

Část 1: Základy

Kapitola 1

Vítejte při stavbě softwaru **31**

- 1.1 Co je to stavba softwaru? 32
- 1.2 Proč je stavba softwaru tak důležitá? 35
- 1.3 Jak číst tuto knihu 36

Kapitola 2

Metafory pro rychlejší pochopení vývoje softwaru **37**

- 2.1 Jak je důležité mít metafory 38
- 2.2 Jak používat softwarové metafory 40
- 2.3 Běžné softwarové metafory 41

Kapitola 3

Dvůkrát měř, jednou řež: Vstupní opatření **49**

- 3.1 Význam vstupních opatření 50
- 3.2 Jak určit povahu softwaru, na němž pracujete 56
- 3.3 Příprava na definici problému 61
- 3.4 Příprava definice požadavků 63
- 3.5 Příprava architektury 68
- 3.6 Čas věnovaný přípravám 78

Kapitola 4

Klíčová stavební rozhodnutí **83**

- 4.1 Volba programovacího jazyka 84
- 4.2 Programovací konvence 88
- 4.3 Vaše místo na technologické vlně 89
- 4.4 Volba hlavních stavebních postupů 91

Část 2: Tvorba vysoce kvalitního kódu

Kapitola 5

Návrh během stavby **95**

- 5.1 Návrhové úkoly 97
- 5.2 Klíčové návrhové koncepce 99
- 5.3 Návrhové stavební bloky: Heuristika 108
- 5.4 Návrhové postupy 130
- 5.5 Interpretace oblíbených metodik 138

Kapitola 6

Pracovní třídy 143

- 6.1 Základy tříd: Abstraktní datové typy 144
- 6.2 Dobrá rozhraní tříd 151
- 6.3 Problematika návrhu a implementace 161
- 6.4 Důvody pro tvorbu třídy 169
- 6.5 Jazykové otázky 173
- 6.6 Nad rámec tříd: Balíčky 174

Kapitola 7

Vysoce kvalitní rutiny 177

- 7.1 Rozumné důvody pro tvorbu rutiny 180
- 7.2 Návrh na úrovni rutiny 184
- 7.3 Dobré názvy rutin 187
- 7.4 Jak dlouhá může rutina být? 189
- 7.5 Jak používat parametry rutin 190
- 7.6 Zvláštní úvahy na téma užití funkcí 195
- 7.7 Makra a vložené rutiny 197

Kapitola 8

Defenzivní programování 201

- 8.1 Ochrana programu před zadáním neplatných informací 202
- 8.2 Aserce 204
- 8.3 Techniky ošetřování chyb 208
- 8.4 Výjimky 212
- 8.5 Jak zabezpečit program, aby zvládl škody napáchané chybami 217
- 8.6 Podpora ladění 218
- 8.7 Jak určit, jaký podíl defenzivního programování nechat v ostrém kódu 222
- 8.8 Hlavně to s defenzivním programováním nepřehánět 223

Kapitola 9

Proces programování v pseudokódu 227

- 9.1 Shrnutí jednotlivých kroků tvorby tříd a rutin 228
- 9.2 Co hovoří pro pseudokód 229
- 9.3 Stavba rutin pomocí procesu programování v pseudokódu (PPP) 232
- 9.4 Alternativy k PPP 245

Část 3: Proměnné

Kapitola 10

Obecně o užití proměnných 249

- 10.1 Datová gramotnost 250
- 10.2 Snadná deklarace proměnných 252

10.3 Jak inicializovat proměnné	253
10.4 Rozsah	257
10.5 Persistence	263
10.6 Kdy vzniká vazba?	264
10.7 Vztah mezi datovými typy a řídicími strukturami	265
10.8 Každá proměnná má mít jeden účel	267

Kapitola 11

Význam názvů proměnných **271**

11.1 Obecné úvahy na téma volby dobrých názvů	272
11.2 Pojmenování jednotlivých datových typů	277
11.3 Význam konvencí pojmenování	282
11.4 Neformální konvence pojmenování	284
11.5 Standardizované předpony	291
11.6 Tvorba krátkých, srozumitelných názvů	293
11.7 Jaké názvy nepoužívat	296

