

### 1) Vyberte pravdivá tvrzení

- a) Volání metody interrupt vždy ukončí vlákno, ve kterém je volána
- b) Volání metody interrupt vždy ukončí vlákno, na jehož objekt je volána
- c) Pro ukončení vlákna se většinou používá volání metody stop
- d) Ukončené vlákno je možno znovu spustit
- e) Volání metody interrupt nemusí způsobit ukončení vlákna. Vlákno se ukončí až opuštěním metody run.

Co dělá Interrupt?

sleeping    waiting	Vyhodí Exception
NOT sleeping && NOT waiting	Nic se nestane a Thread běží dál

### 2) Vyberte pravdivá tvrzení (ošetřením výjimky je myšleno, že se zavolá specifický kód při vyvolání výjimky. Vyhozená výjimka je v tomto kódu dostupná)

- a) Sekce catch ošetří výjimky stejného typu a nadtypu jako je typ uveden v této sekci. Ostatní výjimky jsou ignorovány a program běží dál.
- b) Sekce catch ošetří výjimky stejného typu jako je typ uveden v této sekci
- c) Sekce catch ošetří výjimky stejného typu a nadtypu jako je typ uveden v této sekci
- d) Sekce catch ošetří výjimky stejného typu jako je typ uveden v této sekci a ostatní výjimky jsou ošetřeny v sekci finally
- e) Sekce catch ošetří výjimky stejného typu a podtypu jako je typ uveden v této sekci

Proč?

Jelikož podtyp je součástí nadřazeného typu, program nám vyhodí stejnou výjimku, pokud se jedná o stejný typ problému, který podtyp řeší.

### 3) Vyberte pravdivé tvrzení (uvedené třídy by měly být v balíku java.IO)

- a) Pro čtení a zápis do souborů slouží třída **File**. Má metody fread, fwrite, fseek a další
- b) Potomci třídy **Writer** jsou určeny pro zápis do streamů na úrovni znaků
- c) Potomci třídy **OutputStream** jsou určeny pro čtení ze streamu na úrovni bytů
- d) Potomci třídy **InputStream** jsou určeny pro zápis do streamů na úrovni bytů
- e) Potomci třídy **InputStream** jsou určeny pro čtení ze streamů na úrovni znaků

Proč?

Třída **FILE** neobsahuje metody fread(), fwrite(), fseek() a atd.

**FileReader** a **FileWriter** jsou třídami, které obsahují metody výše obsazené.

Třída **FILE** slouží pro vytváření a mazání souborů, nikoliv úpravu.

#### 4) Vyberte pravdivé tvrzení

- a) V jednom souboru může být definována nejvýše jedna veřejná, nevnořená třída
- b) V jednom souboru může být definována nejvýše jedna veřejná, vnořená třída
- c) Zdrojové kódy třídy jsou v souboru s koncovkou class. Přeložený java bytecode je v souborech s koncovkou java
- d) V jediném zdrojovém Java souboru může být definována právě jedna třída a vždy se jmenuje stejně jak otento soubor (bez koncovky)
- e) Nejsou kladeny žádná omezení na počet tříd ve zdrojových souborech, pouze nesmí mít stejný název

#### Vnořené třídy

Množství vnořených tříd, která třída může mít není omezená.

Vnořené třídy mohou být **public**, **private** i **protected**.

Třída, ve které jsou musí být však **public**!

#### Nevnořené třídy

Je vždy **PUBLIC**!

#### 5) Vyberte pravdivé tvrzení o metodě hashCode (nutné pro správnou funkci hashovací tabulky)

- a) Pro dva objekty (stejného typu), pro které equals vrací true by měla metoda hashCode vracet stejnou hodnotu a pro ostatní musí vracet různou hodnotu
- b) Pro dva objekty (stejného typu), pro které equals vrací true musí metoda hashCode vracet stejnou hodnotu a pro ostatní by měla (aby hashování probíhalo korektně) vracet různou hodnotu
- c) Metoda hashCode je korektně definována ve třídě Object a proto ji nemusíme překrývat
- d) Metoda hashCode je nekorektně definována ve třídě Object a je ji nutno překrýt
- e) Metodu hashCode volá programátor, pokud chce objekt uložit do hashovací tabulky

#### Metoda hashCode()

Tato metoda je obsažena v třídě **INTEGER**.

hashCode in Java is a function that returns the hashcode value of an object on calling.

