

Jak psát (a nepsat) technickou dokumentaci

Ekštein & Pártl

Proč je třeba umět psát technickou dokumentaci?

- škola (předměty, kde se vyžaduje precizní dokumentace SW díla → KIV/PC, KIV/ZSW, KIV/BIT, KIV/WEB, KIV/UPS, KIV/ZOS, ...)
- bakalářská/diplomová práce
- firemní praxe (každý programátor ve firmě píše k SW také dokumentace, podklady pro výroční zprávy, ...)
- **"vizitka inženýra"** – písemný projev vysokoškolsky vzdělaného odborníka musí mít určitou úroveň, nelze tolerovat praktickou negramotnost
 - výmluva „šel jsem studovat počítače, abych už se nemusel trápit s češtinou“ je naprostý blábol, vysokoškolák v jakémkoliv oboru **musí** ovládat svůj rodný jazyk dokonale

2. CHARAKTERISTIKA VYBRANÝCH TYPŮ

2.1. Výběr základních typů

2.1.1. Základní typy

Celý operační systém se skládá z několika set typů souborů. Takových souborů je v počítači několik set tisíc. Podle přípony můžeme rozpoznat o jaký soubor se jedná a jak s ním pracovat. V počítači pracujeme pouze s těmi, které programy využívají nejčastěji, aby mohly správně fungovat.

2.1.2. Rozdělení

Soubory rozdělujeme podle několika kritérií. Můžeme je rozdělit podle velikosti, komprimace nebo využití funkce.

Soubory:

- Zvukové – a) hudební
b) mluvené slovo
- Zálohovací
- Binární

2. CHARAKTERISTIKA VYBRANÝCH TYPŮ

2.1. Výběr základních typů

2.1.1. Základní typy

Celý operační systém se skládá z několika set typů souborů. Takových souborů je v počítači několik set tisíc. Podle přípony můžeme rozpoznat o jaký soubor se jedná a jak s ním pracovat. V počítači pracujeme pouze s těmi, které programy využívají nejčastěji, aby mohly správně fungovat.

Nedává smysl.

2.1.2. Rozdělení

Soubory rozdělujeme podle několika kritérií. Můžeme je rozdělit podle velikosti, komprimace nebo využití funkce.

Soubory:

- Zvukové – a) hudební
b) mluvené slovo
- Zálohovací
- Binární

TOTÉŽ

Co je špatně? – v tomto případě prakticky vše:

- užití fontů
- odsazování nadpisů
- kurzíva v nadpisech
- nesmysly v textu
- nesmyslná struktura textu (nadpis „Soubory”)

Komunikační strategie v případě technické dokumentace

- cílová skupina čtenářů: **experti** (minimálně určité poučení/znalí laici)
- lze předpokládat určitou úroveň znalostí
- cílem komunikace je **popis**, předávání znalostí, zkušeností, představení závěrů vědeckého zkoumání určitého problému
 - nikoliv přesvědčovat, navozovat pocity
 - ani ohromovat estetikou sdělení ("technická lyrika")
- podstatná je **objektivita sdělení**
 - subjektivní či emocionálně zabarvený text je nepřipustný
 - domněnky jsou obecně také nepřipustné (mohou se vyskytnout v diskusi či závěru práce v podobě tzv. kvalifikovaných odhadů)

všechny kombinace se musí „něco“ provést a předat mi správnou hodnotu 0 nebo 1. Oindexoval jsem si tabulku zleva čísla 0 až 15 a shora čísla 0 až 3. Přemýšlím. Nic mě nenapadá. A náhle. Jak blesk z čistého nebe. Ta myšlenka je tu. Heuréka. Je to tak jasné. 0 - 0000, 1 – 0001, 8 – 1000, ... Ovšem rázem mé nadšení opadlo. Jak přečíst jeden bit? Začal jsem přemýšlet. Hledal v knihách. Nic nepřicházelo. Žádná rada. Nic.

