Na čem mi záleží (co bych ráda, abyste si odnesli):

* abyste si uvědomili, že některé údaje se v paměti počítače ukládají přesně a některé se zaokrouhlovací chybou, někdy se mohou zaokrouhlovací chyby akumulovat a způsobit fatální věci
* abyste věděli, že velikost paměti pro ukládání dat má své limity, do standardních / tradičních datových proměnných nelze uložit libovolně velké / malé číslo
* abyste si odnesli tu znalost, že je dobré přemýšlet nad počtem provedených operací a hledat možnosti zrychlení kódu
* abyste nezapomínali na to, že mohou nastat neočekávané chyby, které je potřeba mít ošetřené
* abyste sledovali vlastnosti algoritmu, věděli, že záleží na velikosti datového souboru, který je algoritmem zpracováván, že závislost časové (i paměťové) náročnosti algoritmu na velikosti dat hraje významnou roli

Část 1

**OTÁZKY**

Programovací jazyky se podle míry abstrakce dělí na dvě základní skupiny. Na jaké?

Uveďte příklad jednoho vyššího programovacího jazyka.

Který z uvedených programovacích jazyků je přísně objektový?

1. Assembler
2. C++
3. Java
4. strojový kód

Co vypíše řádek číslo 12 následujícího kódu?

10 string precteno;

11 precteno = "Ahoj!";

12 Console.WriteLine("Zadali jste text: " + precteno);

**ÚKOLY**

- napište program, který na obrazovku napíše:

vaše jméno

ročník

číslo kroužku

- napište program, který bude demonstrovat rozdíl mezi metodou Write a WriteLine

- napište program a zjistěte, co dělá dvojice znaků \n \a \t při vypisování textu na obrazovku

Část 2

**OTÁZKY**

Jaké dva základní datové typy existují v jazyce C#? Jaký je mezi nimi rozdíl?

Který číselný datový typ se ukládá přesně?

Který datový typ je při operacích zatížen zaokrouhlovací chybou?

**ÚKOLY**

Objem koule, kde je chyba?

static void Main(string[] args)

{

// čísla desetinná

// datový typ DOUBLE

// třída Math

// Math.PI Math.Sin() Math.Pow

// deklarace proměnných

double polomer;

double objem;

polomer = 1;

objem = 4 / 3 \* Math.PI \* polomer \* polomer \* polomer;

Console.WriteLine("Objem: " + objem.ToString("F02"));

// vypíše na dvě desetinná místa

Console.ReadKey();

}

Podívejte se na následující kód a vysvětlete, proč nejsou výsledky správně.

// celá čísla - používají se často jako čítač

// pro datový typ INT je maximum 2 147 483 647

// deklarace proměnné

int prom1;

prom1 = 2147483647; // 4 byty

Console.WriteLine("Maximální INTEGER: " + prom1);

// přiřazovací příkaz

prom1 = prom1 + 1;

Console.WriteLine("Přičtena 1 k maximu: " + prom1);

// tiché přetečení

Console.ReadKey();

Podívejte se na následující článek: <https://zahranicni.ihned.cz/c1-63208370-korea-gangnam-style-youtube-pocet-zhlednuti-dve-miliardy-rozbite-pocitadlo>

Úkol: najděte nebo vymyslete nějaký příklad, kdy může snadno dojít k zaokrouhlovací chybě

Podívejte se na video prof. Slavíčka (např. start rakety Arian 5 (15. minuta)): <https://slideslive.com/38902806/slepe-ulicky-vedy-perly-ze-smetiste-vedeckych-myslenek>

Část 3

**OTÁZKY**

Přiřazovací příkaz: jaká hodnota bude uložena v proměnné i za řádkem 03?

02 int i = 10;

03 i = 2\*i + 1;

Co znamená přetížení operátoru? Uveďte příklad.

Co bude uloženo v proměnné k po provedení následující sekvence příkazů?

02 int i = 9, j = 5, k;

03 k = i / j;

Co bude uloženo v proměnné k po provedení následující sekvence příkazů?

02 int i = 9, j = 5, k;

03 k = i % j;

Jaký datový typ použijete pro uložení počtu studentů ve škole, předpokládejte, že studentů je maximálně několik tisíc?

Jaký datový typ je vhodný pro práci s finančními údaji?

**ÚKOLY**

Naučte se přemýšlet o tom, zda daný algoritmus je možné nějakým způsobem zrychlit

* Napište výpočet polynomu tak, abyste použili co nejméně aritmetických operací

// polynom minimum aritmetických operací

double poly, x = 3.0;

poly = x \* x + 5 \* x + 6;

// šlo by to rychleji?

