktorių. Vėliau buvo apibrėžtas *ZaidimoLaukas.cs* lange esančio *Baigti raundą* mygtuko funkcionalumas. Paspaudus mygtuką pagal paprastą palyginimo logiką yra patikrinama, kas yra raundo nugalėtojas. Raundo nugalėtojui yra priskiriami visi taškai buvę *statymas* kintamajame, o jei raundas pasibaigia lygiosiomis, tai taškai esantys *statymas* kintamajame yra padalinami pusiau. Pasibaigus raundui ir nustačius laimėtoją yra patikrinama kiek žaidėjas (objektas *player*) turi bendrų taškų. Jei žaidėjas nebeturi taškų, tai jam yra pasiūloma pradėti iš naujo. Jei žaidėjas turi dar taškų, bet nėra pasiekęs taškų maksimumo, tai jam yra pasiūloma tęsti žaidimą arba užbaigti ir savo rezultatą įrašyti į duomenų bazę. Jei žaidėjas surinko maksimalų taškų skaičių (1 000 000 000), tai jis pamatys pasveikinimo lentelę ir jo rezultatas bus įrašytas į duomenų bazę. Taip pat programoje yra pasikartojantys mygtukai, tokie kaip *Išeiti* ir *Taisyklės.* Paspaudus mygtuką *Išeiti* programa uždaroma panaudojus eilutę *Application.Exit();*, o langas su taisyklėmis aktyvuojamas panaudojus *MessageBox.Show();* ir tarp skliaustų įrašius taisykles.

### Duomenų bazė

Norint išsaugoti rezultatus antrajame projekto etape buvo panaudota duomenų bazė, pavadinimu *PD\_NugaletojuSarasas*. Joje yra lentelė, kurioje yra išsaugomi žaidėjo vardas, taškų suma bei data, kada buvo sukurtas įrašas (*žiūrėti priedą 9 Pav.*). Kiekvienam lentelės įrašui yra priskiriamas unikalus *id* (tai užtikrina atributas *primary key*, kuris neleidžia pasikartoti vienodoms reikšmėms).Vėliau buvo aprašytas pradinio lango *Top 10 žaidėjų* mygtuko funkcionalumas. Jį paspaudus atsiranda naujas langas, kurio sukūrimo metu yra susisiekiama su duomenų baze ir gaunami iš jos įrašai. Jie yra sujungiami į vieną *string* tipo kintamąjį. Jis yra priskiriamas etiketei, esančiai tame lange (*žiūrėti priedą 10 Pav.*). Vėliau buvo aprašyti mygtukai, kurių paskirtis buvo įrašyti rezultatus.

# Išvados

Darant projektinį darbą pavyko:

* Pilnai sukurti antrąją programos versiją
* Susieti programą su duomenų baze
* Iškilusioms problemoms

Projektinio darbo metu nepavyko

* Galutinai pabaigti pirmosios programos versijos
* Panaudoti *DataGridWiev* grafinio objekto

# **SKYRIUS III**

# INFORMACIJOS ŠALTINIAI

* + Straipsnis Nicole Avery (2012) „*21 card game“* prieiga per internetą :

<http://planningwithkids.com/2012/06/20/21-card-game/> [Žiūrėta 2016-12-18]

* + [Microsoft korporacijos programavimo kodo žinynas „Thread.Sleep Method (Int32)“](https://msdn.microsoft.com/en-us/library/d00bd51t%28v=vs.110%29.aspx?f=255&MSPPError=-2147217396)

https://msdn.microsoft.com/en-us/library/d00bd51t%28v=vs.110%29.aspx?f=255&MSPPError=-2147217396 [Žiūrėta 2017-02-01]

* + Programavimo kodo žinynas, straipsnis „SqlConnection“

[https://www.dotnetperls.com/sqlconnection](%20https://www.dotnetperls.com/sqlconnection) [Žiūrėta 2017-01-28]

