Výpočetní technika a lékařská informatika

Data

Mgr. Markéta Trnečková, Ph.D.

Data, informace a znalost

- **Data**
  - Nezpracované, samostatné hodnoty bez kontextu.
  - Krevní tlak: 140/90, věk: 67, diagnóza: E11.
- **Informace**
  - Data ve smysluplném kontextu, kterým rozumíme.
  - Pacient s diabetem 2. typu má hypertenzi a je mu 67 let.
- **Znalost**
  - Interpretace informací na základě zkušeností nebo pravidel.
  - Starší pacienti s diabetem a hypertenzí mají vyšší riziko cévní mozkové příhody. Je třeba upravit léčbu.

Znalost = informace aplikovaná v praxi.
Např. data z monitoru životních funkcí jsou pro sestru informace – ale rozhodnutí volat lékaře, když saturace klesá pod 92 %, je využití znalosti.

Data, informace a znalost
Příklad

Příklad

Rozhodněte, zda se jedná o data, informaci, nebo znalost (a proč).

- Teplota: 38,5 °C
- Pacient má horečku
- Horečka u tohoto pacienta může být projevem infekce, proto je nutné provést krevní testy
- Pacient má akutní infarkt myokardu
- Diagnóza: I21
- Pacienta s infarktem je třeba neprodleně převést na katetrizační sál

Data, informace a znalost
Příklad

Příklad

Rozhodněte, zda se jedná o data, informaci, nebo znalost (a proč).

- Teplota: 38,5 °C → data
- Pacient má horečku → informace
- Horečka u tohoto pacienta může být projevem infekce, proto je nutné provést krevní testy → znalost
- Pacient má akutní infarkt myokardu → informace
- Diagnóza: I21 → data
- Pacienta s infarktem je třeba neprodleně převést na katetrizační sál → znalost

Data, informace a znalost

Příklad na seminář

Příklad

„Pacient, 75 let, diabetik, TK 160/100, diagnóza E11.“

Určete, co je zde data, co je informace, a co lze odvodit jako znalost.

Data, informace a znalost

Příklad na seminář

Příklad

„Pacientka, 62 let, DM2, hypertenze, BMI 33. Laboratoř: glykemie nalačno 9,1 mmol/l, TK 155/95 mmHg, celkový cholesterol 7,2 mmol/l. Užívá metformin, ale nepravidelně.“

Úkol:

1. Identifikujte data
2. Převeďte je na informaci
3. Formulujte znalost – jaký je závěr / doporučení
4. Diskutujte, jak by tato znalost byla zapsána nebo využita v informačním systému

Kódování informací

- Ve zdravotnictví není možné zaznamenávat vše jen slovně. Kódování informací znamená převod pojmů do standardizované podoby, která umožňuje:
  - elektronické zpracování,
  - srovnání dat mezi zařízeními,
  - vyúčtování péče,
  - vědecké analýzy,
  - interoperabilitu mezi systémy.
- Cílem kódování je přesnost, strojová čitelnost, srozumitelnost mezi systémy a zeměmi.

Kódování informací

Příklady používaných klasifikací a číselníků

- **MKN-10 (ICD-10)**
  - Mezinárodní klasifikace nemocí (diagnózy).
  - E11 = diabetes mellitus 2. typu
- **DRG (Diagnosis Related Groups)**
  - Skupiny diagnóz pro účely úhrad a ekonomiky.
  - B03Z = Péče o pacienta s komplikovanou pneumonií
- **SNOMED CT**
  - Systematized NOmenclature of MEDicine Clinical Terms
  - Detailní klinická terminologie (symptomy, procedury, léky aj.) -– podporuje klinické IS.
  - 386661006 = Bolest na hrudi
- **ICZ/IČZ (Identifikátor zdravotnického zařízení)**
  - Jednoznačný kód zařízení v ČR.
  - 89301000 = Fakultní nemocnice Olomouc
- **CŽV, OKRUH, ODBOR (číselníky pro vykazování výkonů)**
  - Označení oddělení, specializace.
  - 101 = všeobecné praktické lékařství

Reprezentace dat

- Data musí být zapsána tak, aby je počítač mohl uložit, přenést a zpracovat.
- Počítače pracují pouze s binárními daty (0 a 1).
- Lidské informace (např. jméno, teplota, diagnóza) je třeba kódovat:
  - texty → znaky (např. kódování UTF-8)
  - čísla → číslicové reprezentace,
  - obrázky → bitmapy (matice čísel),
  - diagnózy → číselníky (např. MKN-10: I10 = hypertenze).
- **Příklad:** „Teplota 38,5 °C“ je pro počítač např. číslo 38.5 typu `float` ve formátu IEEE 754.

