STŘEDOŠKOLSKÁ ODBRONÁ ČINNOST

Obor SOČ: 1. Matematika a statistika

Náhodná a pseudonáhodná čísla

Autor: David Carboch
Škola: Střední škola spojů a informatiky Tábor, Bydlinského 2474, 390 11
Kraj: Jihočeský kraj
Konzultant: Mgr. Jiřina Bartoňová

Tábor 2018

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze prameny uvedené v seznamu na konci práce. Prohlašuji, že tištěná verze a elektronická verze soutěžní práce jsou shodné. Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

v Táboře 10.4.2018 ……

Poděkování

Na tomto místě bych rád poděkoval Mgr. Jiřině Bartoňové za obětavou pomoc a podnětné připomínky, které mi během práce poskytovala.

Anotace

Cílem práce je vysvětlit způsob generování náhodných čísel a vysvětlit rozdíl mezi čísly náhodnými a pseudonáhodnými.

Výstupem jsou dvě aplikace v jazyce C#, které využívají generátor náhodných čísel. Konkrétně jsou to aplikace na kontrolu četnosti losovaných čísel a aplikace simulující falešnou hrací kostku.

Klíčová slova

Náhodná čísla; pseudonáhodná čísla; náhoda; generátor náhodných čísel;

Obsah

Úvod 6

1. Náhoda 7

2. Čísla náhodná a pseudonáhodná 8

2.1. Čísla náhodná 8

2.2. Čísla pseudonáhodná 8

2.3. Generátory náhodných čísel 9

2.3.1. Historie generátorů 9

2.3.2. Moderní generátory 9

3. **Aplikace využívající generátory pseudonáhodných čísel** 10

3.1. Náhodná čísla v programovacím jazyce C# 10

3.2. Kontrola četnosti 10

3.3. Falešná hrací kostka 11

Závěr 13

Přílohy 15

Úvod

V hodinách programování jsem se setkal s použitím generátoru náhodných čísel a s tvrzením, že čísla vytvořená tímto nástrojem ve skutečnosti nejsou náhodná, ale tzv. pseudonáhodná. Protože v matematice nás toto téma teprve čeká, rozhodl jsem se alespoň částečně nastudovat problematiku náhodných čísel sám.

Kromě studia teorie bych rád vytvořil dvě aplikace, které využívají generátor náhodných čísel. Konkrétně aplikaci na kontrolu četnosti losovaných čísel a aplikaci simulující falešnou hrací kostku. K tvorbě aplikací využiji programovací jazyk C# a vývojové prostředí Microsoft Visual Studio 2015.

1. Náhoda

Slovo „náhoda“ a slova od něj odvozená používáme velmi často. Náhodné je to, co nastává zdánlivě bez příčiny nebo spíše nejsme schopni příčinu pojmenovat nebo přesně určit.

Při vyhledávání na internetu jsem narazil na velké množství filozofických úvah, které jsem příliš nechápal. Vzhledem k tomu, že mě náhoda zajímá především z hlediska matematického, hledal jsem nějakou přesnou definici.

Mým představám nejvíce odpovídala citace na wikipedii pod heslem náhoda:

„Náhodu lze charakterizovat jako souhrn drobných, ne zcela zjistitelných, nebo vůbec nezjistitelných vlivů.“
P. Hebák – J. Kahounová: *Počet pravděpodobnosti v příkladech*, 3. vydání, SNTL 1988, s. 9

Dále jsem v učebnicích matematiky a různých učebních materiálech na internetu objevil pojem *náhodný pokus*.

Zjistil jsem, že *náhodný pokus* je takový pokus, který při stejných podmínkách vede k různým výsledkům. Opakem náhodného pokusu je *pokus deterministický*. To jsou pokusy, které známe z výuky například fyziky nebo chemie. Tam se předpokládá, že když vše provedeme správně, získáme právě jeden přesně daný výsledek.
U náhodných pokusů je výsledků více a nemůžeme dopředu vědět, který výsledek nastane.