**It returns an integer** or a 4 bytes value which is generated by the hashing algorithm.

#### 6) Vyberte pravdivou deklaraci metody main, která slouží jako spouštěcí

- a) public static int main(String []args) { ...
- b) public static void main(String []argv, int args) { ...
- c) public static void main(String []args) { ...
- d) public int main(String []args) { ...
- e) public final void main(String []args) { ...

### 7) Jak je definována metoda equals?

- a) public int equals(Object object1)
- b) public boolean equals(Object object1, Object object2)
- c) public boolean equals(Object obj)
- d) int equals (Object object1, Object object2)
- e) boolean equals(Object object1, Object object2)

V kodu

```
String a, b;  
Systém.out.println(a.equals(b));
```

### 8) Vyberte možnost, která je tvořena pouze klíčovými slovy v Javě

- a) class, extends, final, **sleep**, int, throws
- b) interface, char, boolean, else, **Integer**, for
- c) final, instanceof, super, void, interface, throw
- d) abstract, **virtual**, private, while, float, int
- e) break, class, new, **unsigned**, switch, boolean

Neklíčová slova

DELETE, PRINT, STRING, CONST, GOTO, SLEEP, INTEGER, VIRTUAL, UNSIGNED

### 9) Vyberte tvrzení, které není pravdivé

- a) Java podporuje přímo vlákna a poskytuje synchronizační mechanismy mezi vlákny
- b) Jazyk Java je kompilovaný jazyk a program se musí zkompilovat
- c) Java aplikace při přenosu na jinou platformu nevyžadují rekompilaci
- d) Množství paměti, kterou zabírají numerické proměnné (např. int), závisí na verzi virtuálního stroje Javy
- e) Pro běh Java aplikace je vyžadován virtuální stroj instalovaný v systému

## 10) Vyberte pravdivé tvrzení

- a) Každá třída obsahuje vždy bezparametrický konstruktor (pokud nedefinujeme jiný), takže ho nemusíme definovat
- b) Globální proměnné jsou proměnné, které nejsou definovány uvnitř žádné třídy nebo bloku a jsou viditelné odkudkoli
- c) Třídní proměnná je platná pouze pokud existuje objekt dané třídy
- d) Klíčové slovo final u deklarace instanční metody způsobí, že ji potomci nezdědí
- e) Aby "garbage collector" zjistil, že daný objekt není využíván, tak u něho programátor musí nastavit příznak dispose

Globální proměnné v Javě

Nejsou.

Důležité

Třídní proměnná může být platná i bez vytvoření instance třídy.

Final

Pouze způsobí, že se daná metoda či proměnná nedá přepsat.

garbage collector

Funguje automaticky a není třeba nastavovat třídu, aby pro ni byl viditelný.

Metoda Dispose

Používá se pro grafické objekty.

## 1) Vyberte pravdivé tvrzení

- a) Sekce finally se provede vždy, když se opouští blok try
- b) Kontrolované výjimky jsou vyhazovány v kódu pomocí throws a dědí přímo ze třídy Exception
- c) Nekontrolované výjimky nelze ošetřit a dědí ze třídy RuntimeException
- d) Všechny výjimky, vyjma Error, lze ošetřit pomocí konstrukce try catch
- e) V sekci finally se ošetřují všechny výjimky

### Finally

The finally block always executes when the try block exits (finishes).

This ensures that the finally block is executed even if an unexpected exception occurs.

### Throws (Nekontrolované výjimky)

Nemůže dědit, jelikož to je klíčovým slovem.

