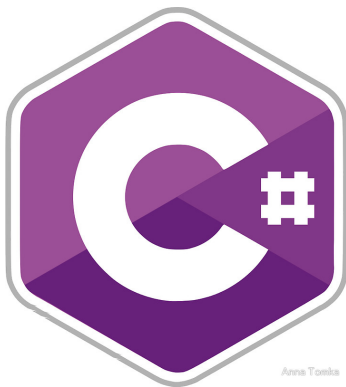




VILNIUSCODINGSCHOOL

# Programavimas C#



Anna Tomka

# Apie mane



- Raimundas Banevicius (raima156@yahoo.com)
- Išsilavinimas: bakalauras, Informatika, Vilniaus Universitetas.
- Darbo patirtis:
  - 16 metu patirties su ActionScript3; (Flash)
  - 1 metu patirtis su c++;
  - 1 mtu patirtis su C#, Unity3d.



mvcExpress

# Klasès

Classes

# Step 1

Kuriam zaidima! Pradedam nuo herojaus:

- sukurti klase Hero, x,y,name, MoveRight(), MoveLeft(), PrintInfo()
- Sukurti Hero objekta >> printInfo >> moveRight >> printInfo.

Prikuriam priesu!

- sukurti klase Enemy, id,x,y,name, MoveDown(), PrintInfo()
- Sukurti 10 List<Enemy> su random x ir y, unikaliais id, Visus PrintInfo >> MoveDown >> visus PrintInfo

# Step 1 solution

- <https://github.com/MindScriptAct/c-consoleGameExercisise/blob/47c4376a9e09d6902179f203acb2cd51b7f6d1cc/ConsoleGame/Program.cs>

# Step 2

Sukuriam nauja klase zaidimui paciam:

- `sukurti klase GameScreen, width,height; Sukurti objekta myGame;`

Sekantis zingsnis - ikeliam heroju ir priesus į kita nauja klase:

- `GameScreen` sukurti metoda: `void SetHero(Hero hero)`, panaudojam ja kad nusiusti musu heroju į `myGame`.
- `GameScreen` sukuriam metoda: `Hero GetHero()`;
- `GameScreen` sukuraim metoda: `void AddEnemy(Enemy enemy)`, panaudojam ja kad nusiusti visus priesus ten.
- `GameScreen` sukuraim metoda: `void MoveAllEnemiesDown()`,
- `GameScreen` sukuraim metoda: `Enemy getEnemyById(int id)`, (`Enemy` sukuriam metoda `int GetId()`)

Gera! butu atspausdinti musu programos busena:

- `GameScreen` sukuraim metoda: `void Render()`; Kurioje panaudojam herojaus ir visu priesu `PrintInfo`.

Pabandom viska panaudoti ka padarem:

- `Render()` >> heroju pajudinam į kaire >> visus priesus zemyn >> antra priesa zamyn >> `Render`

# Step 2 solution

- <https://github.com/MindScriptAct/c-consoleGameExercisise/blob/d2e794545e03803732bc0b362160ff206f11332b/ConsoleGame/Program.cs>

# STOP

Reik ismukti apie inheritance...



# Step 3

Panaudojam inheritance, sujungiam Hero, ir Enemy funkcijas kurios tos pacios:

- Sukurti nauja klase Unit: `x,y, name; PrintInfo();`

Pakeiciam Hero ir Enemy kad extendintu Unit:

- Hero ir Enemy extend Unit.
- Pakeisti Hero ir Enemy konstruktorius, kad naudotu `base()` konstruktoriu.
- Programa turetu veikti kaip anksciau po pakeitimu.

# Step 3 solution

- <https://github.com/MindScriptAct/c-consoleGameExercisise/blob/34241f8fd946d61f3693e344e9b68da441af9a42/ConsoleGame/Program.cs>

# Step 4

Zaidimuose ekranas atnaujinamas pastoviai... pridedam Game loop!:

- Main funkcijos gale sukurti cikla:
- Idedam į ji Consoles išvalymą;
- Idedam į ji myGame.Render();

```
do {  
    } while (needToRender);
```

Zaidimuose mes galime skaityti klavisu paspaudimus, ir keisti busena:

- Ciklo viduje, tol kol konsole turi nauju paspaudimu, juos perskaitykite.
- Jei paspaustas ESC – užbaigti zaidimą.
- Jei paspaustas Left, Right arrows – judinti heroju į Left, right.

Kad ekranas kazka paroditu, mums reikia sustabdyti programos vygdima :

- Panaudokit ciklo gale : `System.Threading.Thread.Sleep(250);`

(Hint 1 : [Google](#) )

(Hint 2 : next slide...)