A náhle, cosi tu bylo

Bitové pole - zdálo se jak krásný sen

Avšak záhy velký problém byl mnou odhalen

Bitové pole - tot' krásná věc

Ovšem k ničemu není mi přec

Netrvalo dlouho a přišel jsem na to, že tudy cesta nevede. Jak přečíst jeden bit? Tot' otázka. Přemýšlím. Vždyť v práci s bity pracujeme celkem běžně. Jakže je to vlastně používáme. Aaa – ano! Bitový součin a bitový součet. Co mi to vlastně říká? Mám číslo. Pořadí kombinace. Jak z něj dostanu hodnotu daného bitu? Ano. Pokud budu mít správné číslo, pak bitový součin bude větší než nula právě tehdy, když bude příslušný bit 1. Tedy, rozšířím si strukturu pole proměnných o proměnnou, do

všechny kombinace se musí „něco“ provést a předat mi správnou hodnotu 0 nebo 1. Oindexoval jsem si tabulku zleva čísla 0 až 15 a shora čísla 0 až 3. Přemýšlím. Nic mě nenapadá. A náhle. Jak blesk z čistého nebe. Ta myšlenka je tu. Heuréka. Je to tak jasné. 0 - 0000, 1 - 0001, 8 - 1000, ... Ovšem rázem mé nadšení opadlo. Jak přečíst jeden bit? Začal jsem přemýšlet. Hledal v knihách. Nic nepřicházelo. Žádná rada. Nic.

A náhle, cosi tu bylo
Bitové pole - zdálo se jak krásný sen
Avšak záhy velký problém byl mnou odhalen
Bitové pole - tot' krásná věc
Ovšem k ničemu není mi přec

KEY

Netrvalo dlouho a přišel jsem na to, že tudy cesta nevede. Jak přečíst jeden bit? Tot' otázka. Přemýšlím. Vždyť v práci s bity pracujeme celkem běžně. Jakže je to vlastně používáme. Aaa - ano! Bitový součin a bitový součet. Co mi to vlastně říká? Mám číslo. Pořadí kombinace. Jak z něj dostanu hodnotu daného bitu? Ano. Pokud budu mít správné číslo, pak bitový součin bude větší než nula právě tehdy, když bude příslušný bit 1. Tedy, rozšířím si strukturu pole proměnných o proměnnou, do

Vědecká metoda poznání

- **nikdy netvrdit nic, co nelze dokázat**
 - dokázat lze **korektní** (nezkreslenou a správně provedenou) statistickou analýzou dat, matematickým důkazem, citací z důvěryhodného zdroje

Základem poznání vědeckou metodou je:

1. **observace**/sběr faktů
2. **analýza** problému/úlohy
3. **návrh** hypotéz
4. **testování** hypotéz (srovnání predikce hypotézou s pozorováním)
5. formulace **závěrů**

Technická čeština v odborných dokumentech

- gramaticky správný český jazyk
 - jak se správně časuje sloveso "být" v přítomném kondicionálu?
- slovník může být ovlivněn oborem/oblastí/tématem – v IT lze tolerovat užití anglicismů, ale v rozumné míře, a jen takové, které už jsou považované za ustálené ("software", "hard disk", "debugger")
- v žádném případě slangové výrazy či novotvary ("fíčura", "heš mapa", "tabulka labelů", "keyrooty")
- zcela nepatřičné jsou subjektivní hodnotící výrazy ("trochu", "poměrně", "relativně", "velice", "obrovské", "víceméně", "o poznání horší", "uspokojivý", ...)
- nepoužívat neexistující slova – ač se vyskytují např. v médiích ("potencionální", "nej/optimálnější", "determinističtější" ...)

Technická čeština v odborných dokumentech

- nepoužívat slova a spojení, která sice existují, ale v daném kontextu nedávají smysl nebo znamenají něco jiného ("demonstrativní aplikace", "klasické pole / obyčejné pole")
- vyhnout se archaickým, nářečným či "rodinným" obratům ("škandál", "jednak / druhak / třeták", "anžto", "kterýtož")

Legendární výroky studentů, které nemají v dokumentaci co dělat

- „Program netvoří memory leaky a vykazuje 0 errorů.”
- „Tato funkce je srdcem celého algoritmu.”
- „Toto pole jsem pojmenoval mask, protože jsem si z počátku myslel, že se tak má jmenovat.”
- „Našel by se prostor pro další vylepšení programu. Některé části by se daly naprogramovat efektivněji. Jen si nejsem jistý kde, jaké a hlavně jak.”

