

PROGRAMOVÁNÍ

1	SPŠ Chomutov	Beránek
18.10.2022	Šibenice v OOP	V4

Zadání:

Pomocí objektově orientovaného programování vytvořte hru šibenice.

- Program načítá dostupná slova ze souboru (např. .csv)
- Slovo je vybíráno náhodně

Teorie:

Objektově orientované programování se zaměřuje více na to, jak program vidí člověk spíše než stroj. Využívá práce s objekty, které mají vlastní atributy a metody. Protože je všechno uspořádané pod objekty je kód přehlednější, a hlavně čitelnější, na první pohled je mnohem jasnější co má program dělat. Objekty se vytváří ze tříd. Ty slouží jako jakési šablony pro vytváření objektů. Z jedné třídy můžeme vytvořit několik objektů, které budou sdílet stejné atributy a metody, ale budou mít jiná data.

Popis programu:

Při spuštění aplikace se načtou slova ze souboru „src.csv“ a zapíšou se do listu „words“, zároveň jsou převedena na velká písmena (omezuje komplikace se stylem zapsání slov v souboru). Následně se vygeneruje náhodné číslo, pomocí kterého se vybere slovo z listu. To je rozděleno na jednotlivé charaktery a ty jsou postupně zapsány do listu „letters“. Ve chvíli kdy uživatel stiskne klávesu, klávesa je převedena na charakter a převedena na velké písmeno. Proběhne kontrola, jestli uživatel již toto písmeno nezkoušel hádat. Pokud ano, program dále nepokračuje, tedy neudělá nic. Pokud ne, písmeno se zapíše do listu „guesses“, který se následně setřídí, a poté se všechny objekty v listu vypíšou do labelu (aby uživatel viděl která písmena již hádal). Program poté pokračuje k vyhodnocení tipu. Program porovná uživatelův tip s písmeny v listu „letters“, pokud najde shodu, písmeno se „odhalí“ v příslušném labelu, a připíše se bod. Pokud program shodu nenajde, podle počtu chyb co doposud uživatel udělal zobrazí příslušný obrázek šibenice a připíše počet chyb. Pokud se načte poslední obrázek šibenice, nebo pokud uživatel dosáhne 5 bodů, hra se ukončí, a program zaznamená výhru nebo prohru. V tomto případě se uživateli zobrazí messagebox, který buď oznámí výhru nebo prohru, a zeptá se uživatele jestli chce hrát dál. Pokud uživatel zvolí možnost ano, vyčistí se všechny aktuální statistiky, zvolí se nové náhodné slovo a uživatel hraje znovu. Pokud zvolí možnost ne, program se ukončí.