### Error vs Exception

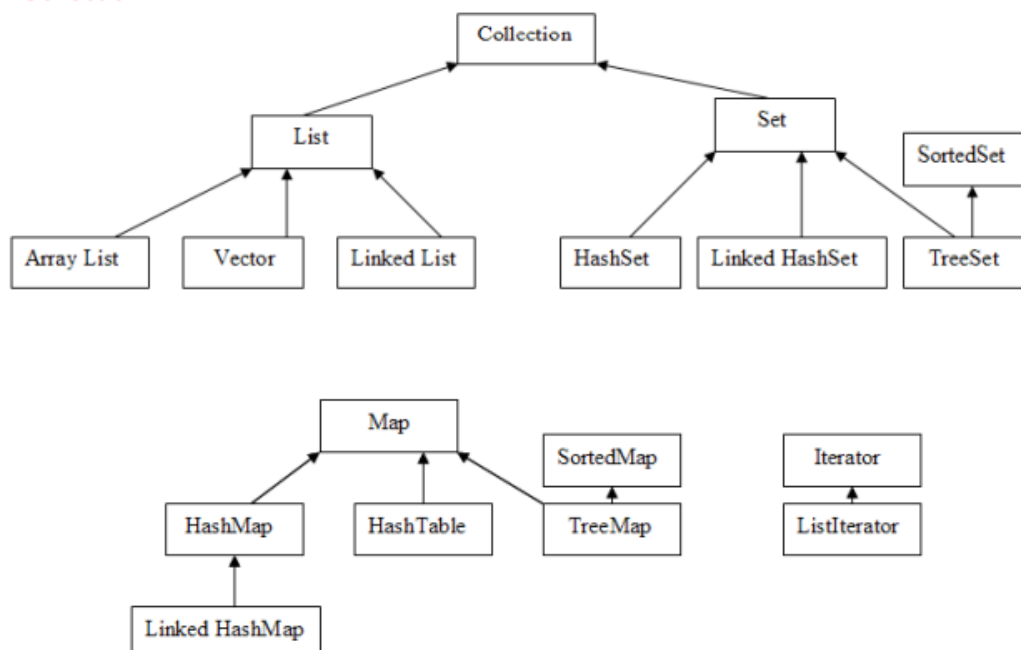
Error: An Error indicates serious problem that a reasonable application should not try to catch.

Exception: Exception indicates conditions that a reasonable application might try to catch.

## 2) Vyberte pravdivé tvrzení

- a) Rozhraní Collection je rozšiřováno rozhraním Set a ArrayList
- b) Rozhraní Collection umožňuje přistupovat k datům nejen pomocí indexů, ale i klíčů
- c) Z rozhraní Collection dědí rozhraní List
- d) Objekty implementující rozhraní Map realizují zobrazení Keys x Values  $\rightarrow N$
- e) Zřetězené seznamy (LinkedList) nedovolují přístup pomocí indexů, protože by to zpomalovalo virtuální stroj

### Dědičnost z Collection



### 3) Vyberte pravdivé tvrzení

- a) Rozhraní definují kontrakt (signatury metod), nikoliv jejich implementaci
- b) Rozhraní byla v JAVĚ definována, protože není možná násobná dědičnost
- c) Třída může implementovat více než jedno rozhraní, tato rozhraní nesmí ovšem obsahovat metody se stejnou signaturou
- d) Metody deklarované v rozhraní jsou buď veřejné (public) nebo chráněné (protected)
- e) Třída, která se přihlásí k implementaci rozhraní nemusí implementovat všechny metody v tomto rozhraní. Tyto metody mají automaticky prázdné tělo a jsou označeny jako @override.

#### Rozhraní (Interface)

Je vždy public, včetně jejich metod (signatura).

Může obsahovat konstanty, které jsou public.

### 4) Vyberte pravdivé tvrzení

- a) Třídy se deklarují pomocí konstrukce virtual class, objekty se vytváří pomocí new()
- b) Třídy se deklarují pomocí klíčového slova class, objekty se vytváří pomocí new()
- c) Třídy se deklarují pomocí klíčového slova class, objekty se vytváří pomocí new() a odstraňují se z paměti pomocí delete
- d) Třídy se deklarují pomocí def class, objekty se vytváří pomocí new() a odstraňují se z paměti pomocí delete
- e) Třídy se deklarují pomocí def class, objekty se vytváří pomocí new()

#### Virtual

Může být pouze metoda.

Nepíše se.