Legendární úseky textu, které nedávají smysl

- „Tento obrázek je reprezentován dvěma barvami, a to bílou a černou.”
- „Aplikace má několik funkcionalit.”

Legendární gramatické chyby

- "... vypočítali by jsme ..."
- "... data vykazovaly chyby ...", "... data vykazovali chyby ..."
 - **obecně shoda podmětu s přísudkem stále působí problémy**
- "analíza", "algorytmus", "problémi"
- "... v plné výšši ..."
- **interpunkce**
 - "..., a tedy/tak/to"

Sazba (DTP) technického dokumentu obecně

- **není to triviální záležitost** – má svá pravidla, z nichž některá jsou stovky let stará (tradiční postupy, léty prověřená estetika) a mají hluboký smysl
- **používat vhodné nástroje** (L^AT_EX, Adobe *) – zajistí dodržení většiny typografických pravidel a zvyklostí samy, bez zásahu uživatele (některé více, některé méně)
- **nepoužívat nevhodné nástroje** (MS Word, MS Malování) – podporují autora dokumentu ve vytváření prasáren či tzv. "letáků do cirkusu"
- preferovat **konzervativní přístup** k sazbě, nevymýšlet nic nového, moderního, "fancy" (pokud na to člověk nemá vzdělání a talent, dopadne to katastrofálně)

Sazebné postupy specifické pro český jazyk

- české uvozovky
- neslabičné předložky

"k", "s", "v", "z" **nesmí**

stát samostatně na konci

řádku – a raději ani jednopísmenné spojky a předložky "a", "i", "o", "u"

(zajišťuje svázání tzv. **nezlomitelnou/nedělitelnou mezerou**)

- pozor na **dělení slov**, kterým může vzniknout vulgarismus
(kni-hovna, nanose-kunda, Mo-ser)

„Uvozovky.“

Užití různých mezer v sazbě

- obyčasná (**MEZERNÍK**) – za všemi jednoduchými interpunkčními znaménky
- **pevná, nezlomitelná/nedělitelná** – na rozdíl od pevné může měnit šířku, váže neslabičné předložky k následujícímu slovu, dále v řadě dalších situací »»»»
- **čtverčík** (= quad, em) – relativní jednotka délky odvozená od stupně písma
- mezery mohou být různých šířek:
 - **čtverčíková** (em, quad)
 - **půlčtverčíková** (en)
 - **čtvrtčtverčíková** (qqquad)
 - **úzká** (thin)

KDYŽ NEVÍM

<https://prirucka.ujc.cas.cz/?id=785>

Nezlomitelná mezera (nbsp = non-breakable space)

- **neslabičné předložky** (a jednopísmenná slova)
- **datum** – 21. ■ 10. ■ 2022 | 23. ■ března 2001
- **fyzikální jednotky** – 4,3 ■ Ghz | 158 ■ MPa | 100 ■ Mbit · s⁻¹
- **symboly měn** – 20,- ■ Kč | 450 ■ US\$ | 1 000 ■ €
- **zkratky** – spol. ■ s ■ r. ■ o., s. ■ r. ■ o., a. ■ s., ...
- **akademické a vědecko-pedagogické tituly, hodnosti**
 - Ing. ■ Josef Baloun, Doc. ■ MUDr. ■ Jan Kozel, ■ CSc., pplk. ■ Jaroslav Šídlo
- dle vlastního uvážení tam, kde by buď zůstalo něco divného na konci řádku nebo na začátku následujícího

Rozdělovník, spojovník, pomlčka, mínus

- ačkoliv na klávesnici je jen klávesa mínus (■), typografie rozlišuje několik různých znamének – liší se délkou a polohou v písmové osnově