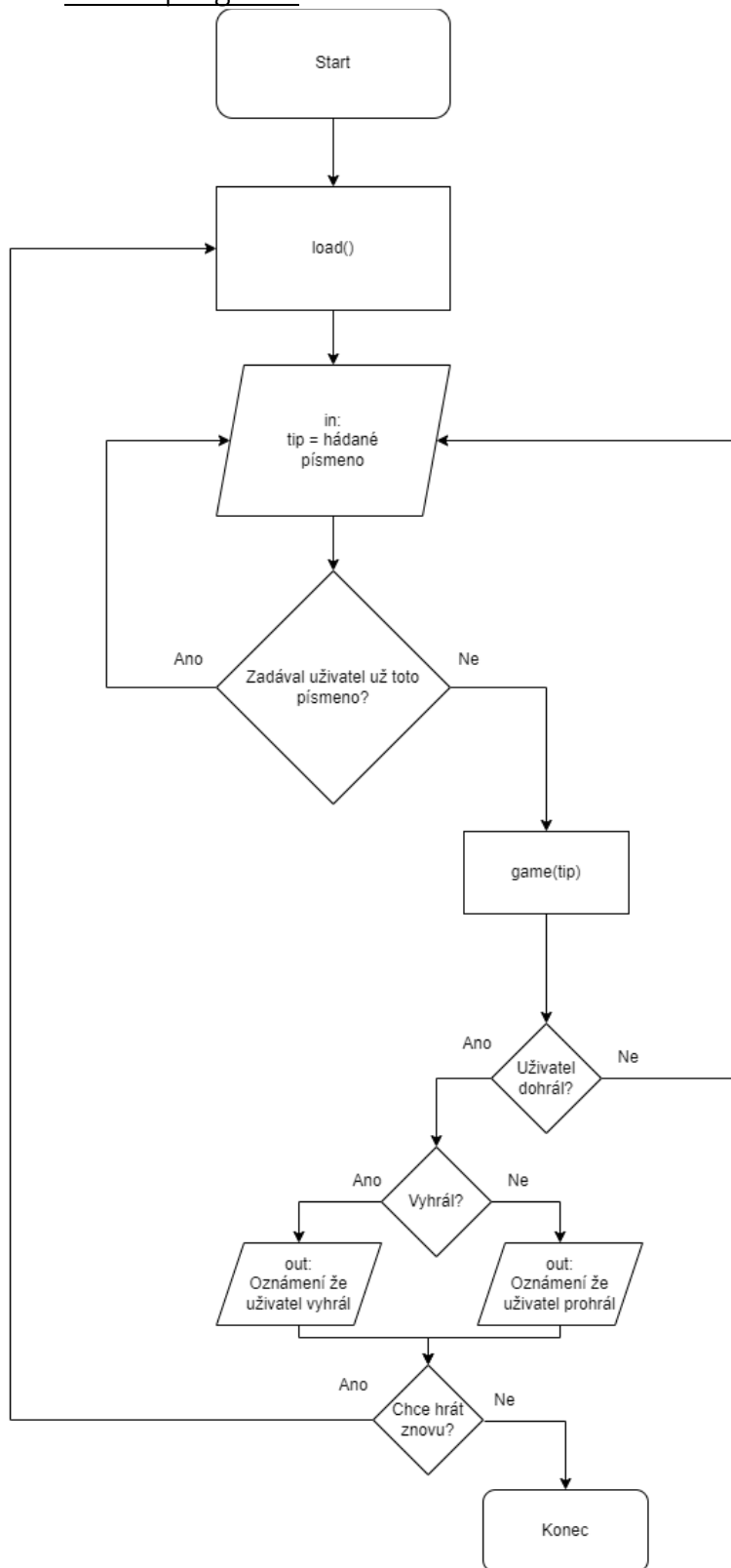
Seznam funkcí a proměnných:

Typ	Proměnná	Účel
Class Form1		
char	guess	Hádaný znak
hangmanclass	hang	Instance třídy
DialogResult	result	Rozhodnutí uživatele, jestli chce hrát další hru
Class hangmanclass		
List<string>	words	Seznam možných slov
List<char>	letters	Písmena vybraného slova
List<char>	guesses	Hádaná písmena
List<Control>	labels	Labely pro ukazování písmen
int	error	Počet chyb
int	points	Počet uhodnutých písmenm
string	word	Hádané slovo
string	a	String hádaných písmen, který se následně vypíše do labelu
string	line	Řádek načítaného souboru
bool	endgame	Určuje, jestli uživatel dohrál
bool	winresult	Určuje, jestli uživatel vyhrál
Control	used	Label hádaných písmen
Control	pictureBox1	Obrázek šibenice
Control[]	label	Nalezený label
DialogResult	result	Rozhodnutí uživatele, jestli chce hrát další hru
StreamReader	fload	Načítání souboru
Random	rnd	Generace náhodného čísla