* Filmuota medžiaga VoidRealms (2011) „C# 20 - First Windows Forms Application“

<https://www.youtube.com/watch?v=Oxg6ciIcO3U> [Žiūrėta 2017-01-15]

* + Filmuota medžiaga Fredrik Hoffmann (2010) „C# Tutorial 44 - Transferring data between forms“ <https://www.youtube.com/watch?v=QgvAwTV3jmM> [Žiūrėta 2017-01-23]
  + Filmuota medžiaga winforms (2014) „open new form, close existing windows form c#”

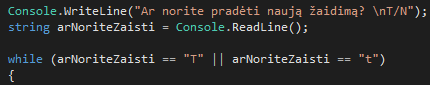
<https://www.youtube.com/watch?v=T06aHrAqfRw> [Žiūrėta 2017-02-04]

* + SQL programavimo kalbos žinynas „SQL GROUP BY Statement“

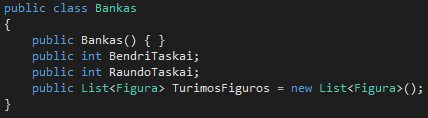
<https://www.w3schools.com/sql/sql_groupby.asp> [Žiūrėta 2017-02-10]

# **SKYRIUS IV**

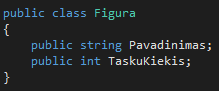
# Priedai



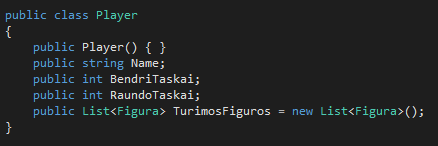
1 Pav. Pradinis ciklas. Šaltinis: Sudaryta autoriaus



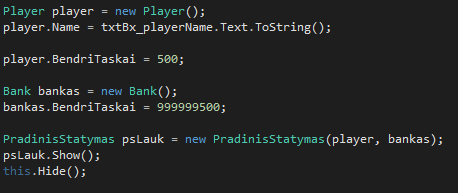
2 Pav. Klasės Bankas aprašymas. Šaltinis: Sudaryta autoriaus



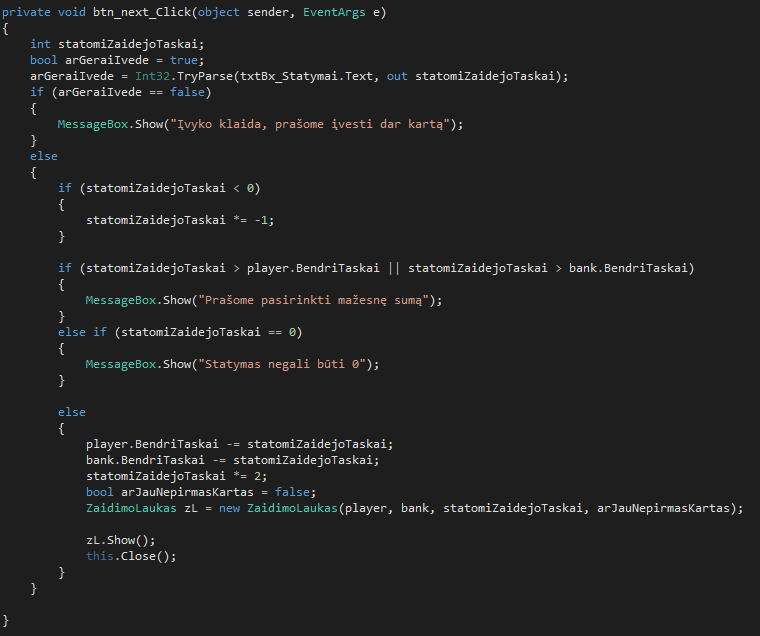
3 Pav. Klasės Figura aprašymas. Šaltinis: Sudaryta autoriaus