Jako typické příklady náhodných pokusů se uvádějí často různé loterie, tahání karet nebo hody kostkou nebo mincí. Jde tedy buď o náhradu spravedlnosti nebo rozhodnutí nerozhodného výsledku nebo o hazardní hry. Ostatně hazardní hry byly příčinou vzniku a vývoje kombinatoriky a teorie pravděpodobnosti.

Mě ke zkoumání náhody přivedla potřeba vytvořit při programování náhodné číslo podle daných kritérií, tedy přesněji v daném rozsahu. Dále tedy opustím širší hledisko definování náhody a budu se věnovat problematice generování náhodného čísla.

2. Náhodná a pseudonáhodná čísla.

2.1. Čísla náhodná

Nalézt číslo skutečně náhodné je zřejmě velice těžké.

Za skutečně náhodná čísla lze považovat výsledky v praxi provozovaných výše zmíněných náhodných pokusů, tedy např. losování nebo házení kostkou. Dalo by se za ně považovat i to, že nějaký člověk bude čísla vymýšlet, i když ten člověk může být něčím ovlivněn, ale když i volba člověka bude náhodná, třeba losem, tak by se o náhodném čísle dalo uvažovat. To mi ovšem při programování příliš nepomůže.

Pro vytvoření náhodného čísla v počítači je potřeba hardwarový generátor náhodných čísel (*true random number generator - TRNG*). Pro generování náhodného čísla se zde využívá výsledků náhodných fyzikálních procesů. Tato zařízení jsou často založena na mikroskopických jevech, které generují nízkoúrovňové, statisticky náhodné šumové signály, například z tepelného šumu či fotoelektrického jevu nebo jiných kvantových jevů.

V běžném počítači takovou součástku nenajdeme. Tyto generátory se používají tam, kde je třeba zabránit zneužití znalosti algoritmů pro tvorbu čísel pseudonáhodných. Například ve vysoce zabezpečených šifrovacích aplikacích například pro vojenské nebo obchodní účely. Použitím hardwarového generátoru náhodných čísel zamezíme nebezpečí dešifrovacího útoku.

Ani hardwarové generátory náhodných čísel nejsou zcela bezchybné. Jsou často relativně pomalé a mohou produkovat zkreslené sekvence, kdy některé hodnoty mohou být častější než ostatní. K ovlivnění náhodnosti generátoru může dojít například díky elektromagnetickému rušení z jiného zdroje.

2.2. Čísla pseudonáhodná

Pro běžné programování aplikací stačí používání čísel pseudonáhodných. To jsou čísla, která se generují softwarově, pomocí generátorů pseudonáhodných čísel. Takový generátor je program, který generuje posloupnost čísel statisticky velmi podobnou posloupnosti čísel náhodných.

Generátory pseudonáhodných čísel nejčastěji fungují tak, že v okamžiku aktivování generátoru náhodných čísel vznikne tzv. *random seed*, tedy vlastně výchozí stav nebo výchozí hodnota, ze které se další hodnoty vypočítají jako funkční hodnoty nějaké složitější matematické funkce. Výchozí hodnotou mohou být např. různé kombinace systémového času a data apod.

Nevýhodou algoritmů pro generování pseudonáhodných čísel je jejich periodičnost, tedy že se po určité, většinou dlouhé, době opakují. Této skutečnosti lze i využít. Například při generování testovacích dat při ladění programů.

Z hlediska vysoké bezpečnosti jsou tedy posloupnosti generovaných dat teoreticky odhalitelné, ale pro běžné použití jsou postačující a bezpečné.

2.3. Generátory pseudonáhodných čísel

2.3.1. Historie generátorů

Jedním z nejjednodušších a nejstarších generátorů je *Lineární kongruentní generátor*. Byl definován v roce 1949.
Je dán vztahem:

$$x_{i+1} = (ax_{i+}c)\ mod\ m$$

x_0 je počáteční stav; a,c a m jsou vhodně zvolené konstanty;
operace *mod* (modulo) je zbytek po dělení;
čísla c a m by měla být nesoudělná;
číslo by mělo být m dostatečně velké, protože perioda je maximálně m.