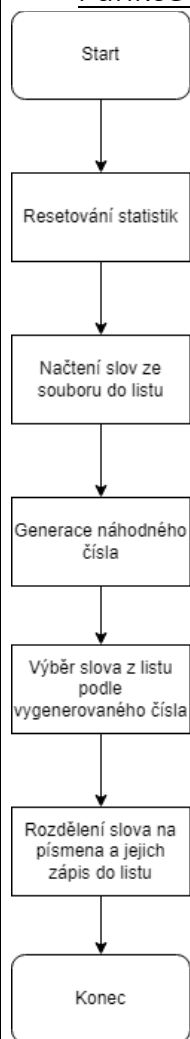
Typ	Argumenty	Metoda	Účel
Class Form1			
void	sender, e	Form1_KeyPress	Zaznamená stisknutou klávesu
void	sender, e	Form1_Load	Prvotní načtení při spuštění programu
Class hangmanclass			
void	-	load	Načte/resetuje hru
bool	guess	guesscheck	Kontroluje, jestli uživatel dané písmeno už nehádal
void	guess	game	Vyhodnotí uživatellovo hádané písmeno
DialogResult	win	resetcheck	Ukáže uživateli výsledek hry, zeptá se jestli chce hrát znovu

Vývojový diagram:

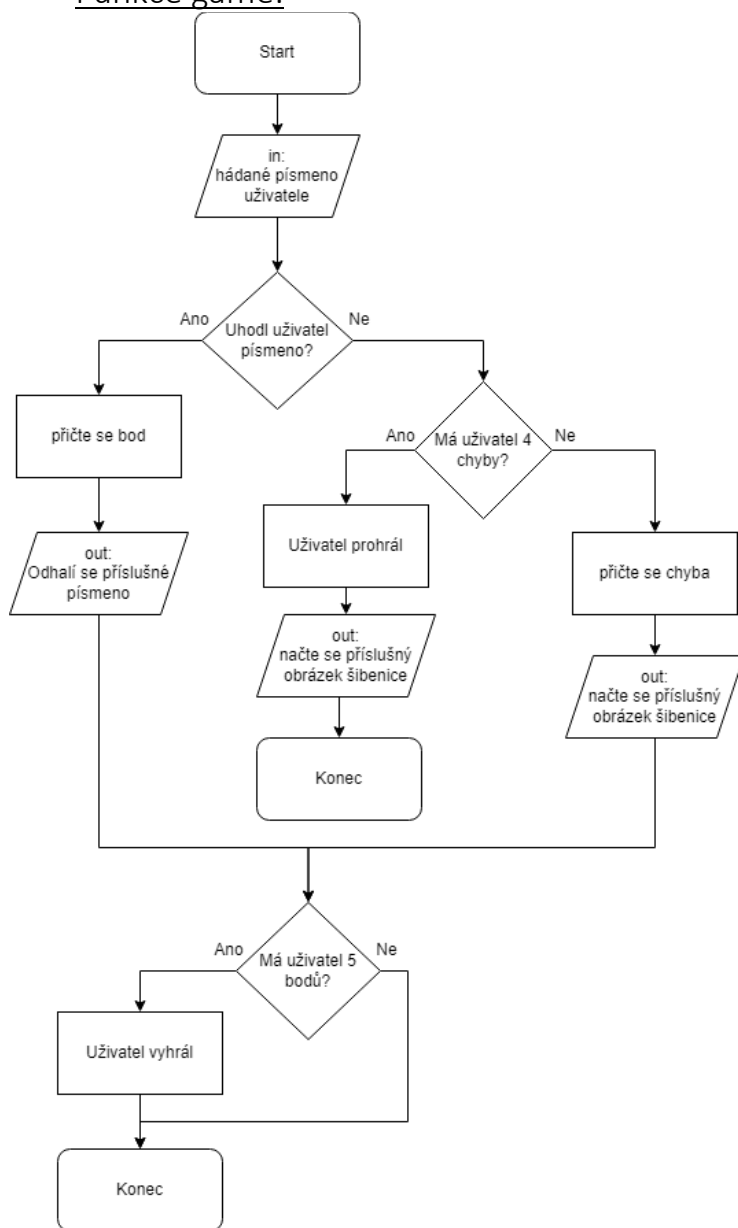
Hlavní program:



Funkce load:



Funkce game:



Výpis Programu:

```
Form1.cs
hangmanclass hang = new hangmanclass();
public Form1()
{
    InitializeComponent();
}
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    char guess = char.ToUpper(e.KeyChar); // převod stisknuté klávesy na character
    if (hang.guesscheck(guess) == true) // pokud uživatel zadal písmeno které ještě
    nezadával...
    {
        hang.game(guess); // vyhodnotí se jeho guess
    }
    if (hang.endgame == true) // Pokud uživatel dohrál...
    {
        DialogResult result = hang.resetcheck(hang.winresult); // Program se zeptá jestli
    chce uživatel hrát znovu
        if (result == DialogResult.Yes) hang.load(); // Pokud ano, hra se resetuje
        else this.Close(); // Pokud ne, program se ukončí
    }
}

private void Form1_Load(object sender, EventArgs e)
{
    hang.used = used;
    hang.pictureBox1 = pictureBox1;
    for (int i = 0; i < 5; i++)
    {
        hang.label = Controls.Find("letter" + i, true);
        hang.labels.Add(hang.label[0]);
    }
    // dosazení ovládacích prvků aplikace do příslušných proměnných objektu
    hang.load(); // Načtení hry
}
```

hangmanclass.cs

```
namespace hangman
{
    class hangmanclass
    {
        public List<string> words = new List<string>(); // seznam možných slov
        public List<char> letters = new List<char>(); // vybrané slovo rozdělené na písmena
        public List<char> guesses = new List<char>(); // hádaná písmena
        public List<Control> labels = new List<Control>(); // list labelů
        public int error = 0; // počet chyb
        public int points; // počet uhodnutých písmen
        public string word; // hádané slovo
        public string a; // Tvoří string hádaných písmen, který se následně vypíše
        public bool endgame = false; // určuje, jestli uživatel dohrál
        public bool winresult; // určuje, jestli uživatel vyhrál
        public Control used; // label hádaných písmen
        public Control pictureBox1; // obrázek šibenice
        public Control[] label; // nalezený label
        public DialogResult result; // Rozhodnutí uživatele jestli chce pokračovat ve hře

        /// <summary>
        /// Načte/resetuje hru
        /// </summary>
        public void load()
        {
            used.Text = "";
            words.Clear();
            letters.Clear();
            guesses.Clear();
        }
    }
}
```

```

error = 0;
points = 0;
endgame = false;
pictureBox1.BackgroundImage = Image.FromFile("Noose0.png");
for (int i = 0; i < 5; i++)
{
    labels[i].Text = "-";
}
// Vyčištění z předchozí hry
string line;
StreamReader fload = new StreamReader("src.csv");
while ((line = fload.ReadLine()) != null)
{
    words.Add(line.ToUpper()); // načte dostupná slova ze souboru src.csv do listu
}
Random rnd = new Random();
int numbah = rnd.Next(0, words.Count - 1);
word = words[numbah]; // náhodně vybere jedno slovo z listu words
foreach (char c in words[numbah])
{
    letters.Add(c); // rozdělí vybrané slovo na jednotlivá písmena
}
}
/// <summary>
/// Kontroluje, jestli uživatel dané písmeno už nehádal
/// </summary>
/// <param name="guess">
/// Uživatelův guess
/// </param>
/// <returns>
/// Bool, jestli bylo písmeno už hádáno
/// </returns>
public bool guesscheck(char guess)
{
    if (guesses.Count == 0) // je potřeba zvlášť podmínka pokavaď uživatel ještě neudělal
        žádné guesy, jinak by nefungoval for cyklus níže
    {
        used.Text += " " + guess; // guess se zapíše do labelu
        guesses.Add(guess); // guess se přidá do listu guessů
        return true;
    }
    for (int j = 0; j < guesses.Count; j++)
    {
        if (guess == guesses[j]) // pokavaď se uživatel pokusí znovu hádat písmeno které
            již zkoušel...
            return false; // program nebude pokračovat
        if (j == guesses.Count - 1) // pokavaď se nenajde stisknuté písmeno mezi již
            hádanými...
            {
                guesses.Add(guess); // přidá se do listu hádaných písmen
                guesses.Sort(); // abecedně se seřadí
                for (int i = 0; i < guesses.Count; i++)
                {
                    a = string.Join(" ", guesses); // všechna hádaná písmena z listu se zapíší
                        do labelu
                }
                used.Text = a;
                return true;
            }
    }
    return true;
}
/// <summary>
/// Vyhodnotí uživatelské hádané písmeno
/// </summary>
/// <param name="guess">