# Step 4 - hints

```
Console.Clear();
```

```
while (Console.KeyAvailable) {  
    ConsoleKeyInfo pressedChar = Console.ReadKey(true);  
    switch (pressedChar.Key) {  
        case ConsoleKey.Escape:  
            break;  
        case ConsoleKey.RightArrow:  
            break;  
        case ConsoleKey.LeftArrow:  
            break;  
    }  
}
```

ARBA

```
while (Console.KeyAvailable) {  
    ConsoleKeyInfo pressedChar = Console.ReadKey(true);  
    int hashCode = pressedChar.Key.GetHashCode();  
  
    switch (hashCode) {  
        case 27: //ConsoleKey.Escape:  
            break;  
        case 39: // ConsoleKey.RightArrow:  
            break;  
        case 37: // ConsoleKey.LeftArrow:  
            break;  
    }  
}
```

# Step 4 solution

- <https://github.com/MindScriptAct/c-consoleGameExercisise/blob/f0b45c23e976c338a3bab907c85a6fafa942dd50/ConsoleGame/Program.cs>

# Step 5

Atejo bosas, ir sako kad reikia mesti viska, ir susikaupti ties GUI!

Musu pagrindineme faile labai jau daug visko prideta, persitvarkom!

- Sukuriam naujus failus: `Game.cs`, `GameGui.cs`
- Viska iskyrus `Main()` metodo klase perkeliame i `Game.cs`.
- Isitikinam kad programa veikia kaip anksciau.
- Viska is main funkcijos perkelti i **`static void StartGame()`** funkcija.  
(Musu `Main()` lieka tuscias mums testuoti GUI.)

# Step 5 solution

- <https://github.com/MindScriptAct/c-consoleGameExercisise/tree/8d41ac1e50a50e0020b1cf957fb077a8587a5ddd/ConsoleGame>

## Step 6

- Ataina dizaineris su popieriuku... sako.. noriu kad atrodytu taip:  
Menu langas su remeliu, su 3 migtukais, ir titulinio tekstu!

[illegible]



## Step 6

- ... ir taip atrodo credit langas uzmetas ant menu lango:  
kureju vardai ir back mygtukas, lange su remeliu.



## Uzduotis:

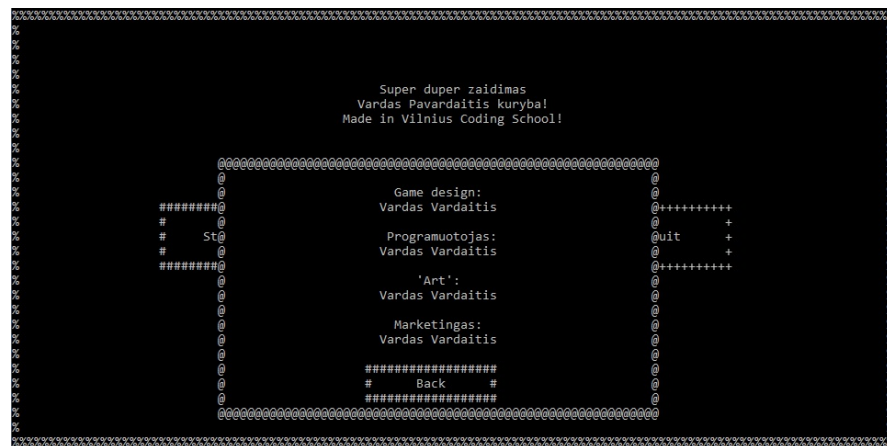
- Paimiti lapa popieriaus, ir nupiesti visu reikalingu klasių diagrama. Pasistenkite sukurti >5 klases

(Pagalvokit kokios klases extendina  
kitas klases, kokios klases naudoja  
kitas klases, )

- Musu tikslas kad Main programa atrodytu taip:

```
static void Main() {
    Console.CursorVisible = false;
    GameWindow gameWindow = new GameWindow();
    CreditWindow creditWindow = new CreditWindow();
    Console.ReadKey();
}
```

## Step 6 – hints 1



- Window
- Button
- TextBlock

## Step 6 – hints 2



- Window = Frame
- Button = Frame \* 2 + TextLine
- TextBlock = X \* TextLine
- Taip pat gerai butu tureti bendra klase visiem bendriems GUI duomenims laikyti:  
x,y,width,height,Render()