Z tohoto vztahu vychází nebo ho modifikuje celá řada dalších generátorů, například *zpožděný Fibonacciho generátor*, *Lehmerův generátor* nebo *inverzní kongruentní generátor*.

2.3.2. Moderní generátory

V současnosti se používají dokonalejší generátory. Jejich algoritmy jsou poměrně složité. Jejich periody (doba, po které se hodnoty začnou opakovat) mohou být 2^{1000} i víc.

3. Aplikace využívající generátory pseudonáhodných čísel

3.1. Náhodná čísla v programovacím jazyce C#

Pro vývoj aplikací jsem použil programovací jazyk C# a vývojové prostředí Microsoft Visual Studio 2015.

Pro generování náhodných čísel se zde používají objekty třídy `Random`. Vytvoření tohoto objektu většinou probíhá při spuštění aplikace a generátor dále běží po celou dobu běhu této aplikace, přesněji po celou dobu existence okna, ve kterém aplikace běží. Pokud by se generátor vytvářel až v okamžiku, kdy ho potřebujeme v metodě, kde ho potřebujeme, došlo by k opakovanému vytváření výchozího stavu a popření alespoň přibližné náhodnosti vytvořených čísel.

K vytvoření nového objektu nazvaného *náhoda* dojde příkazem:

```
Random náhoda = new Random();
```

Přiřazení náhodné celočíselné hodnoty v rozsahu od čísla *dolníMezi* do čísla *horníMez* (včetně mezí) číslu *x* dojde příkazem:

```
int x = náhoda.Next(dolníMez, horníMez+1);
```

Nevím, proč se dolní mez bere včetně a horní ne, tedy se musí psát číslo o 1 větší.

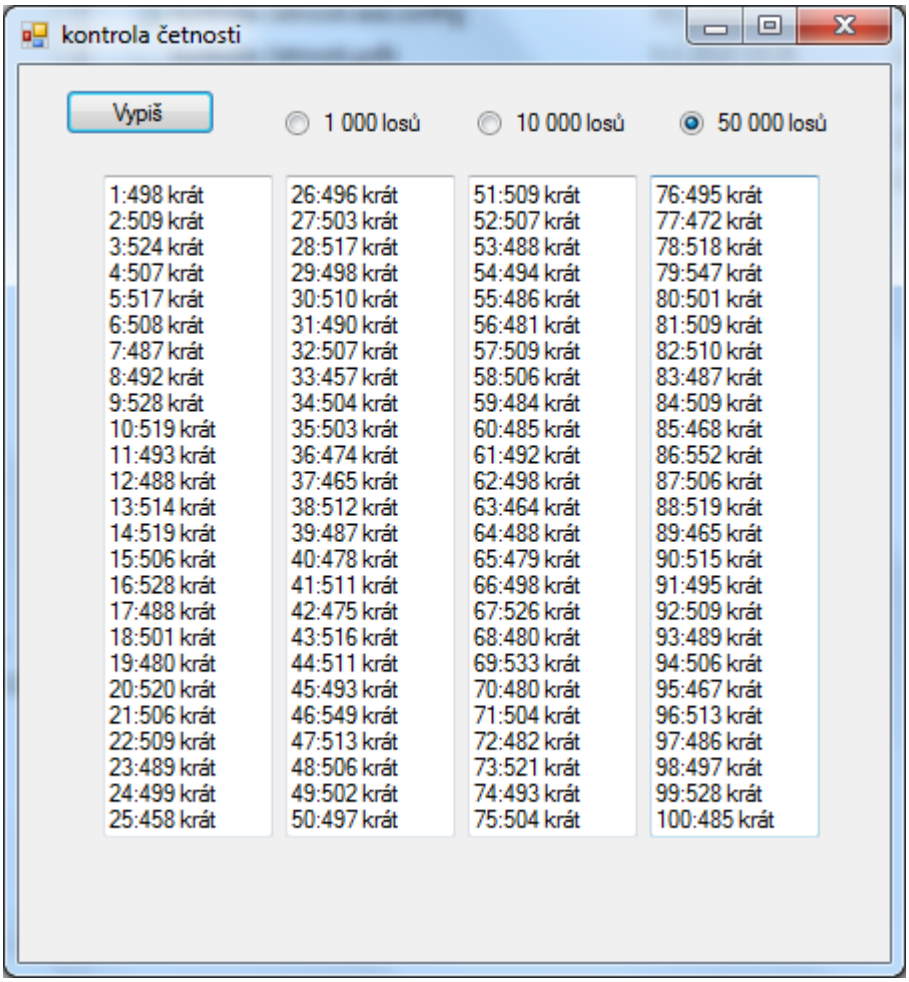
I když z předchozího textu jasně vyplývá, že vytvořená čísla jsou pseudonáhodná, budu dále v souvislosti s programováním používat pojmy *náhodné číslo* a *generátor náhodných čísel*, tak jak je to obvyklé v odborné literatuře, např. v učebnicích programování.