```



```

    /// uživatelův guess
    /// </param>
    public void game(char guess)
    {
        for (int i = 0; i < letters.Count; i++)
        {
            if (guess == letters[i]) // pokavaď se hádané písmeno shoduje s jedním z písmen
            slova...
            {
                labels[i].Text = Convert.ToString(letters[i]); // písmeno se "odhalí" v
                příslušném labelu
                points++; // přičte se bod
                break;
            }
            if (i == letters.Count - 1) // pokavaď se hádané písmeno neshoduje s jedním z
            písmen slova...
            {
                switch (error) // podle počtu chyb se načte příslušný obrázek šibenice
                {
                    case 0:
                        pictureBox1.BackgroundImage = Image.FromFile("Noose1.png");
                        pictureBox1.Size = pictureBox1.PreferredSize;
                        error++;
                        break;
                    case 1:
                        pictureBox1.BackgroundImage = Image.FromFile("Noose2.png");
                        pictureBox1.Size = pictureBox1.PreferredSize;
                        error++;
                        break;
                    case 2:
                        pictureBox1.BackgroundImage = Image.FromFile("Noose3.png");
                        pictureBox1.Size = pictureBox1.PreferredSize;
                        error++;
                        break;
                    case 3:
                        pictureBox1.BackgroundImage = Image.FromFile("Noose4.png");
                        pictureBox1.Size = pictureBox1.PreferredSize;
                        error++;
                        break;
                    case 4:
                        pictureBox1.BackgroundImage = Image.FromFile("Noose5.png");
                        pictureBox1.Size = pictureBox1.PreferredSize;
                        endgame = true; // Uživatel dohrál
                        winresult = false; // Uživatel prohrál
                        return;
                }
            }
        }
        if (points == 5) // pokud uživatel uhodne všechna písmena...
        {
            endgame = true; // uživatel dohrál
            winresult = true; // uživatel vyhrál
            return;
        }
        return;
    }
    /// <summary>
    /// Ukáže uživateli výsledek hry, zeptá se jestli chce hrát znovu
    /// </summary>
    /// <param name="win">
    /// Výsledek hry
    /// </param>
    /// <returns>
    /// Uživatelův výběr v MessageBoxu
    /// </returns>
    public DialogResult resetcheck(bool win)
    {

```

```

        if (win == true) // Pokud uživatel vyhrál...
        {
            result = MessageBox.Show("Vyhrál jsi" + Environment.NewLine + "Chceš pokračovat?",
"WIN WIN WIN", MessageBoxButtons.YesNo);
            // MessageBox oznámí výhru, zeptá se jestli chce další hru
            return result;
        }
        else // pokud uživatel prohrál...
        {
            result = MessageBox.Show("Prohrál jsi" + Environment.NewLine + "hádané slovo: " +
word + Environment.NewLine + "Chceš pokračovat?", "L L L", MessageBoxButtons.YesNo);
            // MessageBox oznámí prohru, zeptá se jestli chce další hru
            return result;
        }
    }
}
}

```

Závěr:

Hra funguje, ale aktuálně podporuje pouze slova na 5 písmen. Mohla by se přidat podpora pro různě dlouhá písmena. Stejně tak program nekontroluje jestli slova načtená ze souboru opravdu jsou na 5 písmen, tudíž pokud se do programu dostane slovo kratší nebo delší, hra nebude fungovat jak má.