GYMNÁZIUM VEĽKÁ OKRUŽNÁ
Veľká okružná 22, 010 01 Žilina 1

STREDOŠKOLSKÁ ODBORNÁ ČINNOSŤ

č. odboru: 02 – matematika, fyzika, informatika

Jednoduchý prehliadač súborov DVI

2010
Žilina

riešiteľ
Ľubomír Jagoš
ročník štúdia: **štvrtý**

konzultant
Ing. Michal Karas

Čestné prehlásenie

Čestne prehlasujem, že prácu na tému jednoduchý prehliadač DVI súborov som vypracoval samostatne na základe vlastných teoretických a praktických poznatkov pričom použité pramene a literatúru uvádzam.

Poďakovanie

Chcel by som sa poďakovať svojmu konzultantovi za poskytnutú pomoc pri riešení problémov, ktoré nastali počas tvorby tohto projektu.

Taktiež by som chcel vyjadriť vďaku svojim rodičom, ktorý mali dostatok trpezlivosti a pochopenia v čase keď som sa venoval vývoji tohto projektu.

Obsah

0 Úvod

Keď Donald Knuth vydal v roku 1969 prvý diel svojej knihy „The Art of Computer Programming“, bol s kvalitou sadzby, ktorá bola vytvorená ešte na strojoch Monotype, spokojný.

Pri druhom vydaní druhého dielu knihy bola už technológia Monotype úplne nahradená rýchlejšou, no menej kvalitnou fotosadzbou a sadzači do knihy zaniesli viaceré chyby. Knuth sa preto rozhodol vytvoriť program, ktorý by mu umožnil vysádzať knihu svojpomocne. Predpokladal, že prácu na programe dokončí v roku 1978, ale trvalo to o viac ako 10 rokov dlhšie.

V roku 1989 uvoľnil Knuth program TeX 3.0, v ktorom bolo použité 8-bitové kódovanie znakov, kým predošlé verzie (1.0 a 2.0) používali pre väčinu jazykov nedostatočné 7-bitové kódovanie. Kód TeXu je od verzie 3.0 zmrazený (feature freeze) a jediné povolené úpravy sú opravy chýb. Od tejto verzie sa tiež zmenil spôsob značenia jednotlivých verzí — každá novšia verzia programu TeX má o jednu cifru čísla pí viacej (3.1 → 3.14 → 3.141 atď.). Knuth si želá, aby po jeho smrti bol program úplne zmrazený a označoval sa ako TeX, version π (pí) — symbol dokonalosti. Chyby nájdené v tejto verzii budú označené ako vlastnosti programu. Aktuálna verzia TeXu je 3.141592. Knuth vytvoril aj program Metafont na dizajn písma a navrhol v ňom klasicistické písmo Computer Modern. (1)

Túto prácu som sa rozhodol vypracovať, pretože TeX je aj napriek prítomnosti mnohých iných sádzacích systémov a textových editorov v súčasnosti používaný najmä v akademickom prostredí pri písaní matematických prác. Preto som sa rozhodol vytvoriť tento jednoduchý prehliadač aby bol jeho užívateľ schopný prezerať dokumenty v DVI formáte.

1 Ciele práce

Hlavným cieľom mojej práce bolo vytvorenie prehliadača výstupných dokumentov TeX-u, tj. DVI súborov (Device Independet). Tento cieľ zahrňuje riešenie menších čiastkových problémov a to:

- čítanie bitmapových súborov PK (packed fonts) a GF (generic font)
- rekonštrukcia bitmáp znakov z informácií uložených v PK a GF súboroch
- získavanie informácií (šírka, výška …) o jednotlivých znakoch z PK a GF súborov
- čítanie pozícií jednotlivých znakov z DVI súboru
- vytvorenie strán umiestnením znakov na správne miesta
- vytvorenie stránky ako celku a možnosť jej prehliadnutia užívateľom

2 Materiál a metodika práce

Pri písaní tejto práce som taktiež vychádzal z informácií z odborných publikácií zameraných na problematiku sádzacieho systému TeX a programovacieho jazyka Java a dôležitým prameňom bol pre mňa aj internet.

2.1 Materiál

2.1.1 Java

Na implementáciu návrhu prehliadača DVI súborov som sa rozhodol použiť jazyk Java. Hlavný dôraz pri rozhodovaní bol kladený na prenositeľnosť a jednoduchosť. Jazyk Java spĺňa obe tieto požiadavky, a preto bol použitý na naprogramovanie tohto projektu.