// šetříme čas

poly = x \* x \*x + 5 \* x \*x + 6 \*x + 7;

* prohlédněte si kód pro vytvoření jednotkové matice a popřemýšlejte, zda je možné ho zrychlit

static void Main(string[] args)

{

// jednotková matice

// deklarace

int[,] jednotkova;

int N = 10; // rozmer matice

jednotkova = new int[N, N];

for (int i=0; i<N;i++)

for (int j = 0; j < N; j++) // vnoreny cyklus

{

if (i == j)

jednotkova[i, j] = 1; // diagonála

else

jednotkova[i, j] = 0; // mimo diagonálu

// 200 operaci

}

//tisk

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

Console.Write(jednotkova[i, j] + " ");

}

Console.WriteLine(); // prejde na dalsi radek

}

// jak to napsat rychleji

// aby se pouzilo mene prikazu

Console.ReadKey();

}

Část 4

**OTÁZKY**

B. Co chybí v následujícím kódu?

1. podmínka na začátku
2. středník na konci posledního řádku
3. čítač pro načítání počtu iterací

do

{

Console.WriteLine("doPocitadlo is at " + doPocitadlo);

doPocitadlo++;

} while (doPocitadlo < 5)

C. Vyberte tři rozdíly mezi příkazem while a do…while

1. příkaz do má testování podmínky na konci smyčky
2. příkaz do testuje podmínku na začátku smyčky
3. příkaz while testuje pravdivost podmínky na konci
4. příkaz while testuje pravdivost podmínky na začátku
5. příkazy ve smyčce do se vždy provedou alespoň jednou bez ohledu na podmínku
6. příkazy ve smyčce while se vždy provedou alespoň jednou bez ohledu na podmínku

D. Jaké hodnoty vypíše následující smyčka:

for (int i = 0 ; i < 4; i++)

{

Console.WriteLine(“hodnota čítače i: “ + i);

}

E.Jaké hodnoty vypíše následující smyčka

for (int i = 2 ; i <= 20; i=i+2)

{

Console.WriteLine(“hodnota čítače i: “ + i);

}

F. Jaká hrubá chyba je v následujícím kódu

int i=0;

while (i < 4)

{

Console.WriteLine(“hodnota čítače i: “ + i);

}

**ÚKOLY**

Podívejte se na následující program. Ten počítá druhou odmocninu pomocí metody půlení intervalu. Zamyslete se, jaké chyby mohou nastat a jaká ošetření chyb byste měli udělat, aby program nespadl. Předpokládejte, že veličinu x si vyžádáte od uživatele.

static void Main(string[] args)

{

// metoda půlení intervalu

// druhá odmocnina ze zadaného čísla

double horni, dolni, stred;

double x = 0.49, presnost = 0.00001;

stred = (horni + dolni) / 2;

Console.WriteLine(stred);

// dokud není splněna přesnost, výpočet se opakuje

while (Math.Abs(stred\*stred-x) > presnost)

{

// v které polovině má pokračovat výpočet

if (stred \* stred > x)

horni = stred; // levá polovina intervalu

else

dolni = stred;

// nový střed intervalu

stred = (horni + dolni) / 2;

Console.WriteLine(stred);

}

Console.ReadKey();

}

Přečtěte si článek: <https://www.idnes.cz/technet/vesmir/margaret-hamilton-programatorka-nasa-apollo-error-free-software.A160721_114104_tec_vesmir_pka>

Část 5

**OTÁZKY**

Který blok se vykoná vždy? Nezáleží na tom, zda nastala výjimka nebo ne.

1. blok try
2. specifický blok catch
3. blok finally

**ÚKOLY**

Prohlédněte si kód pro třídění podle velikosti, v kódu je zařazeno měření času. Program několikrát pusťte pro různé velikosti vektoru pole, zapište si spotřebovaný čas do tabulky. Údaje vyneste do grafu: závislost času na velikosti datového souboru.

static void Main(string[] args)

{

// algoritmus Select Sort

int[] pole = new int[10000];

// objektová proměnná pro náhodná čísla

Random nahodne = new Random();

// naplnit pole náhodnými celými čísli

for (int i = 0; i < pole.Length; i++)

pole[i]= nahodne.Next();

// prochází prvky pole a hledá minimum, zapamutuje si index minima

// a zamění minimum a posadí ho na správné místo

// pomocny tisk

for (int i = 0; i < 7; i++)

Console.Write(pole[i] + " ");

Console.WriteLine();

// mereni casu

DateTime start = DateTime.Now;

for (int i = 0; i < pole.Length; i++)

{

int indexminima = i;

// cyklus, ktery hleda minimum

for (int j = i+1; j < pole.Length; j++)

{

// hledam minimum

if (pole[j] < pole[indexminima])

indexminima = j;

// pokud najde mensi cislo, zapamatuje si novy index minima

}

// zamena

int pomocna = pole[i];

pole[i] = pole[indexminima];

pole[indexminima] = pomocna;

}

// mereni casu

DateTime konec = DateTime.Now;

Console.WriteLine("Mereni casu " + (konec-start));

// pomocny tisk

for (int i = 0; i < 7; i++)

Console.Write(pole[i] + " ");

Console.WriteLine();

Console.ReadKey();

}