V následujících kapitolách stručně popíšu v\tvořené aplikace. Jejich kompletní zdrojový kód uvedu v příloze 1. Složky s kompletní aplikací a spustitelné aplikace budou součástí elektronické přílohy 2.

3.2. Kontrola četnosti

V této aplikaci si uživatel z nabízených možností vybere počet provedených pokusů losování čísel od 1 do 100. K dispozici má 1000, 10000 a 50000 pokusů. Po stisku tlačítka se vyvolá metoda, která provede daný počet generování čísla od 1 do 100. V proměnné typu pole se uchovává pro každé ze sta čísel počet jejich vylosování. Obsah pole se poté vypíše do textových polí pod tlačítkem. Při použití *n* pokusů, by četnost každého z čísel od 1 do 100 měla být přibližně $\frac{n}{100}$.

Pro vytváření statistik počtu četností losování daných čísel, by bylo třeba testování mnohokrát opakovat a ukládat do souborů tak, aby bylo možné je dále zpracovávat. Tak daleko ovšem mé znalosti programování zatím nesahají.



obrázek 1: formulář aplikace „kontrola četnosti“ - výpis četností losování čísel od 1 do 100, pro 50000 provedených losování

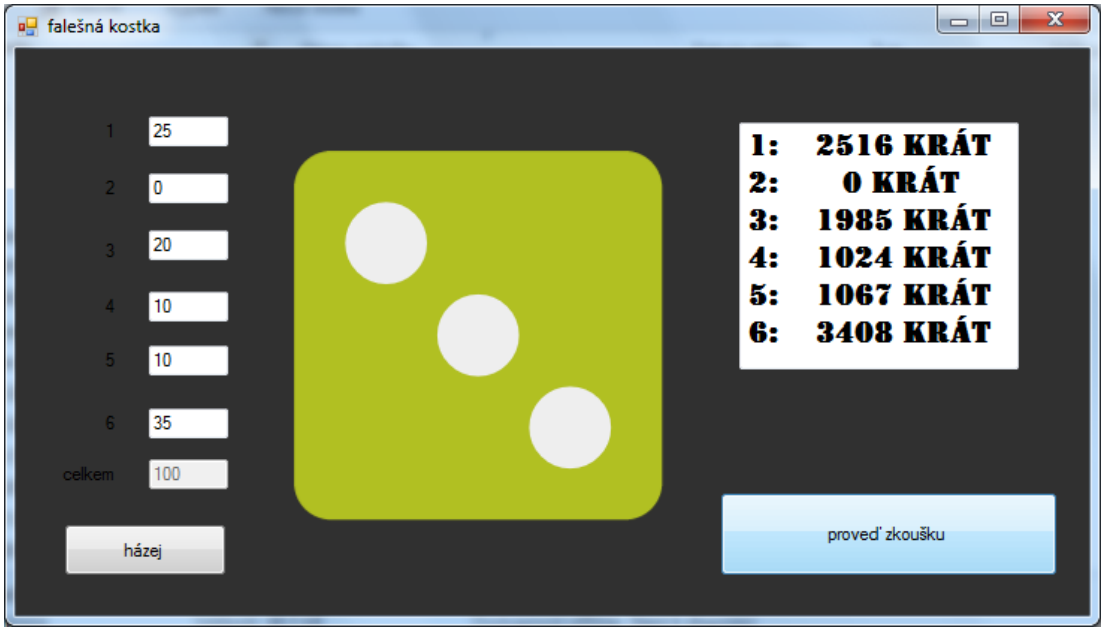
3.3. Falešná hrací kostka

Pro zpestření jsem se rozhodl naprogramovat hrací kostku, která nerespektuje stejnou pravděpodobnost padnutí každého z šesti čísel.