Java je multiplatformný objektový programovací jazyk kompilovaný do bajtkódu, ktorý je interpretovaný virtuálnym strojom, skladajúcim sa z častí zajisťujúcich väzbu na hardware a z časti interpretujúcej bajtkód.

Java je v súčasnosti otvorený programovací jazyk, tj. jeho zdrojové kódy sú voľne dostupné pre každého.

Pre spustenie DVI prehliadača je nutné mať nainštalovanú virtuálnu mašinu.

2.1.2 Editor

Textový editor som použil na písanie zdrojových kódov, ktoré som následne kompiloval pomocou java compileru.

2.1.3 GF súbory

GF je skratka pre „generic font“ čo v preklade znamená „všeobecné písmo“. Je to výstup produkovaný METAFONT-om, programovacieho jazyka na tvorbu písma. Slovo „všeobecný“ znamená, že sa formát nepridŕža žiadnych manufaktúrnych pravidiel.

GF súbor je prúd 8-bitových znakov, ktoré sú interpretované ako séria príkazov v strojovom jazyku. Prvý byte každého príkazu je kód operácie, tento kód operácie je

nasledovaný žiadnym alebo viacerými byte-mi poskytujúcimi parametre pre tento príkaz. Parametre sami o sebe môžu byť zložené z viacerých byte-ov, avšak operácie sú vždy jednobyte-ové.

Aj keď sú informácie v GF súbore uložené relatívne (napr. každý znak obsahuje parameter ukazujúci na umiestnenie predošlého znaku v súbore), štandardný PASCAL v dobe vytvorenia tohto formátu nedisponoval schopnosťou náhodného prístupu do súborou, preto je možné pracovať s GF súborom aj tak, že bude čítaný od začiatku do konca.

Obraz písmena uloženého v GF súbore si môžeme predstaviť ako čierno-bielu obdĺžnikovú bitmapu, pri čítaní ktorej si musíme pamätať aktuálny riadok a stĺpec, v ktorom sa nachádzame. Referenčný bod, od ktorého začíname kresliť sa nachádza v ľavom hornom rohu. GF súbor potom obsahuje príkazy na vykreslenie niekoľkých pixelov v aktuálnom riadku, príkazy na preskočenie alebo vykreslenie celých riadkov, zmenu farby z čiernej na bielu a naopak atď.

2.1.4 PK súbory

PK je skratka pre pre „packed file format“ čo znamená „zbalený súborový formát“. PK formát je kompaktná reprezentácia dát obsiahnutých v GF súboroch. Obsah informácií je ten istý ako v GF súboroch, ale ich veľkosť je takmer vždy polovičnej veľkosti ako veľkosť GF súboru obsahujúceho rovnaké písmo. Taktiež sú ľahšie konvertovateľné do rastrovej reprezentácie, pretože neobsahujú príkazy ako paint, skip a new_row.

PK súbory sú taktiež prúd 8-byteových znakov. Občas sú tieto byte rozdelené do menších celkov – 4-bitových nybblov alebo na jednotlivé bity alebo sú kombinované do mnohobyteových parametrov. Keď sú byty rozdelené do na menšie kúsky, prvá časť byte-u je najdôležitejšia. Ak sú parametre v PK súbore negatívne čísla, bývajú uložené v doplnkovom kóde.

PK súbor bol navrhnutý tak, aby bol ľahko čitateľný a interpretovaný strojmi. Dĺžka každého znakového packetu je uložená spolu s ním, preto zariadenie/program čítajúci PK súbor, môže rýchlo a jednoducho preskakovať jednotlivé písmená.

V GF súboroch sú pri každom písmene uložené aj jeho šírka dĺžka, pri vykresľovaní písmena potom len pohybujem „perom“ v tejto mriežke. V PK súbore je pri každom znaku

uložená len jeho šírka.

GF súbor pakuje jednotlivé bity písmena počítaním počtu po sebe idúcich bielych a čiernych pixelov v riadku a zakóduje toto číslo. Namiesto počítania počtu pixelov v jednotlivých riadkoch však PK súbor spojí všetky riadky do jedného dlhého riadku a počíta biele a čierne po sebe idúce pixely v tomto dlhom riadku, pričom tieto čísla potom ukladá v spakovanej forme.

Keďže veľkosť PK súboru je menšia v porovnaní s GF súborom používa sa v dnešnej dobe v TeX-ových distribúciách namiesto GF súborov len PK súbory.