5 – 3, spo-jov-ník

pomlčka – rozdělovník (divis) —

Sazba zkratk a jejich použití v textu

- ve formálním textu je lepší se zkratkám vyhýbat
- zkratka je obvykle ukončena tečkou '.', pokud nestojí na konci věty (2 tečky za sebou jsou přípustné)
- používat jen běžné, známé zkratky (mj. | tj. | atp. | atd. | apod.)
 - atd. = a tak dále, tj. nepíše se před ní čárka ("pivo, víno, rum atd.")
- **"viz" není zkratka**
- "vs" (versus) v češtině neužíváme
- některé zastaralé zkratky je také lepší spíše neužívat ("dtto")

Parchanty: Vdovy a sirotci

- technické typografické chyby v rozvržení textu na stránce
- kvalitní DTP systémy brání jejich vzniku automaticky

sirotek – stránka začíná posledním (často podplněným) řádkem odstavce, poslední řádek odstavce nenavazuje na text odstavce na stejné stránce, ale je samostatně na začátku další stránky

vdova – stránka končí prvním řádkem nového odstavce, text odstavce nenavazuje na první řádek na téže straně, ale na další, takže první řádek stojí osamoceně na konci předchozí stránky

Řezy, opakování slov, mezislovní mezery

- kombinace zarovnání do bloku a absence dělení slov v méně kvalitních editorech vede ke vzniku tzv. **řek**
- pozor na několikanásobný výskyt stejného slova v jediné větě, obzvláště když v důsledku řádkového zlomu vyjdou nad sebe
- při nuceném zarovnávání do bloku (full justify) mohou některé editory nadměrně zvětšovat mezislovní mezery, až za hranici estetické přijatelnosti (řeší dělení slov)

Řeka se
objevuje
v případě, že
má editor
potíže se
zarovnáním
textu (např.
do bloku)

Kapitola 4

Uživatelská příručka

4.1 Překlad programu

K programu je přiložen **makefile**, proto pro překlad stačí v daném adresáři se zdrojovými kódy a **makefile** spustit příkaz **make**, který program přeloží. Samozřejmě musí na daném operačním systému být nainstalován překladač **GCC** (a v operačním systému Windows správně nastaveny cesty k překladači v **PATH**). Pod operačním systémem Linux lze provádět několik příkazů s **make** navíc. Je možno program nainstalovat (**make install**) i odinstalovat (**make uninstall**) a tím být dostupný odkudkoliv v konzoli. Další možností je přímo odstranění přeložených souborů pomocí **make clean**.

4.2 Ovládání programu

Program se ovládá pomocí parametrů na příkazové řádce:

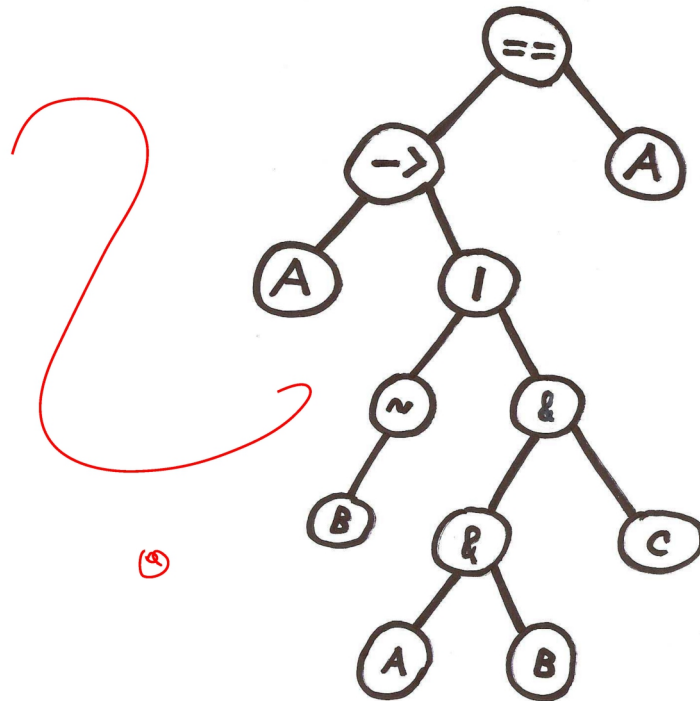
1. parametr - přepínač komprese/dekomprese. **-c** znamená, že vstupní soubor se bude komprimovat. **-d** znamená, že vstupní soubor je komprimovaný (ověřování správného souboru probíhá kontrolou hlavičky, která byla přidána při komprimaci) a bude se dekomprimovat;
2. parametr - cesta ke vstupnímu souboru, pokud neexistuje, vypíše se varovné hlášení a program skončí s nenulovým návratovým kódem;
3. parametr - cesta k výstupnímu souboru, pokud existuje, dotáže se program na jeho přepsání: Odpověď může být pouze **Y** - pak bude soubor přepsán, nebo **N** - tehdy program skončí s nenulovým návratovým kódem.