Informace více teoreticky

- Jednotka informace **bit** (1 b) – dvě různé hodnoty (0 a 1)
- Osminásobek jednoho bitu = **byte** (1 B)
- 1 byte = 256 různých hodnot ($2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^8 = 256$)
- hodnoty $0, \ldots 255$
- data v počítači – binární reprezentace
- data:
  - znaky abecedy
  - celá čísla
  - necelá čísla (čísla s desetinnou čárkou)
  - řídící a speciální znaky
- vše kódujeme do posloupnosti 1 a 0

Informace více teoreticky

Celá čísla

- Obecně se pro zápis čísel používají tzv. **číselné soustavy**
  - číslo dané soustavy je posloupností symbolů, které se nazývají číslice (nebo cifry),
  - každá číselná soustava je určena **základem** z, což je nenulové přirozené číslo, které udává maximální počet použitelných číslic,
  - skutečná hodnota každé číslice je pak dána pozicí ve zmíněné posloupnosti symbolů.
- číslo A v číselné soustavě o základu z můžeme napsat jako posloupnost
  $A = a_n a_{n-1} a_{n-2} \dots a_1 a_0$
  kde $a_n a_{n-1} a_{n-2} \dots a_1 a_0$ jsou jednotlivé číslice čísla A, přičemž a_n je nejvýznamnější číslice a a_0 je nejméně významná číslice,
- hodnota čísla A se pak určí jako součet mocnin základu, které jsou vynásobené jednotlivými číslicemi:
  $A = a_n \cdot z^n + a_{n-1} \cdot z^{n-1} + a_{n-2} \cdot z^{n-2} + a_1 \cdot z^1 + a_0 \cdot z^0$

Příklad

Vysvětlete číslo 1234 zapsané v soustavě o základu 10.
$(1234)_{10}$

Informace více teoreticky

Celá čísla

Příklad

Jaké číslo se skrývá v $(101010)_2$?

Informace více teoreticky

Celá čísla

- Záporná čísla:
  - Nejvýznamnější bit je vyhrazený pro znaménko
  - Doplňkový kód – záporné číslo je zaznamenáno jako binární negace (záměna všech 0 za 1) původního čísla zvětšená o 1.
- Desetinná čísla:
  - S pevnou řadovou čárkou
  - S pohyblivou řadovou čárkou

Informace více teoreticky

Kódování znaků – ASCII tabulka

b_7 →				0	0	0	0	1	1	1	1
b_6 →				0	0	1	1	0	0	1	1
b_5 →				0	1	0	1	0	1	0	1
Bits	b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Row ↓ / Column →	0	1	2	3	4	5	6	7
	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	(	8	H	X	h	x
	1	0	0	1	9	HT	EM	)	9	I	Y	i	y
	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	11	VT	ESC	+	;	K	[	k	{
	1	1	0	0	12	FF	FC	,	<	L	\	l	|
	1	1	0	1	13	CR	GS	-	=	M	]	m	}
	1	1	1	0	14	SO	RS	.	>	N	^	n	~
	1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Informace více teoreticky

Kódování znaků – české znaky

- pro znaky české abecedy (východoevropské/středoevropské jazyky):
  - ISO 8859-2 (ISO Latin 2) – standard ISO, používaný v UNIXových operačních systémech (OS),
  - Windows 1250 (CP1250) – kód firmy Microsoft, používaný v OS MS Windows, od ISO 8859-2 se liší např. ve znacích š, t, ž,
  - Mac CE – kód firmy Apple, používaný v Apple MAC OS,
  - CP852 (PC Latin 2)– kód firmy IBM, používaný v OS MS DOS,
- **Unicode**
  - každý znak v Unicode má jednoznačný číselný kód a svůj název.
  - prvních 128 znaků (tj. sedmibitové kódy) obsahuje znakovou sadu ASCII
  - UCS = otevřená množina pojmenovaných znaků všech abeced a kombinovaných znaků (např. diakritických), v současnosti (2015) 120 737 znaků ve 129 písmech (poslední verze 8.0 z roku 2015), znaky jen přidávány, prostor pro více než milion znaku
  - UTF-8, UTF-16, UTF-32