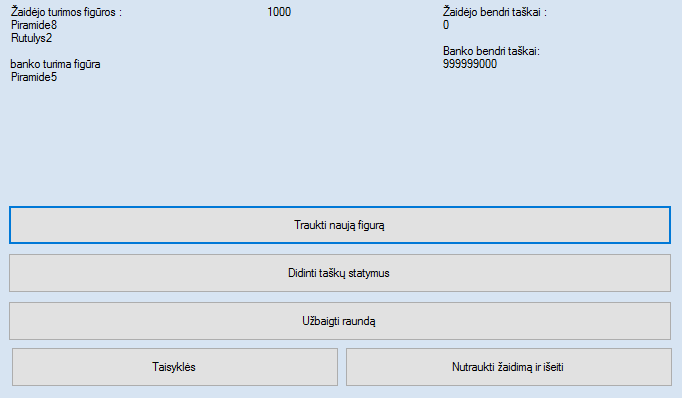
4 Pav. Klasės Player aprašymas. Šaltinis: Sudaryta autoriaus



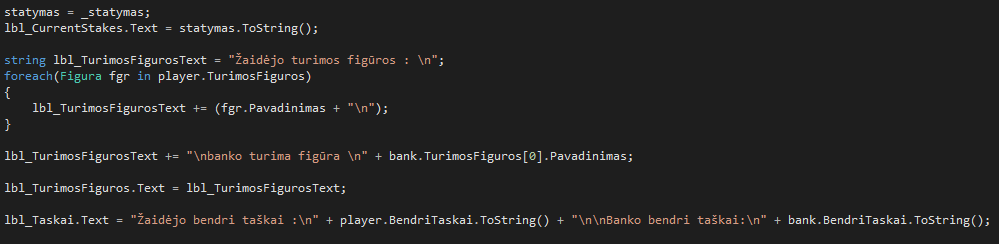
5 Pav. Objektų player ir bank sukurimas. Šaltinis: Sudaryta autoriaus



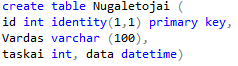
6 Pav. Įvesties patikrinimas. Šaltinis: Sudaryta autoriaus



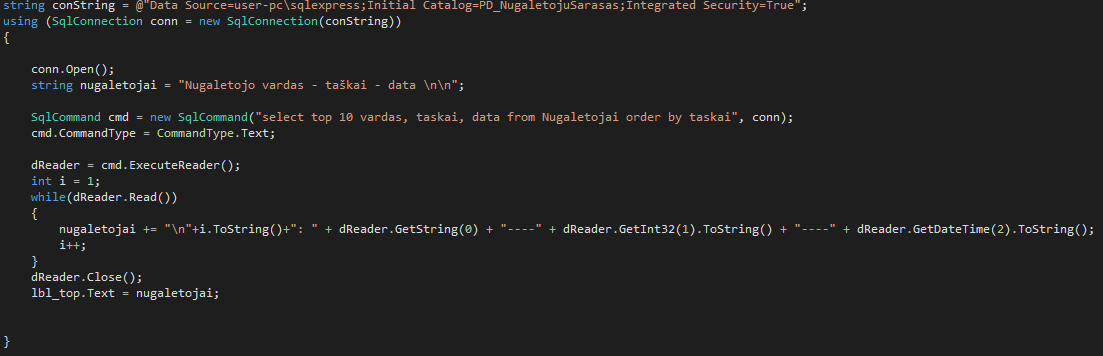
7 Pav. Žaidimo langas. Šaltinis: Sudaryta autoriaus



8 Pav. Žaidimo lango etikečių turinio nustatymas. Šaltinis: Sudaryta autoriaus



9 Pav. Nugaletojai lentelės sukurima. Šaltinis: Sudaryta autoriaus s



10 Pav. Susiekimas su duomenų baze ir rezultatų priėmimas iš jos. Šaltinis: Sudaryta autoriaus



11 Pav. Įrašo įrašymas į duomenų bazę. Šaltinis: Sudaryta autoriaus