- kromě opakujících se slov se v ukázce vyskytují ještě další typografické chyby – jaké?

Další technické chyby sazby

- chybějící/přebývající mezery
- řeky, úplná absence nebo nevhodné dělení slov
- nevhodná/násilná změna rozpalu a/nebo prokladu a/nebo rozstrkání
- bitmapové obrázky v nízkém rozlišení, příp. s kompresními artefakty
- nesprávně provedené číslování stránek (např. číslo stránky na titulce)
- objekty vyčnívající mimo zrcadlo
- chybějící znaky ve fontu (některé fonty nemají např. matematické symboly nebo některé znaky národních abeced)

FVV!!!
 $A \rightarrow \sim B \mid A \& B \& C == A$



Ukázka stromu vytvořeného ze zadané formule.

Ukázka výsledné tabulky pravdivostních hodnot pro formulí na obrázku výše:

A	B	C	\$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

- používat **vhodné nástroje**
 - yEd, TikZ, GraphViz, draw.io, Inkscape, ...
- exportovat z nich do PDF nebo jiného **vektorového** formátu
- v krajním případě do bitmapy s rozlišením alespoň 300 DPI

Chyby při sazbě textu mimo odstavcový text

- **popisky tabulek a obrázků**
 - u tabulek vždy **nad**
 - u obrázků vždy **pod**
 - popisek končí tečkou jen pokud se jedná o celou větu
 - velikost písma buď stejná jako odstavcový text nebo o stupeň menší
- **důležitá je konzistence**
- obrázky, tabulky apod. **odkazovat z textu** – nesnažit se o umístění floatu na konkrétním místě dokumentu (umístění "násilím" často vede k rozbití textu)
 - „... na obrázku vidíme...“, ale obrázek je na jiné stránce

high, které odpovídají počtu iterací laplaceovského vyhlazování podle tabulky 7.1. Pro všechny úrovně vyhlazení je nastaveno na experimentálně stanovenou hodnotu $\lambda = \frac{1}{2}$. Křivky vzorových cviků jsou nejdříve vyhlazeny

úroveň vyhlazení	počet iterací
low	10
medium	20
high	30

Tabulka 7.1: Úrovně vyhlazování

laplaceovským vyhlazováním, což kromě vyhlazení každé křivky zajistí také pravidelnější rozložení bodů napříč křivkou. Z tohoto důvodu interpolace

Ukázku spuštění Vašeho programu naleznete na ukázce konzolového rozhraní 1 na straně 3.

Hotovou práci odevzdejte v jediném archivu typu ZIP prostřednictvím automatického odevzdávacího a validačního systému. Postupujte podle instrukcí uvedených na webu předmětu. Archiv nechť obsahuje všechny zdrojové soubory potřebné k přeložení programu, **Makefile** pro Windows i Linux (pro překlad v Linuxu připravte soubor pojmenovaný Makefile a pro Windows Makefile.win) a dokumentaci ve formátu PDF vytvořenou v typografickém systému T_EX (L^AT_EX). Bude-li některá z částí chybět, kontrolní skript Vaši práci odmítne.