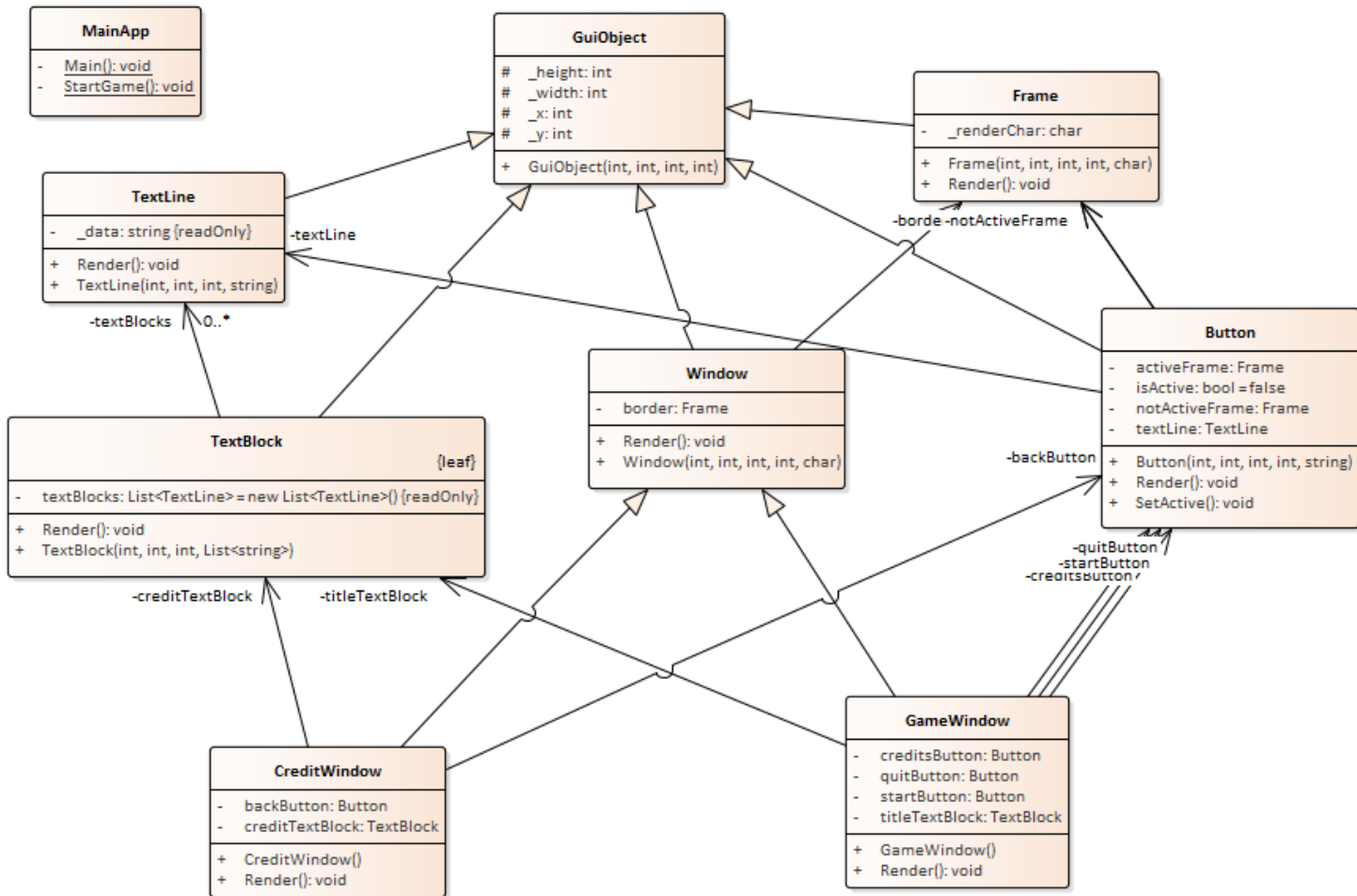
## Step 6 – answer

Kitame puslapyje etsakymas!  
(ne.. as nepaisiau jo pats... yra tam programos...)

...ready?