2.1.5 DVI súbor

V DVI súbore sú uložené informácie o rozložení znakov na stránke, teda vytvára nám hotovú sadzbu nášho textu. Výstupný DVI súbor čítajú ovládače jednotlivých zariadení: ovládač obrazovky = prehliadač dvi a ovládač tlačiarne = tlačový program dvi. Tieto programy teda zviditeľňujú prácu TeX-u v podobe hotovej sadzby.

Binárny formát DVI bol navrhnutý tak, aby boli splnené nasledujúce požiadavky:

- dĺžka súboru by mala byť v rámci možností čo najmenšia
- súbor by mali byť schopné ľahko a rýchlo čítať ovládače výstupných zariadení
- TeX vytvára DVI sekvenčne, tj. nevracia sa k raz zapísanej informácií
- ovládače môžu čítať DVI sekvenčne, alebo vyhľadávať požadované strany

Jednotlivé informácie sú uložené v DVI vo forme povel, parametre, povel, parametre, atď. Povel má vždy dĺžku 1B. Máme teda možnosť nadefinovať 256 povelov.

TeX pracuje pri sadzbe s jednotkou sp = 2^-16 pt. Rozmerové údaje v DVI formáte sú uložené v tvare: celé číslo x použitá jednotka . Pokiaľ je celé číslo záporné je uložené ako dvojkový doplnok.

2.2 Postup práce

Pri programovaní tohto prehliadača som sa rozhodol rozdeliť riešenie na 2 základné časti a to:

- jadro prehliadača zaoberajúce sa čítaním písmových súborov, ich prevodom do rastrovej podoby, čítaním DVI súboru a následnou tvorbou rastrovej reprezentácie stránky
- externú nadstavbu nad vykresľovacím jadrom starajúcu sa o výslednú prezentáciu jednotlivých strán DVI súboru užívateľovi

2.2.1 UML diagriam

Viď obrázok 1 v prílohe.
Postupne rozoberiem jednotlivé časti v poradí v akom boli programované.

2.2.2 Organizácia tried do balíkov

Triedy sú organizované do nasledovnej štruktúry balíkov:

DviViewer

IO

DviRandomAccessFile.class

Image

BaseImg.class

DviPageImg.class

GfLetterImg.class

PkLetterImg.class

Info

ChrInfo.class

FontManager.class

PageInfo.class

GfChrInfo.class

PkChrInfo.class

Reader

ChrReader.class

DviReader.class

GfReader.class

PkReader.class

2.2.3 DviRandomAccessFile

Aby som mohol vytvoriť z DVI súboru stránku, musím čitať samotný DVI súbor, ale okrem neho aj písmové súbory. Na uľahčenie tohto čítania som vytvoril túto triedu, ktorá je adaptérom javovskú triedu RandomAccessFile. Poskytuje možnosti náhodného čítania súborov triedy RandomAccessFile, ale ich aj ďalej rozširuje. Umožňuje čítať súbor spätným smerom, tj, od konca po začiatok a taktiež umožňuje čítať súbor po nybbloch (4-bitové časti). Tieto rozširujúce vlastnosti sú využívané najmä pri tvorbe bitmáp z PK a GF súborov.

2.2.4 Triedy vykresľujúce písmená

Tu patria triedy: *GfLetterImg* a *PkLetterImg*

Tieto triedy uchovávajú bitmapy jednotlivých písiem a obsahujú metódy na ich vykresľovanie. Sú využívané čítačmi písmových súborov. Vytvoril som 2 triedy pre vykresľovanie písiem pre oba formáty, GF aj PK. Obe triedy implementujú rozhranie BaseImg cez ktoré k nim pristupuje DVI čítač pri ich vysádzaní na konkrétne miesto na stránke. Implementáciou tohto rozhrania som teda dosiahol nezávislosť na použití písma v triede DVI čítača.

2.2.5 Triedy uchovávajúce informácie o jednotlivých znakoch

Patria sem triedy: *GfChrInfo* a *PkChrInfo*

Tieto triedy uchovávajú informácie potrebné pre vytvorenie bitmapy jednotlivých písiem, ale taktiež informácie potrebné pre ich vysadenie na správne miesto v dokumente. Pretože k týmto informáciam potrebuje pristupovať aj čítač DVI súboru, dedia obe tieto

triedy spoločné vlastnosti od triedy BaseImg, čím je DVI čítač od nich odtienený.