Popis činnosti programu

Program zpracuje postupně zadané příkazy a aritmetické výrazy obsahující celočíselné operandy s neomezenou velikostí. Po zadání (ať už z konzole nebo načtením ze souboru) výraz vyhodnotí a jeho výsledek okamžitě vypíše na obrazovku. Chování interpretu (číselnou soustavu výsledků vyhodnocení aritmetických výrazů) bude možné změnit pomocí příkazu uvedených v tabulce 2 na straně 5. Minimální množinu unárních či binárních operátorů, které tento interpret musí zvládnout zpracovat, aby úspěšně prošel validátorem, je uveden v tabulce 3 na straně 6 (z vlastní iniciativy můžete naprogramovat i další operátory či funkce).

Ještě k popiskům...

- tabulky, obrázky, výpisy kódu apod. – tzv. **plovoucí objekty** – musí být vždy opatřeny číslovaným popiskem
- jiné objekty, např. matematické vzorce, příkazy apod. – tedy **neplovoucí objekty** –, je vhodné číslovat, a to i tehdy, když na ně není z textu přímý odkaz, kvůli případné recenzi (aby recenzent mohl napsat např. „Vzorec číslo (3) nedává smysl.”)

Číslování stránek (paginace)

- za normálních okolností (v jednoduchém dokumentu) je **první číslovanou stránkou první strana obsahu** (tzn. je to první stránka, kde je číslo, ale není tam číslo 1)
- je-li dokument složitěji strukturovaný, lze využít více číselných řad
 - front matter → římské číslice (i), (ii), (iii)
 - text dokumentu → arabské číslice – 1 –, – 2 –, – 3 –
 - přílohy → kombinace písmen a arabských číslic A–1, A–2, A–3
 - nebo jinak, ale musí to být konzistentní, estetické a dávat smysl

ZÁPADOČESKÁ UNIVERZITA



Semestrální práce z předmětu KIV/PC

Vyhodnocování logických formulí

Jméno: Jan Černý

Číslo: 140217

Rok: 2007 / 2008

- kromě číslované titulní stránky se v ukázce vyskytují ještě další typografické chyby – **jaké?**

Obsah

1. Zadání
2. Analýza úlohy
 - 2.1. Specifikace úlohy
 - 2.2. Očekávané problémy
 - 2.3. Zpracování proměnných
 - 2.4. Tvorba tabulky pravdivostních hodnot
 - 2.5. Odhalení chyb
3. Popis implementace
 - 3.1. Mechanismus fungování programu
 - 3.2. Rozpoznání proměnných
 - 3.3. Zpracování operátorů
 - 3.4. Využívání ostatních znaků
 - 3.5. Konverze do RPN
 - 3.6. Analýza získaných dat
 - 3.7. Přiřazení hodnot proměnným
 - 3.8. Výpočet hodnoty výrazu
 - 3.9. Tisk řešení
 - 3.10. Uvolnění zdrojů
4. Uživatelská příručka
 - 4.1. Přeložení programu
 - 4.2. Formát vstupu
 - 4.3. Interpretace výsledků
 - 4.4. Vlastnosti a omezení programu
5. Závěr

Sazba citací a seznamu použitých zdrojů

- poměrně složitá pravidla, norma ČSN ISO 690

... Objem komolého kuželu se vypočítá jednoduchou metodou, popsanou v (Bar06). ...

(Bar06) BARTSCH, Hans-Jochen. *Matematické vzorce*. Academia Praha, 2006.
ISBN 80-200-1448-9.

- neprovádět citace poznámkou
pod čarou (ač je to mnohdy
doporučováno)

KDYŽ NEVÍM

<https://www.citace.com>

6 Kapitola šestá: Zdroje

- Pavel Herout
Učebnice jazyka C (1. a 2. díl)
KOOP, České Budějovice 2009,

- webové stránky a fóra:

Programovací jazyk C:

<http://www.linuxsoft.cz/>,

<http://www.sallyx.org/sally/c/>,

<http://www.root.cz/vyvojarsky-software/>,

<http://rechneronline.de/function-graphs/>,

<http://www.root.cz/clanky/postscript-vyplnovane-cesty-transformace-a-natoceny-text/>,

<http://www.root.cz/clanky/valgrind/>,

<http://stackoverflow.com/>,

<http://www.cprogramming.com/debugging/valgrind.html>,

PostScript:

http://kmt.hku.nl/pieter/EDU/postscript/clinic/ps_grafiek.c,

<http://www.physics.emory.edu/weeks/graphics/graph.html>,

<http://www.adobe.com/products/postscript/pdfs/PLRM.pdf> .