Informace více teoreticky

- Dosud jsme pracovali s jednoduchými daty – jednotlivými hodnotami
  - je důležité vědět, **jakého typu** data jsou (číslo, text, logická hodnota...)
  - operace mají smysl jen pro určitý datový typ (např. teploměrem změříme teplotu, ale těžko s ním změříme váhu)
- Jak uchovávat více hodnot najednou?
- Abstraktní datové typy:
  - definují logiku práce s daty (jak s nimi pracujeme – jak je vkládáme, odebíráme a pod.)
  - neřeší se, jak je to implementováno
  - příklad: zásobník, fronta, seznam, ...
- Konkrétní datové struktury:
  - popisují jak se data fyzicky ukládají a organizují v paměti
  - příklad: pole, strom, halda, ...

Abstraktní datové typy

Zásobník (stack)

- LIFO (last in, first out) struktura
- Nový prvek vždy přidáváme na vrchol, a odebíráme také z vrcholu
- Přirovnání: stoh táců v menze – vždy bereme ten, který je nahoře, nové přidáváme nahoru

Příklad

Uveďte nějaký příklad z praxe.

Ukázka: `https://stack-visualizer-wheat.vercel.app/`

Abstraktní datové typy
Fronta (queue)

- FIFO (first in, first out) struktura
- Prvek, který přidáme první, je také první odebrán
- Přirovnání: fronta v obchodě – první příchozí je první u pokladny

Příklad

Uveďte nějaký příklad z praxe.

Abstraktní datové typy
Prioritní fronta (priority queue)

- Stejný princip jako fronta, ale každý prvek má **prioritu**
- Prvek s vyšší prioritou je obsloužen dříve než ten, který přišel před ním
- Přirovnání: čekárna na pohotovosti – akutní případy jdou na řadu před méně závažnými

Příklad

Uveďte nějaký příklad z praxe.

Abstraktní datové typy
Příklad

Příklad

Jakou datovou strukturu byste použili pro následující případy:

- Výdej jídla v menze.
- Farmaceut připravuje sérii kroků pro výrobu léku. Potřebuje během práce kontrolovat, jaký krok udělal naposledy.
- Čekárna u praktického lékaře.

Abstraktní datové typy
Seznam (list)

- Uspořádaná kolekce prvků
- Přístup podle pořadí (indexu) – např. 1., 2., 3. položka
- Může obsahovat duplicity
- Příklad: seznam pacientů v péči lékaře (přistupujeme k nim např. podle rodného čísla)

Abstraktní datové typy
Množina (set)

- Kolekce unikátních prvků (žádné duplicity)
- Pořadí prvků není důležité
- Příklad: příznaky pacienta (horečka, kašel, bolest hlavy, . . .)

Abstraktní datové typy
Mapa (map, dictionary)

- Kolekce dvojic (klíč, hodnota)
- Každý klíč je jedinečný, hodnota může být libovolná
- Rychlý přístup k hodnotě podle klíče
- Příklad: rodné číslo pacienta $\rightarrow$ zdravotní karta

Abstraktní datové typy
Procvičení

Příklad

Student má napsaný rozvrh přednášek v přesném pořadí během dne. Jaká struktura je pro něj nejvhodnější?

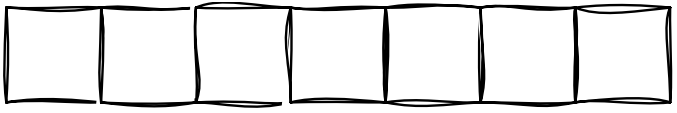
Abstraktní struktury

- jak jsou data doopravdy ukládána např. v počítači
- každý abstraktní datový typ je vhodné ukládat v paměti různě – kvůli efektivitě operací
- **Příklady**
  - pole
  - spojový seznam
  - strom
  - ...