Uživatel si nejprve zvolí pro každou z hodnot na kostce počet procent, ve kterých má padat. Pro kontrolu se po každé změně zadání vytváří součet a po dosažení 100% se aktivuje tlačítko pro házení. Po hození kostkou se objeví příslušný obrázek.

Pro ověření správné „falešnosti“ kostky jsem přidal možnost testování kostky v cyklu 10000 hodů. Vycházel jsem z předchozí aplikace. Jestliže p_n je počet procent pro hodnotu n na kostce, pak počet padnutí by měl být přibližně $p_n * 100$.

Samotný hod na kostce funguje tak, že ve skutečnosti nelosuji číslo od 1 do 6, ale od 1 do 100. Pro vylosované číslo v rozmezí $1 \ldots p_1$ kostce přiřadí hozené číslo 1. Číslo 2 padne pro vylosovaná čísla větší než p_1 a zároveň menší nebo rovna $p_1 + p_2$, atd. až 6 padne pro čísla větší $p_1 + p_2 + p_3 + p_4 + p_5$.



obrázek 2: formulář aplikace „falešná kostka“

Závěr

Při vyhledávání informací o náhodných a pseudonáhodných číslech jsem se dozvěděl řadu zajímavostí. Chtěl jsem se ještě více zabývat algoritmy pro generování pseudonáhodných čísel, ale v záplavě vzorců jsem se trochu ztrácel.

Programování aplikací, pro mě bylo zajímavější, než sběr výše uvedených informací. S celkem nepříliš rozsáhlými zkušenostmi s programováním jsem napsal dvě aplikace na využití generátoru náhodných (spíše pseudonáhodných) čísel.

Na základě zkušeností s tvorbou obou aplikací bych rád v programování na toto téma pokračoval. Mohl bych aplikace vylepšit například o ukládání a další statistické zpracování získaných dat. Také bych se rád pokusil vytvořit aplikaci pro losování tomboly, tak aby byla v praxi opravdu použitelná.

V řešení problematiky generování náhodných a pseudonáhodných čísel vidím potenciál pro maturitní práci, která mě v příštím školním roce čeká.

Zdroje:

https://cs.wikipedia.org/wiki/N%C3%A1hoda

https://cs.wikipedia.org/wiki/Pseudon%C3%A1hodn%C3%A1_%C4%8D%C3%ADsla

https://cs.wikipedia.org/wiki/Hardwarov%C3%BD_gener%C3%A1tor_n%C3%A1hodn%C3%BDch_%C4%8D%C3%ADsel

https://cs.wikipedia.org/wiki/Gener%C3%A1tor_pseudon%C3%A1hodn%C3%BDch_%C4%8D%C3%ADsel