2.2.6 Triedy čítajúce písmové súbory

Sem patria triedy: *PkReader* a *GfReader*

Tieto triedy sa starajú o čítanie písmových súborov vo formáte GF a PK. Obe implementujú spoločné rozhranie BaseImg, cez ktoré k nim pristupuje čítač DVI súboru. Implementáciou rozhrania BaseImg som dosiahol odtienenie tried čítajúcich písmo od čítača DVI súboru. Takto môžu byť pri tvorbe DVI súboru použité súbory v oboch formátoch a DVI čítač sa tak nestal závislý len na jednom druhe písma.

Princíp fungovania týchto tried je identický. Každá trieda si pri vytvorení načíta údaje o polohe v súbore, šírke a rôznych ďalších parametroch jednotlivých znakov písma uložených v danom súbore do triedy na uschovanie týchto informácií. Pri požiadaní DVI čítača o konkrétny znak potom za pomoci týchto údajov skočia v súbore na miesto, kde sa nachádza bitmapa písma, vytvoria ju za pomoci tried vykresľujúcich písmená a odovzdajú DVI čítaču triedu starajúcu sa o vykresľovanie písmien, pretože tá uchováva bitmapu písmena a ďalšie funkcie na manipuláciu s ním.

2.2.7 Trieda PageInfo

Táto trieda slúži na uchovanie čísla strany a jej polohy v DVI súbore.

2.2.8 FontManager

Táto trieda slúži na vyhľadávanie písmových súborov na disku. Obsahuje zoznam adresárov (možno do neho pridávať alebo z neho odoberať adresáre), v ktorých po požiadaní o daný súbor rekurzívne vyhľadáva do hĺbky 8. Rekurzívne hľadanie som obmedzil z dôvodu, chýb užívateľov pri zadávaní adresárov.

2.2.9 Rozhrania a triedy pre spoločny prístup k čítačom písmových súborov, triedam uchovávajúcim informácie o jednotlivých znakoch a triedam vykresľujúcim písmená

Do tejto kategórie sa zaraďujú: rozhranie ChrInfo a ChrReader a abstraktná trieda BaseImg

Tieto triedy slúži pre jednotný prístup k nástrojom na tvorbu písma a informáciam o jednotlivých znakoch alebo ich obrázkoch. Tieto rozhrania a triedy boli vytvorené za účelom odtieniť DVI čítač od tried starajúcich sa o tovrbu písma a docieliť tak možnosť použitia GF alebo PK písmových súborov.

2.2.10 Čítač DVI súborov

Sem patrí trieda DviReader

Táto trieda sa stará o čítanie DVI súboru a vytváranie bitmapy stránky. Spája dokopy všetky ostatné triedy, ktoré plnia čiastkové úlohy ako tvorbu písma, čítanie písmových súborov, vyhľadávanie písmových súborov na disku atď. a s ich pomocou vytvára výslednú stránku.

2.3 Vytváranie stránky z DVI súboru

Vytváranie stránky z DVI súboru je pomerne komplexná záležitosť. Pozostáva z viacerých menších procesov:

- načítanie údajov o dokumente (počet strán, použité písma, umiestnenie jednotlivých strán v DVI dokumente atď)
- nájdenie jednotlivých súborov s písmom na disku
- čítanie súborov s písmom a vytváranie konkrétnych znakov
- umiestňovanie znakov na stránku na správne miesto
- odovzdanie hotovej stránky ďalej

2.3.1 Načítanie údajov o dokumente

Načítanie údajov o dokumente prebieha rovnako ako pri čítaní písmových súborov. DVI čítač najprv nájde v DVI súbore oblasť definícií použitých písiem, počet strán a polohy jednotlivých stránok v súbore. Polohy jednotlivých stránok v súbore si uschováva do tried PageInfo, na vyhľadávanie jednotlivých písmových súborov na disku mu slúži trieda FontManager.

2.3.2 Nájdenie písmových súborov na disku

O túto časť sa stará trieda FontManager. Možno jej pridávať a odoberať adresáre, v ktorých sa má vyhľadávať. Vyhľadáva sa rekurzívne do hĺbky 8 podadresárov. DVI čítač vždy vkladá do FontManager-a medzi adresáre v ktorých sa má hľadať aktuálny adresár.

2.3.3 Čítanie súborov s písmom a vytváranie konkrétnych znakov

DVI čítač pracuje s písmovými čítačmi cez rozhranie ChrReader. Pri načítaní príkazu na vysadenie konkrétneho písmena, pošle požiadavku na čítač písmového súboru a ten mu odovzdá triedu BaseImg. Tento proces sa opakuje kým sa nedosiahne posledný znak na stránke.