Abstraktní struktury

Pole (array)

- Pevně daná velikost
- Prvky jsou uloženy za sebou v paměti
- Rychlý přístup podle indexu (např. 5. pacient v seznamu)
- Nevýhoda: obtížné vkládání a mazání uprostřed

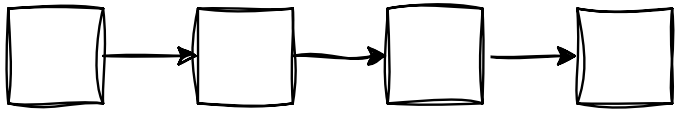


Ukázka implementace fronty pomocí pole:
`https://www.cs.usfca.edu/~galles/visualization/QueueArray.html`

Abstraktní struktury

Spojový seznam (linked list)

- Prvky (uzly) obsahují data a ukazatel na další prvek
- Velikost se může dynamicky měnit
- Snadné vkládání a mazání kdekoliv
- Nevýhoda: pomalý přístup k prvku podle pořadí (musíme procházet od začátku)

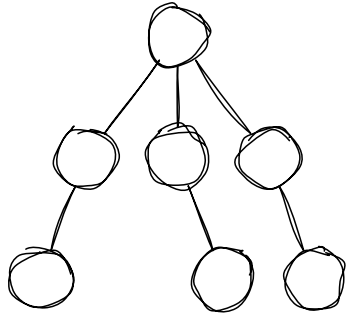


Ukázka implementace fronty pomocí pole:
`https://www.cs.usfca.edu/~galles/visualization/QueueLL.html`

Abstraktní struktury

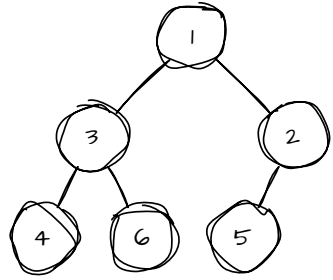
Stromy (trees)

- Hierarchická struktura s kořenem a potomky
- Každý prvek (uzel) může mít více poduzlů
- Příklady použití:
  - rodokmen pacienta
  - klasifikace nemocí (ICD kódy)
  - adresářová struktura ve zdravotnickém systému



Abstraktní struktury
Halda (heap)

- Stromová struktura s uspořádáním podle priorit
- Vždy snadný přístup k největšímu nebo nejmenšímu prvku
- Časté použití:
  - plánování operací podle naléhavosti
  - systém triáže v nemocnici

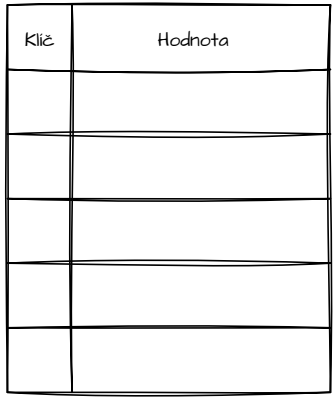


min heap

Abstraktní struktury

Hashovací tabulka (hash table)

- Klíč $\rightarrow$ hodnota
- Hashovací funkce určí, kam se prvek uloží
- Rychlé vyhledávání podle klíče (např. rodné číslo pacienta $\rightarrow$ zdravotní karta)
- Nevýhoda: kolize (dva klíče se mapují na stejné místo)



Hashovací tabulka
Příklad

Klíč = rodné číslo, Hodnota = Jméno pacienta

Pacienti:

- 820305/5678 → Petra Svobodová
- 750101/1234 → Jan Novák
- 900707/9012 → Martin Dvořák
- 830405/5678 → Anna Malá

Jednoduchá hashovací funkce: hash(key) = (součet všech číslic rodného čísla) mod 10

Výpočet pro Petra Svobodová:
$8 + 2 + 0 + 3 + 0 + 5 + 5 + 6 + 7 + 8 = 44 \quad \Rightarrow \quad 44 \bmod 10 = 4$

Hashovací tabulka

Příklad

Tabulka:

HASH	RČ	Jméno
0		
1		
2		
3		
4	820305/5678	Petra Svobodová
5		
6		
7		
8		
9		

Příklad

Přidejte do tabulky zbylé pacienty.

Hashovací tabulka
Příklad

Řešení kolize: Řetězení (chaining)

- U Anny Malé došlo ke kolizi
- Každý řádek tabulky
- Pokud dojde ke kolizi, nový prvek se přidá do seznamu
- Hledání $\rightarrow$ projdeme seznam v daném řádku

Procvičení

Příklad

Máte databázi léků, kde každý lék má unikátní kód. Jakou strukturu byste zvolili pro rychlé vyhledávání podle kódu?

Příklad

Studenti stojí v dlouhé řadě na menzu, ale občas přijde vyučující, který jde před ně. Jakou strukturu tato situace připomíná?