Sazba citací pomocí nástroje BibTeX

- detaily publikace se zapíší do bibliografické databáze dokumentu spec. deklarativním jazykem:

```
@book{bar06,  
  title={{Matematické vzorce}},  
  author={{Bartsch, Hans-Jochen}},  
  isbn={80-200-1448-9},  
  year={2006},  
  publisher={{Academia Praha}}  
}
```

- nástroj pak provede sazbu jednotným, předem zvoleným stylem

Sazba matematických vzorců a fyzikálních veličin

- dodržovat **matematickou notaci** – nezapisovat vzorce např. tak, jak se píšou v nějakém programovacím jazyce, ale tak, jak se sázejí v matem. knihách (evergreen je používání znaků '*' a 'x' místo znaků násobení '×' či '·')
- **proměnné** pokud možno **jednopísmenné**, sázené pravou kurzívou, následně vysvětlit, co daná proměnná vyjadřuje
- hodnota veličiny je od symbolu veličiny oddělena mezerou, symbol se sází vzpřímeným řezem (**15Hz** → **15 Hz**)
- **sazba procent** – s mezerou nebo bez mezery podle významu („Pražská burza zaznamenala pokles o 17 %.” | „Vlastní 25% podíl ve firmě.”)

vynásobeny vahami jednotlivých senzorů o_i určených konfiguračním parametrem pro orientace.

$$d_{rot} = \alpha_1 * o_1 + \alpha_2 * o_2 + \alpha_3 * o_3 + \alpha_4 * o_4 \quad (3.4)$$

Celková vzdálenost aktuální pozice pacienta od vzorové pozice je poté určena podle vzorce 3.5 jako součet vzdáleností pozic a orientací.

$$d_{total} = d_{pos} + d_{rot} \quad (3.5)$$

$$d_i = \sum_{j=1}^4 |p_i^j, p_{i+1}^j| * c_{pos}^j \quad (3.6)$$

Z prohledaných bodů se nalezne ten, který je nejbližší (nejvíce se podobá) aktuální pozici trackerů, a vypočítají se hodnotící parametry. Jestliže se tento bod nachází na indexu i , je aktuální fáze vypočtena podle vzorce 3.7.

$$phase_{curr} = \frac{\sum_{j=0}^{i-1} d_j}{d_{total}} \quad (3.7)$$

$$maxTunnelDistance =$$

$$= \sum_{j=1}^4 \left((tunnelRadius + maxQualityDistance) * c_{pos}^j \right) + \sum_{j=1}^4 \left(\pi * c_{rot}^j \right) \quad (3.8)$$

popsán $N = 512$ vzorky diskrétního odhadu spektra. Každý z nich tedy popisuje energie v pásmech frekvencí s šířkou

$$\frac{f_{max}}{N} = \frac{4000}{512} = 7,8125 \text{ Hz.}$$

Podle jednoho z Eulerových vzorců, který je popsán vztahem (4.3) lze DFT přepsat do podoby (4.4) [17].

$$e^{i\varphi} = \cos \varphi + i \sin \varphi \quad (4.3)$$

$$S_k = \sum_{n=0}^{N-1} s_n \left[\cos \left(\frac{2\pi nk}{N} \right) - i \sin \left(\frac{2\pi nk}{N} \right) \right] \quad (4.4)$$

Vzorek diskrétního odhadu spektra má podobu $S_k = \text{Re}\{S_k\} + i \text{Im}\{S_k\}$, kde $\text{Re}\{S_k\}$, resp. $\text{Im}\{S_k\}$ je reálná, resp. imaginární složka komplexního čísla S_k . Číslo lze vizualizovat v Gaussově rovině, což ilustruje obrázek 4.7. Tato

Analýza úlohy:

Aritmetická komprese funguje podobně jako Huffmanův kód. Dosahuje podobných kompresních výsledků, v některých případech i lepších. Aritmetická komprese uloží soubor jako jedno veliký desetinný číslo v intervalu mezi <0;1). Tohoto dosáhne svým zvláštním algoritmem, například pokud budeme mít písmena R, C, S, ... tak tato metoda postupuje takto:

1. Znak R

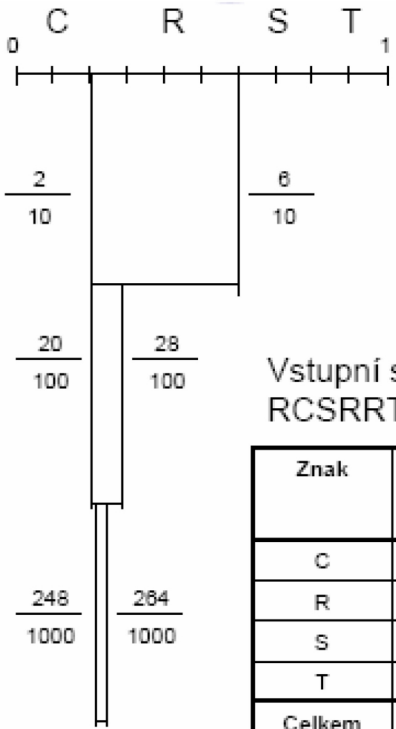
$$D_1 = D_0 + \frac{m_R}{n} \cdot L_0 = 0 + \frac{2}{10} \cdot 1 = \frac{2}{10}$$
$$H_1 = D_1 + \frac{n_R}{n} \cdot L_0 = \frac{2}{10} + \frac{4}{10} \cdot 1 = \frac{6}{10}$$
$$L_1 = H_1 - D_1 = \frac{6}{10} - \frac{2}{10} = \frac{4}{10}$$

2. Znak C

$$D_2 = D_1 + \frac{m_C}{n} \cdot L_1 = \frac{2}{10} + \frac{0}{10} \cdot \frac{4}{10} = \frac{20}{100}$$
$$H_2 = D_2 + \frac{n_C}{n} \cdot L_1 = \frac{20}{100} + \frac{2}{10} \cdot \frac{4}{10} = \frac{28}{100}$$
$$L_2 = H_2 - D_2 = \frac{28}{100} - \frac{20}{100} = \frac{8}{100}$$

3. Znak S

$$D_3 = D_2 + \frac{m_S}{n} \cdot L_2 = \frac{20}{100} + \frac{6}{10} \cdot \frac{8}{100} = \frac{248}{1000}$$
$$H_3 = D_3 + \frac{n_S}{n} \cdot L_2 = \frac{248}{1000} + \frac{2}{10} \cdot \frac{8}{100} = \frac{264}{1000}$$
$$L_3 = H_3 - D_3 = \frac{264}{1000} - \frac{248}{1000} = \frac{16}{1000}$$



Vstupní soubor:
RCSRRTCSRT

Znak	Počet znaků n_{znak}	Počet menších m_{znak}
C	2	0
R	4	2
S	2	6
T	2	8
Celkem	10	

Nejprve musí projít celý soubor a určit jednotlivé počty písmen (znaků), ze kterých pak vypočítá kódové značky pro horní a dolní mez (resp. délku intervalu). Pokud se komprimuje

Estetické chyby sazby technických dokumentů

#1 **podtrhané nadpisy**

#2 **podtrhaný text** – podtrhávání se v typografickém designu nepoužívá (resp. je to extrémní prostředek), ke zvýraznění textu se používá tzv. **zvýrazňovací řez**, což je **většinou pravá kurzíva** (méně často tučné písmo).

#3 **kombinace více fontů** (zejména takových, které se k sobě nehodí) v jediném dokumentu – normální stav je mít 2 – 3 fonty (odstavcový text, nadpisy)

#4 **použití nevhodných fontů**

- destrukční fonty, comic, dekorativní, rukopisné, futuristické, archaické (gotické, unciála), kaligrafické

KDYŽ NEVÍM

<https://www.ceskefonty.cz>