2.3.4 Umiestňovanie znakov na stránku na správne miesto

DVI čítač po prijatí triedy BaseImg, prispôsobí bitmapu znaku a vysadí ho na správne miesto odovzdaním bitmapy znaku triede DviPageImg s údajmi o jeho polohe.

2.3.5 Odovzdanie hotovej stránky ďalej

Výsledkom vytvárania stránky je trieda DviPageImg uchovávajúca bitmapu stránky. Túto triedu odovzdá DviReader na ďašie spracovanie žiadateľovi, ktorý ju môže podľa ľubovôle spracovať. K triede DviPageImg môže potom žiadateľ pristupovať za pomoci triedy BaseImg od ktorej je táto trieda zdedená.

2.3.6 Grafická nadstavba nad vykresľovacím jadrom

Túto nadstavbu som vytvoril ako nezávislú časť nad vykresľovacím jadrom. Využíva jadro na vykreslenie konkrétnej stránky, ktorú zobrazuje v okne. Taktiež umožňuje jednoduché približovanie a oddaľovanie stránky, presúvanie sa na nasledujúce stránky dokumentu, pridávanie adresárov v ktorých sa vyhľadávajú písmové súbory a taktiež zjednodušuje voľbu zobrazovaného súboru.

3 Výsledky a diskusia

Výsledkom mojej práce je jadro DVI prehliadača fungujúceho na jednoduchej tvorbe rastrovej reprezentácie stránky a grafické rozhranie k tomuto jadru, čím som umožnil jednoduchšiu interakciu s užívateľom.

Jadro a grafické rozhranie sú na sebe nezávislé, teda je možné naprogamovať aj iné výstupné triedy využívajúce vykresľovacie jadro. Jadro by bolo možné použiť napríklad v appletoch na vykresľovanie DVI súborov na internete alebo ho zabudovať do iných aplikácií.

Taktiež je možné využitie jednotlivých tried riešiacich vykresľovanie písiem samostatne a použiť ich na tvorbu obrázku písma v danom písmovom súbore. Môžeme teda použiť túto triedu na vytvorenie katalógu písma bez jedinej zmeny tried zaoberajúcich sa tvorbou písma. Táto modulárna štruktúra jadra umožňuje vylepšovanie jednotlivých častí DVI prehliadača bez nutnosti zásahu do ostatných častí a taktiež umožňuje využitie tried aj v iných projektoch.

4 Závery práce

Snažil som sa vytvoriť funkčný prehliadač DVI dokumentov, ktorý by bol jednoduchý ako po technickej, tak aj po užívateľskej stránke. DVI formát sa v súčasnosti už takmer nepoužíva, nahradili ho iné omnoho schopnejšie formáty majúce lepšie vlastnosti. No napriek tomu by sme nemali tento formát prehliadať. Môže nám poslúžiť ako ukážka ako sa v minulosti vyvíjala počítačová typografia, ukážka organizácie dát v binárnom formáte, rôzne komprimovacie techniky a taktiež nám môže poslúžiť ako inšpirácia. DVI formát a formáty písma GF a PK boli v nedávnej minulosti jedny z mála možností na tvorbu dokumentov a ako inšpirácia nám môžu dobre poslúžiť aj dnes. Veď aj v súčasnej dobe sa požívajú binárne formáty písma ako TrueType Font, Type 1, Type 2 atď.

Chcel som vytvoriť tento prehliadač ako pomôcku na prehliadanie DVI dokumentov, ale aj ako inšpiráciu pre ďalších ľudí, ktorí sa pokúšajú vytvoriť čítače dokumentov. Pri jeho tovrbe som sa snažil používať princípy objektového programovania pri vytváraní jednotlivých tried a ich vzájomných previazaní.

5 Resumé

The goal of my effort was the creation of the simply viewer of the DVI files. I choose for this purpose the progamme language Java which I consider the best for this kind of application because of his performance, multiplatform use a good documentation. I also use the OOP principles while I created the skeleton of this application. I also considered about the future extensability and so take the encapsulation of the objects to the mind.

The application which was created behaves regulerly, but there are also the areas where the functionality could be better. I fullfill the minimal requirements which I determined at the beginning.

The main parts of the application core can be reuse in the other projects or they can be use to fulfill the partial purpose for which were created.

I can tell, that I'm satisfied with this application.