http://www.simulace.info/index.php/Pseudorandom_number_generators/cs

Příloha 1

Zdrojový kód aplikace „kontrola četnosti“

```
namespace kontrola_četnosti
{
    public partial class Form1 : Form
    {
        Random náhoda = new Random();
        public Form1()
        {
            InitializeComponent();
        }
    private void tlacitko_Click(object sender, EventArgs e)
        {
            int[] kolikrát = new int[101];
            int početlosování = 1000;
            if (radioButton10.Checked == true)
                početlosování += 9000;
            if (radioButton50.Checked == true)
                početlosování += 49000;
             for (int číslolosu = 1; číslolosu <= početlosování;
číslolosu++)
            {
                int náhodnéčíslo = náhoda.Next(1, 101);
                kolikrát[náhodnéčíslo]++;
            }
            // sloupec 1
            string zpráva1 = "";
            for (int číslo = 1; číslo <= 25; číslo++)
            {
                zpráva1 += číslo.ToString() + ":" +
kolikrát[číslo].ToString().PadLeft(3) +
                    " krát" + Environment.NewLine;
            }
            textBox1.Text = zpráva1;
            //sloupec 2
            string zpráva2 = "";
            for (int číslo = 26; číslo <= 50; číslo++)
            {
                zpráva2 += číslo.ToString() + ":" +
kolikrát[číslo].ToString().PadLeft(3) +
                    " krát" + Environment.NewLine;
            }
            textBox2.Text = zpráva2;
            // sloupec 3
            string zpráva3 = "";
            for (int číslo = 51; číslo <= 75; číslo++)
            {
                zpráva3 += číslo.ToString() + ":" +
kolikrát[číslo].ToString().PadLeft(3) +
                    " krát" + Environment.NewLine;
            }
            textBox3.Text = zpráva3;
            // sloupec 4
            string zpráva4 = "";
            for (int číslo = 76; číslo <= 100; číslo++)
            {
                zpráva4 += číslo.ToString() + ":" +
kolikrát[číslo].ToString().PadLeft(3) +
                    " krát" + Environment.NewLine;
            }
            textBox4.Text = zpráva4;
        }
    }
}
```