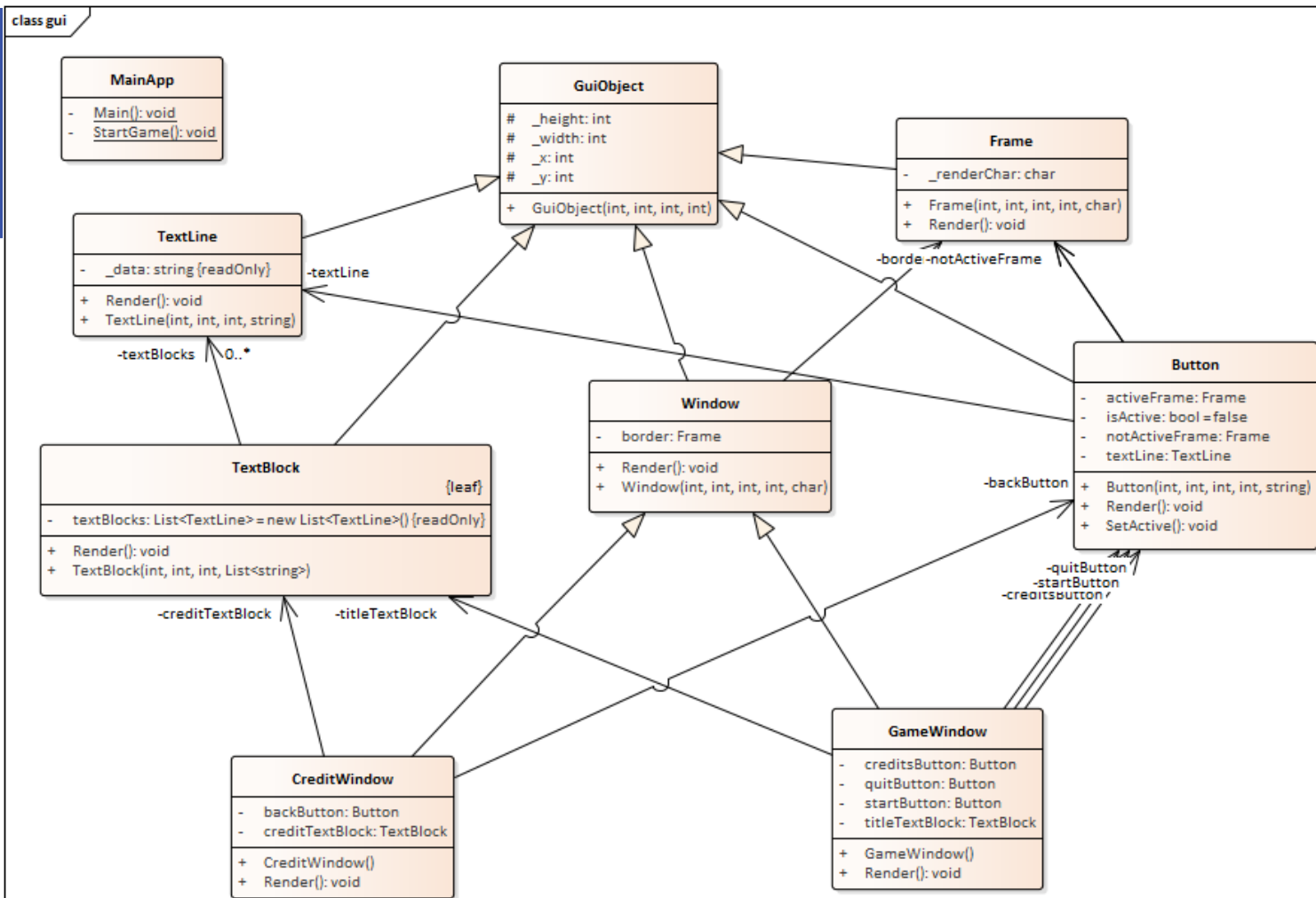
# Step 6 diagram:

class gui



# Step 7 implement:

- Sukuriam visas klases ir ju sasajas, pagal sita diagrama.
- Palikite tuscia implementacija.



# Step 8 implement:



- Sukurkit visas klasiu implementacijas.
- Pratestuokite:

```
static void Main() {
    Console.CursorVisible = false;
    new GameWindow();
    new CreditWindow();
    Console.ReadKey();
}
```



# Step 8 solution

- <https://github.com/MindScriptAct/c-consoleGameExercisise/tree/2c6845e56bffa8f3c5cf6fca72d21d9ae749b7291/ConsoleGame>



# Step 9

- **Ismokti teorija : override, virtual, abstract, sealed!**
- Suraskime gera vieta virtual Render() funkcijai.
- Suraskime klases/metodus kuriuos galima pazimeti abstract.
- Suraskime klases/metodus kuriuos galima pazimeti sealed.

# Step 9 solution

- <https://github.com/MindScriptAct/c-consoleGameExercisise/commit/8a2b02fc00240ca479fa1f79952d3110c39d3198>

# Step 10 - WIP

- Sukurkite `IRenderable` visom klasem su `Render()` funkcijom.
- Sukurkite interfaisa button funkcijom `ISelectable`.
- Visiem objektam sukurkite nauja **`privat List<IRenderable> components`**
- Visiem objektam sukurkite metoda **`private void AddComponent(IRenderable component)`**
- Pakeiskite savo objektus kad jie naudotu sita funkcija elementam prideti.

# Step 10 solution

- ...

# Step 11 - WIP

- Sukuriam atskira klase : GuiController – jos darbas bus leisti vartotokui vaikscioti per menu, startuoti musu programa, iseiti is programas.

# Step 11 solution

- ...

# Step 12 - WIP

- Gryztam prie zaidimo... ismokstam static funkcijas.
- Sukuriam SuperConsole.WriteAt(5,5, "tekstas")
- Surankiotam visas static constantas.

# Step 12 solution

- ...



# Step 13 - WIP

- Pabaigiam zaidima – snakes.

# Step 13 solution

- ...