Kapitola 12

Základní datové typy **301**

12.1 Čísla obecně	302
12.2 Celá čísla	304
12.3 Reálná čísla	305
12.4 Znaky a textové řetězce	308
12.5 Booleovské proměnné	312
12.6 Výčtové typy	313
12.7 Pojmenované konstanty	318
12.8 Pole	320
12.9 Tvorba vlastních typů (přezdívaní)	322

Kapitola 13

Neobvyklé datové typy **329**

13.1 Struktury	330
13.2 Ukazatele	333
13.3 Globální data	345

Část 4: Příkazy

Kapitola 14

Sekvenční uspořádání kódu **357**

14.1 Příkazy, které musí být vykonány v určitém pořadí	358
14.2 Příkazy, které mohou být vykonány v libovolném pořadí	361

Kapitola 15	
Práce s podmínkovými příkazy	365
15.1 Příkazy if	366
15.2 Přepínače	372
Kapitola 16	
Řídicí cykly	379
16.1 Volba správného typu cyklu	380
16.2 Řízení cyklu	385
16.3 Snadná tvorba cyklů – vezmeme to z opačného konce	397
16.4 Souvislost mezi cykly a poli	399
Kapitola 17	
Neobvyklé řídicí struktury	403
17.1 Více návratových cest z rutiny	404
17.2 Rekurze	406
17.3 Příkaz goto	410
17.4 V kontextu neobvyklých řídicích struktur	422
Kapitola 18	
Metody řízené tabulkami	425
18.1 Obecné úvahy na téma užití metod řízených tabulkami	426
18.2 Tabulky s přímým přístupem	427
18.3 Tabulky s indexovaným přístupem	438
18.4 Tabulky se schodovým přístupem	440
18.5 Další příklady prohledávání tabulek	443
Kapitola 19	
Obecná témata spojená s řízením	445
19.1 Booleovské výrazy	446
19.2 Složené výrazy (bloky)	457
19.3 Prázdné příkazy	458
19.4 Jak zabránit příliš hlubokému vnoření	459
19.5 Základy programování: Strukturované programování	468
19.6 Řídicí struktury a složitost	471
Část 5: Vylepšení kódu	
Kapitola 20	
Kvalita softwaru	477
20.1 Atributy kvality softwaru	478
20.2 Techniky zlepšení kvality softwaru	480
20.3 Relativní efektivnost kvalitních technik	484

20.4 Kdy zajišťovat kvalitu	487
20.5 Obecné zásady kvality softwaru	487
Kapitola 21	
Stavba ve spolupráci	491
21.1 Přehled metod vývoje ve spolupráci	492
21.2 Párové programování	495
21.3 Formální inspekce	497
21.4 Jiné metody společného vývoje	503
Kapitola 22	
Vývojářské testování	509
22.1 Vliv vývojářského testování na kvalitu softwaru	511
22.2 Doporučené postupy pro vývojářské testování	513
22.3 Tipy pro zlepšení testování	515
22.4 Typické chyby	526
22.5 Nástroje pro podporu testování	531
22.6 Jak zlepšit testování	536
22.7 Ukládejte výsledky testování	537
Kapitola 23	
Ladění	543
23.1 Obecný pohled na ladění	544
23.2 Hledáme chyby	548
23.3 Oprava chyby	558
23.4 Psychologické úvahy o ladění	561
23.5 Ladicí nástroje – samozřejmé i ty méně samozřejmé	564
Kapitola 24	
Restrukturalizace kódu (refaktorování)	571
24.1 Kudy se ubírá evoluce softwaru	573
24.2 Úvod do refaktorování	573
24.3 Charakteristické restrukturalizace	579
24.4 Jak bezpečně restrukturalizovat	586
24.5 Strategie restrukturalizace	588
Kapitola 25	
Strategie ladění výkonu	593
25.1 Výkon obecně	594
25.2 Úvod do problematiky ladění výkonu	598
25.3 Různé formy oběžných a těžkopádných programů	603
25.4 Měření	609
25.5 Iterace	610