Zdrojový kód aplikace „falešná kostka“

```
namespace falešná_kostka
{
    public partial class Form1 : Form
    {
        int číslo1 = 0;
        int číslo2 = 0;
        int číslo3 = 0;
        int číslo4 = 0;
        int číslo5 = 0;
        int číslo6 = 0;
        int celkem = 0;
        Image obrázek = Properties.Resources.nula;
        Random náhoda = new Random();
        public Form1()
        {
            InitializeComponent();
        }

        private void polečíslo1_TextChanged(object sender, EventArgs e)
        {
            try
            { číslo1 = Convert.ToInt32(polečíslo1.Text); }
            catch { return; }
            if (číslo1 < 0)
            {
                číslo1 = 0;
                polečíslo1.Text = číslo1.ToString();
            }
            celkem = číslo1+číslo2+číslo3+číslo4+číslo5+číslo6;
            polecelkem.Text = celkem.ToString();

        }

        private void polečíslo2_TextChanged(object sender, EventArgs e)
        {
            try
            { číslo2 = Convert.ToInt32(polečíslo2.Text); }
            catch { return; }
            if (číslo2 < 0)
            {
                číslo2 = 0;
                polečíslo2.Text = číslo2.ToString();
            }
            celkem = číslo1 + číslo2 + číslo3 + číslo4 + číslo5 + číslo6;
            polecelkem.Text = celkem.ToString();
        }

        private void polečíslo3_TextChanged(object sender, EventArgs e)
        {
            try
            { číslo3 = Convert.ToInt32(polečíslo3.Text); }
            catch { return; }
            if (číslo3 < 0)
            {
                číslo3 = 0;
                polečíslo3.Text = číslo3.ToString();
            }
            celkem = číslo1 + číslo2 + číslo3 + číslo4 + číslo5 + číslo6;
            polecelkem.Text = celkem.ToString();
        }

        private void polečíslo4_TextChanged(object sender, EventArgs e)
        {
            try
            { číslo4 = Convert.ToInt32(polečíslo4.Text); }
            catch { return; }
```

```
            if (číslo4 < 0)
            {
                číslo4 = 0;
                polečíslo4.Text = číslo4.ToString();
            }
            celkem = číslo1 + číslo2 + číslo3 + číslo4 + číslo5 + číslo6;
            polecelkem.Text = celkem.ToString();
        }

        private void polečíslo5_TextChanged(object sender, EventArgs e)
        {
            try
            { číslo5 = Convert.ToInt32(polečíslo5.Text); }
            catch { return; }
            if (číslo5 < 0)
            {
                číslo5 = 0;
                polečíslo5.Text = číslo5.ToString();
            }
            celkem = číslo1 + číslo2 + číslo3 + číslo4 + číslo5 + číslo6;
            polecelkem.Text = celkem.ToString();
        }

        private void polečíslo6_TextChanged(object sender, EventArgs e)
        {
            try
            { číslo6 = Convert.ToInt32(polečíslo6.Text); }
            catch { return; }
            if (číslo6 < 0)
            {
                číslo6 = 0;
                polečíslo6.Text = číslo6.ToString();
            }
            celkem = číslo1 + číslo2 + číslo3 + číslo4 + číslo5 + číslo6;
            polecelkem.Text = celkem.ToString();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            polečíslo1.Text = číslo1.ToString();
            polečíslo2.Text = číslo2.ToString();
            polečíslo3.Text = číslo3.ToString();
            polečíslo4.Text = číslo4.ToString();
            polečíslo5.Text = číslo5.ToString();
            polečíslo6.Text = číslo6.ToString();
            polecelkem.Text = celkem.ToString();
        }

        private void polecelkem_TextChanged(object sender, EventArgs e)
        {
            if (celkem == 100) { buttonHázej.Enabled = true; button1.Enabled
= true; }
            else { buttonHázej.Enabled = false; button1.Enabled = false; }
        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Graphics kp = e.Graphics;
            kp.DrawImage(obrázek,180,60,260,260);
        }

        private void buttonHázej_Click(object sender, EventArgs e)
        {
            int hozenéČíslo = náhoda.Next(1, 101);
            int čísloKostka=0;
            if (hozenéČíslo <= číslo1) čísloKostka = 1;
            else if (hozenéČíslo <= (číslo1 + číslo2)) čísloKostka = 2;
```

```
            else if (hozenéČíslo <= (číslo1 + číslo2 + číslo3)) čísloKostka
= 3;
            else if (hozenéČíslo <= (číslo1 + číslo2 + číslo3 + číslo4))
čísloKostka = 4;
            else if (hozenéČíslo <= (číslo1 + číslo2 + číslo3 + číslo4 +
číslo5)) čísloKostka = 5;
            else čísloKostka = 6;
            switch (čísloKostka)
            {
                case 1:obrázek = Properties.Resources.jedna;
                    break;
                case 2:obrázek = Properties.Resources.dva;
                    break;
                case 3:
                    obrázek = Properties.Resources.tri;
                    break;
                case 4:
                    obrázek = Properties.Resources.ctyri;
                    break;
                case 5:
                    obrázek = Properties.Resources.pet;
                    break;
                case 6:
                    obrázek = Properties.Resources.sest;
                    break;
            }
            Refresh();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int[] kolikrát = new int[7];
            int početlosování = 10000;
            for (int číslolosu = 1; číslolosu <= početlosování; číslolosu++)
            {   //1
                int náhodnéčíslo = náhoda.Next(1, 101);
                if(náhodnéčíslo<=číslo1)
                kolikrát[1]++;
                //2
               else if (náhodnéčíslo <= číslo1+číslo2)
                    kolikrát[2]++;
                //3
               else if (náhodnéčíslo <= číslo1+číslo2+číslo3)
                    kolikrát[3]++;
                //4
                else if (náhodnéčíslo <= číslo1+číslo2+číslo3+číslo4)
                    kolikrát[4]++;
                //5
                else if (náhodnéčíslo <= číslo1+číslo2+číslo3+číslo4+číslo5)
                    kolikrát[5]++;
                //6
                else

                    kolikrát[6]++;
            }
            //výpis
            string zpráva1 = "";
            for (int číslo = 1; číslo <= 6; číslo++)
            {
                zpráva1 += číslo.ToString() + ":" +
kolikrát[číslo].ToString().PadLeft(8) +
                   " krát" + Environment.NewLine;
            }
            textBox1.Text = zpráva1;
        }
    }
}
```

Příloha 2 – obsah vloženého CD

- Text práce ve formátu .PDF
- Složka s kompletním zdrojovým kódem aplikace *kontrola četnosti*
- Samostatně spustitelná aplikace *falešná kostka*
- Samostatně spustitelná aplikace kontrola četnosti