Příklad

V nemocničním systému potřebujete uchovávat klasifikaci nemocí podle kapitol a podkapitol. Jaká struktura by byla nejvhodnější?

Procvičení

Příklad

K implementaci prioritní fronty se používá halda (binární strom).
Na `https://priority-queue-visualizer.vercel.app/`
můžeme vidět vizualizaci, jak pracuje.
Nejprve haldu vyprázdněte a pak vložte pacienty (pacient – priorita):

- Jan Novák – střední
- Petra Svobodová – vysoká
- Martin Dvořák – nízká
- Anna Malá – vysoká

Podívejte se, jak se halda tvoří a jak jsou prvky (pacienti) z fronty odebíráni.

Příklad

Co by se stalo, kdyby všichni pacienti měli stejnou prioritu?

Příklad

Příklad

Jsme na urgentním příjmu. U každého pacienta chceme uchovávat Jméno, Čas příchodu, Prioritu.
Máme tyto pacienty.

- Jan Novák, 2, 08:05
- Petra Svobodová, 1, 08:10
- Tomáš Dvořák, 3, 08:15
- Eva Králová, 2, 08:20
- Lukáš Malý, 1, 08:25
- Martina Černá, 3, 08:30
- Pavel Novotný, 2, 08:35
- Hana Procházková, 1, 08:40

Vytvořte pro ně nějakou „strukturu“ (dokument), kde je budeme uchovávat.

Příklad

Příklad

Nyní

- Zavolejte pacienta do ordinace.
- Zavolejte dalšího pacienta.
- v 9:00 přišel pacient Tomáš Marný a byla mu přiřazena priorita 2. Přidejte ho do svých dat.
- v 9:05 přišla pacientka Hana Veselá a byla jí přiřazena priorita 1. Přidejte ji do svých dat.
- Zavolejte dalšího pacienta.
- ...

Zamyslete se, jestli jste zvolili vhodné ukládání dat.

Data

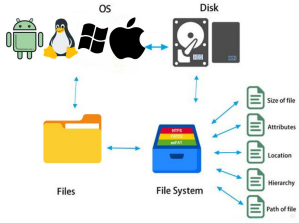
- data vs. informace
- Data
  - údaje získané pozorováním, nebo měřením
  - představují fakta, text, obraz, zvuk, video, nejčastěji v kontextu sledovaného procesu nebo situace
  - nezávislá na uživateli, většinou odráží současný stav reality
- Informace
  - informací se data a vztahy mezi nimi stávají vhodnou interpretací pro uživatele vytvořením struktur, které odhalují uspořádání, vzory, tendence a trendy
  - strukturovaná, organizovaná, shrnutá a interpretovaná data, silně závislé na tom, kdo je požaduje, tedy závislé na uživateli (individuální hloubka znalostí, zkušeností . . .)

Kartotéka

- ukládání dat pomocí karet
- nevýhody:
  - složité vyhledávání
    Potřebuji najít všechny pacienty s diabetem.
  - obtížná změna dat
    Změnil se název pojišťovny.
  - nadbytečnost dat
    U všech pacientů z Olomouce uložené stejné PSČ.
  - složité získávání informací
    Potřebuji pacienty, kteří byli nemocní více jak 10×.

Souborový systém

- pro každou oblast samostatný soubor – pacienti, lékaři, léky
- jednodušší organizace dat a vyhledávání
- nevýhody:
  - nekonzistence = různé verze dat v různých souborech
  - datové anomálie
    - modifikace – změna je potřeba udělat u všech výskytu dat (i v různých souborech)
    - vkládání – pokud vkládáme např. novou fakturu, musíme k ní přiřadit klienta, který může být nový nebo již založený v souboru klientů = možnost způsobení nekonzistence
    - mazání – při smazání klienta mohou zůstat faktury bez klienta



Databázový systém

- databáze
- **Skládá se z**
  - data – syrová fakta z oblasti zájmu uživatele
  - metadata — popisují data a vztahy mezi daty v databázi
  - programové vybavení – Systém řízení báze dat (SŘDB) = kolekce programů, která:
    - řídí databázovou strukturu
    - kontroluje přístup k datům uloženým v databázi
    - umožňuje přístup více uživatelům do databáze
    - pomáhá k efektivnější správě dat
  - technické prostředky (hardware)
  - uživatelé
- více se databázím budeme věnovat příště