### 5) Vyberte variantu, kde jednotlivá slova jsou používána a funkční klíčová slova

- a) final, finally, import, void, int, throw
- b) class, extends, final, print, int, throws
- c) abstract, const, private, while, float, int
- d) interface, char, boolean, else, String, for
- e) break, class, new, delete, switch, boolean

### 6) Vyberte pravdivá tvrzení

- a) Garbage kolektor uvolní paměť vždy, když se objekt nepoužívá (je disposable).
- b) Operátor new se používá pro dynamickou konstrukci objektu jinak se konstruuje bez new.
- c) Přiřazení hodnoty mezi objektovými proměnnými dochází ke kopírování objektu.
- d) Při přiřazení hodnoty mezi objektovými proměnnými dochází se používá reference. (tak nějak to bylo)
- e) --doplňte, nepamatuju

### 7) Vyberte pravdivá tvrzení

- a) Lokální proměnná je vždy platná po celou dobu existence objektu, v němž je definovaná
- b) Instanční proměnné se dají modifikovat vždy pomocí třídních metod, i přestože jsou private
- c) Volání metody s názvem super v konstruktoru způsobí volání konstruktoru předka, nesmí mu předcházet žádný jiný příkaz
- d) Klíčové slovo return je vždy pouze na posledním řádku zdrojového kódu s definicí metody
- e) Klíčové slovo final použité u třídy zabrání vytváření potomků z této třídy

### 8) Vyberte pravdivé tvrzení

- a) Pro spuštění vlákna překrýváme metodu start v potomkovi třídy Thread a pak ji voláme
- b) Pro spuštění vlákna překrýváme metodu run v potomkovi třídy Thread a pak ji voláme
- c) Pro spuštění vlákna překrýváme metodu start v potomkovi třídy Thread a pak voláme metodu run
- d) Vlákno při vstupování do sekce synchronized výhradně získá monitor zde uvedeného objektu nebo počká až monitor uvolní jiné vlákno
- e) K volání metody notify na určitý objekt je nutné vlastnit monitor k vláknu, které je uspáno pomocí wait

### 9) Vyberte pravdivá tvrzení

- a) Metoda equals nesmí pro dvě různé instance dané třídy vracet true
- b) Metodu equals je možno překrýt ve třídě Object
- c) Metoda equals nikdy nevrací pro dvě instance třídy false, pokud mají stejný stav
- d) Metoda equals se volá vždy, když použijeme operátor == na dva objekty
- e) Metoda equals může vracet pro dva různé objekty stejné třídy true

### 10) Vyberte pravdivé tvrzení

- a) Přetížené metody se liší pouze počtem vstupních parametrů
- b) Přetížené metody jsou jiné implementace zděděných metod v potomkovi, realizují polymorfismus
- c) Přetížené metody se liší počtem nebo typem vstupních argumentů
- d) Přetížené metody se liší počtem nebo typem vstupních argumentů anebo návratovým typem
- e) Přetížené metody jsou označeny pomocí anotace @overload

### 11) Letadlo je třída. Vyberte pravdivé tvrzení

- a) Konstrukce objektu a přiřazení do proměnné je ... let := new Letadlo()
- b) Konstrukce objektu a přiřazení do proměnné je ... let = Letadlo()
- c) Konstrukce objektu a přiřazení do proměnné je ... let = new Letadlo(10)
- d) Konstrukce objektu a přiřazení do proměnné je ... let = new Letadlo()
- e) Konstrukce objektu a přiřazení do proměnné je ... let = new Letadlo(int rychlost)

### 1) Vyberte pravdivé tvrzení:

- a) Že třída dědí z nějaké třídy je definováno pomocí klíčového slova extends. Toto se používá i pro definici, že implementuje nějaké rozhraní.
- b) Že třída dědí z nějakých tříd (není omezeno z kolika) je definováno pomocí klíčového slova extends.
- c) **Že rozhraní dědí z jiného rozhraní je definováno pomocí klíčového slova extends.**
- d) Že třída dědí z nějaké třídy je definováno pomocí klíčového slova extends. Pro dědičnost mezi rozhraními se používá implements.
- e) Že třída dědí z nějaké třídy je definováno pomocí klíčového slova implements. Tato konstrukce se používá i u rozhraní.

Rozhraní které dědí jiná rozhraní

```
public interface GroupedInterface extends Interface1, Interface2, Interface3
```

### 2) Necht' Osoba je třída. Vyberte pravdivé tvrzení:

- a) Správné volání instanční (nestatické) metody této třídy může být skoc(10)-> joe
- b) Správné volání instanční (nestatické) metody této třídy může být joe->skoc(10)
- c) Správné volání instanční (nestatické) metody této třídy může být Osoba.skoc()
- d) **Správné volání instanční (nestatické) metody této třídy může být joe.skoc(10)**
- e) Správné volání instanční (nestatické) metody této třídy může být joe.skoc