6 Zoznam použitej literatúry

PETR OLŠÁK: TexBook naruby, Konvoj 1997

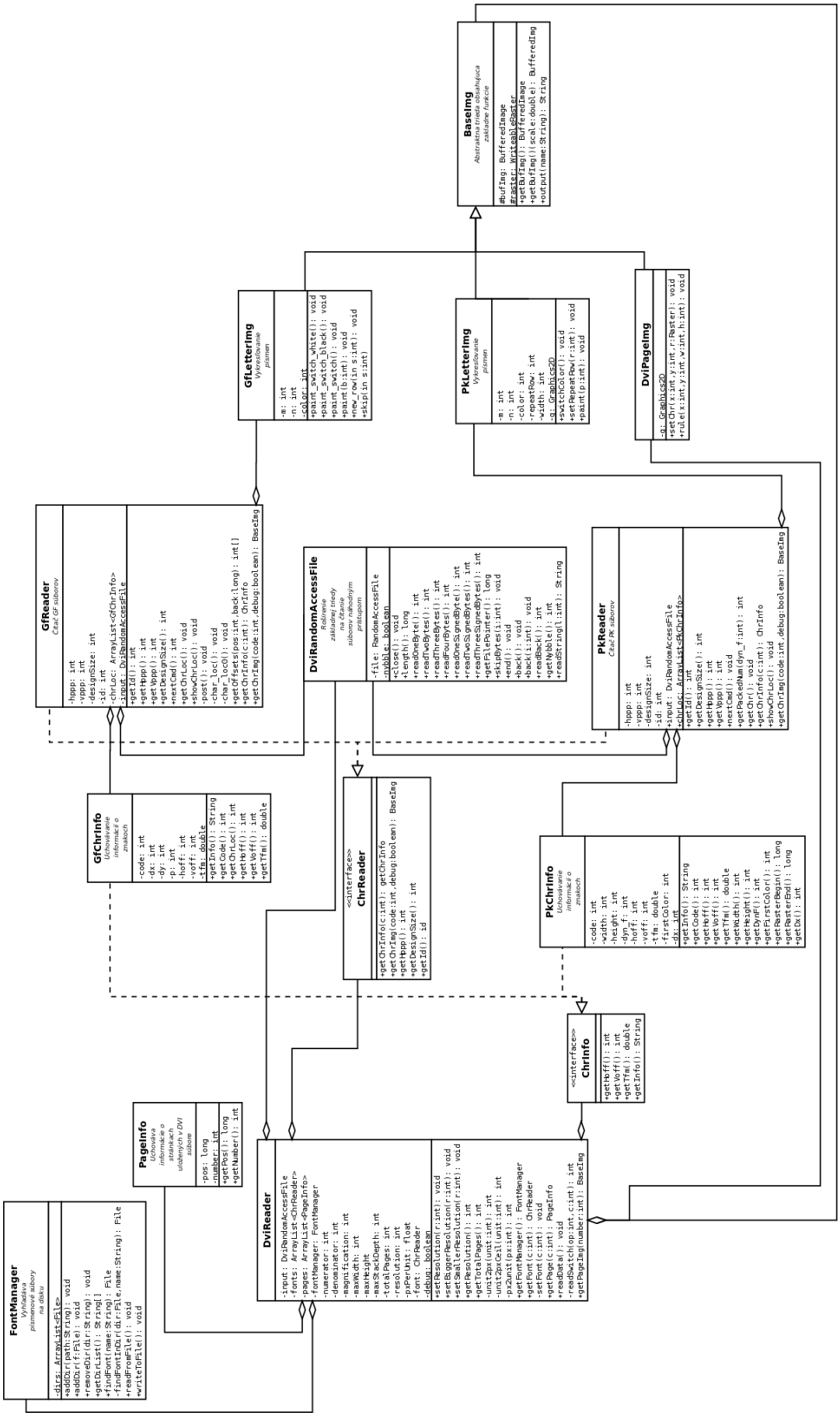
PAVEL HEROUT: Učebnice jazyka Java, České Budějovice 2001

Documentation of gftopk programme: Parts 14 – 36
http://ftp.cvut.cz/tex-archive/systems/knuth/dist/mfware/gftopk.web (1990-07-29)

Java™ Platform, Standard Edition 6, API Specification
http://java.sun.com/javase/6/docs/api/

1. http://sk.wikipedia.org/wiki/TeX

7 Prílohy



Obrázok 1: UML diagram vykresľovacieho jadra