25.6 Shrnutí přístupů k ladění výkonu	611
---------------------------------------	-----

Kapitola 26

Techniky ladění výkonu **615**

26.1 Logika	617
26.2 Cykly	622
26.3 Transformace dat	630
26.4 Výrazy	635
26.5 Rutiny	644
26.6 Přepis v jazyku nižší úrovně	644
26.7 Čím více usilujete o změnu, tím méně se vám to daří	648

Část 6: Systémové úvahy

Kapitola 27

Jak velikost programu ovlivňuje jeho stavbu **653**

27.1 Komunikace a velikost	654
27.2 Spektrum velikostí projektu	655
27.3 Vliv velikosti projektu na chyby	656
27.4 Vliv velikosti projektu na produktivitu	657
27.5 Vliv velikosti projektu na vývojové aktivity	658

Kapitola 28

Řízení stavby **665**

28.1 Podpora dobrých programátorských postupů	667
28.2 Systém řízení konfigurace	669
28.3 Odhad harmonogramu stavby	675
28.4 Měření	681
28.5 Programátoři jsou také lidé	684
28.6 Jak řídit svého vedoucího	689

Kapitola 29

Integrace **693**

29.1 Význam způsobu integrace	694
29.2 Integrovat fázově, nebo přírůstkově?	695
29.3 Strategie přírůstkové integrace	698
29.4 Každodenní překlad a odzkoušení	705

Kapitola 30

Programovací nástroje **713**

30.1 Návrhové nástroje	714
30.2 Nástroje pro práci se zdrojovým kódem	715
30.3 Nástroje pro práci se spustitelným kódem	720

30.4 Na prostředí orientované na nástroje	724
30.5 Tvorba vlastních programovacích nástrojů	725
30.6 Pohádka plná nástrojů	726

Část 7: Softwarové mistrovství

Kapitola 31

Rozvržení a styl 731

31.1 Základy rozvržení	732
31.2 Techniky rozvržení	738
31.3 Styly rozvržení	740
31.4 Rozvržení řídicích struktur	747
31.5 Rozvržení jednotlivých příkazů	754
31.6 Rozvržení komentářů	764
31.7 Rozvržení rutin	767
31.8 Rozvržení tříd	769

Kapitola 32

Dostatečně výmluvný kód 777

32.1 Externí dokumentace	778
32.2 Programovací styl jako dokumentace	779
32.3 Komentovat, či nekomentovat?	782
32.4 Klíč k účinným komentářům	785
32.5 Techniky tvorby komentářů	791
32.6 Standardy IEEE	810

Kapitola 33

Povahové vlastnosti 815

33.1 Jakou mají povahové vlastnosti souvislost s tématem této knihy?	817
33.2 Inteligence a skromnost	817
33.3 Zvědavost	818
33.4 Duševní upřímnost	822
33.5 Komunikace a spolupráce	824
33.6 Kreativita a disciplína	824
33.7 Lenost	825
33.8 Vlastnosti s menším významem, než by se mohlo zdát	826
33.9 Návyky	828

Kapitola 34

Motivy softwarových dovedností 831

34.1 Jak zvítězit nad složitostí	832
34.2 Zvolte vhodný postup	834
34.3 Pište programy především pro lidi, teprve pak pro stroje	835

34.4 Programujte do jazyka, nikoli v něm	837
34.5 Soustřeďte svou pozornost pomocí konvencí	838
34.6 Programujte z hlediska problémové domény	839
34.7 Pozor, padající kamení	841
34.8 Opakujte znovu a znovu	843
34.9 Oddělte software od náboženství	844
Kapitola 35	
Kde najdete další informace	849
35.1 Informace o stavbě softwaru	850
35.2 Témata přesahující rámec stavby	851
35.3 Časopisy	853
35.4 Plán doporučené četby vývojáře softwaru	854
35.5 Staňte se členy profesní organizace	856
Přílohy	
Bibliografie	859
Rejstřík	877