### 3) Vyberte pravdivé tvrzení o tom, kdy fakticky skončí (korektně) vlákno (spuštěné pomocí Thread)?

- a) Když se naň zavolá metoda interrupt
- b) Když se ukončí metoda main
- c) Když se naň zavolá metoda stop
- d) **Když se ukončí metoda run vlákna**
- e) Když se naň zavolá metoda join

### 4) Mějme konstantu objektového typu "final Person auditor". Vyberte pravdivé tvrzení.

- a) **Nemůžeme změnit to, na který objekt odkazuje uvedený atribut, ale můžeme změnit stav tohoto objektu.**
- b) Můžeme změnit to, na který objekt odkazuje daný atribut, ale nemůžeme měnit stav tohoto objektu. Objekt je "immutable".
- c) Objekt na nějž odkazuje atribut auditor, je zničen až při skončení programu - je "indestructible". Stav tohoto objektu je neměnný.
- d) Ze třídy Person není možno dědit, stejně jako není možno měnit stav objektu auditor.
- e) Nemůžeme změnit to, na který objekt odkazuje uvedený atribut, a ani stav tohoto objektu. Objekt je "immutable".



## 5) Vyberte pravdivá tvrzení:

- a) Java přímo podporuje vlákna, jazyk je poměrně komplikovaný ale má jednoduché API.
- b) Jazyk Java je interpretovaný jazyk, a proto může běžet na různých platformách. Podporuje více vláknové aplikace.
- c) Java je kompilovaný jazyk a přenositelnost mezi platformami je zajištěna pomocí univerzálního bytecode.
- d) Jazyk Java je kompilovaný jazyk a aplikace se kompilují pro každou platformu zvlášť (Linux, Windows, MAC OS).
- e) Java je objektově orientovaný jazyk, ale nepodporuje přímo více vláknové aplikace.

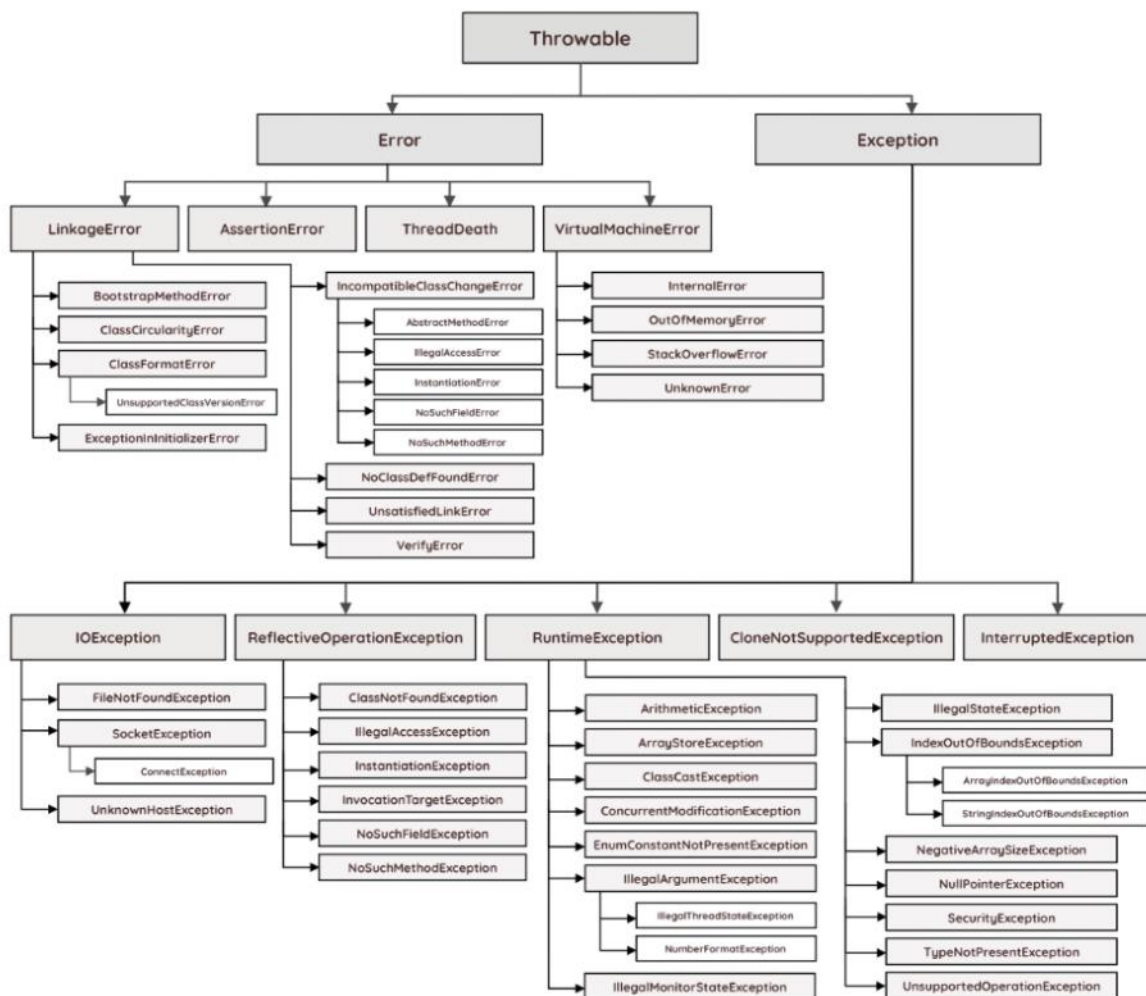
## 6) Vyberte pravdivá tvrzení:

- a) Společnou nadtřídou všech výjimek je Exception
- b) Kontrolované výjimky dědí ze třídy Exception
- c) Kontrolované výjimky se smí šířit z metody, pokud je tato metoda označena klíčovým slovem throw.
- d) Třída RuntimeException dědí ze třídy Error a je nadtřídou pro nekontrolované výjimky.
- e) Klíčové slovo final označuje sekci, kterou při obsluhování výjimky nelze přerušit.

### Výjimky

Kontrolované výjimky = Exception

Nekontrolované výjimky = Error



### 7) Vyberte pravdivé tvrzení:

- a) Třída DataInputStream slouží pro serializace / deserializace dat
- b) Instanční proměnné, u kterých nechceme serializovat jejich hodnoty, označíme override
- c) Třída (a její potomci) InputStream slouží pro práci s binárními streamy.
- d) Třídy pro práci se streamy se skládají pomocí návrhového vzoru Kompozit.
- e) Třída streamu, která se používá pro serializaci, musí implementovat rozhraní Serializable

### 8) Vyberte pravdivé tvrzení o ternárním operátoru:

- a) Ternární operátor je zkrácením konstrukce if else.
- b) Všechny parametry ternárního operátoru musí být numerického typu.
- c) Ternární operátor musí mít jako druhý parametr výraz typu boolean.
- d) Ternární operátor se používá k bitovým manipulacím s pamětí.
- e) Výsledkem ternárního operátoru je jediná hodnota.

Ternární operátor

(condition)? result 1: result 2;

### 9) Vyberte pravdivé tvrzení:

- a) Z konstruktoru můžeme volat jiný konstrukt (téže třídy) pomocí this(). Toto volání se nesmí vyskytovat za jinými příkazy v konstrukturu
- b) Pokud neuvedeme modifikátor viditelnosti (napr. public) tak je vše private.
- c) Pro porovnání dvou řetězců se používá ==, protože to jsou hodnoty primitivního typu.
- d) Blok final se používá za bloky catch.
- e) Datový typ char má délku 8 bitů.

### 10) Vyberte pravdivá tvrzení o konstrukci for(<Type>i : <var>) <statement>; :

- a) Toto konstrukce vyžaduje, aby proměnná <var> byla typu Iterable nebo pole.
- b) Toto konstrukce vyžaduje, aby proměnná <var> byla typu Iterator.
- c) Toto konstrukce vyžaduje, aby proměnná i byla vždy primitivního typu, protože pracuje s poli.
- d) Toto konstrukce vyžaduje, aby jako <Type> bylo použit typ Collection, nebo pole.
- e) Toto konstrukce vyžaduje, aby jako <Type> byl použit Comparable.

### 11) Vyberte všechna pravdivá tvrzení (co platí v jazyce Java) :

- a) Datový typ byte je to samé co datový typ char.
- b) Datový typ float zabírá vždy 16 bitů.
- c) Datový typ String je definován jako třída.
- d) Datový typ void se používá při deklaraci metod, pokud metoda nevrací žádnou hodnotu.
- e) Datový typ int zabírá podle použitého operačního systému 64 nebo 32 bitů