



Przetłumaczono na język: [niderlandzki](#)

Pokaż oryginał

Opcje ▼

niderlandzki ▼

Technologia  Tłumacz

Universiteit van Silezië



Afdeling Technologie

*Instituut voor Toegepaste Informatica*

# **MASTERPROEF**

Onderwerp:

Printer- en plotteremulatie in  
grafische modus

Uitgevoerd door: Begeleider:

Ryszard Czekaj, prof. doktor hab. in . Dariusz Badura

Paweł Szymik Adviseur:

mgr Ireneusz Gocinak

## PIJNBOOM 1996

## Inleiding

Naast het afdrukken van eenvoudige teksten kunt u met moderne printers ook gebruikmaken van diverse schaalbare lettertypen en zelfs grafische afbeeldingen op de pagina plaatsen. Plotters worden, net als printers, aangestuurd door speciale sequenties, die in veel gevallen kunnen worden behandeld als afdrukbesturingstalen. De stroom besturingscodes kan samen met de af te drukken gegevens naar een printer worden gestuurd of in een bestand op de schijf van de computer worden opgeslagen. Een dergelijk bestand met gegevens voor een printer of plotter wordt een printbestand genoemd. Dit bestand kan later naar een apparaat worden verzonden dat is aangesloten op een ander computerstation. Het afdrukbestand bevat besturingssequenties die specifiek zijn voor een specifiek type apparaat en die de sequenties later correct zullen interpreteren.

Het doel van dit werk is om een computerprogramma te ontwikkelen dat de werking van een geselecteerde printer en plotter nabootst, waardoor het mogelijk wordt om het afdrukbestand grafisch weer te geven op een computerscherm.

Om dit doel te bereiken, zijn de volgende aannames gedaan:

- er is een afdrukbestand van geselecteerde printers en plotters in het voor hen geschikte formaat
- het printer- of plottermodel waarvoor het afdrukbestand is gemaakt, is bekend
- het afdrukbestand kan eenvoudige tekstgegevens en een grafische afbeelding bevatten
  - het ontwikkelde programma zal de afbeelding in het bestand zo getrouw mogelijk weergeven
- het programma zal het afdrukken van gegevens op een andere printer mogelijk maken dan de printer waarvoor het afdrukbestand is voorbereid
- Het wordt aanbevolen om de grafische omgeving Windows 3.xx en het Borland C++-programmeerhulpprogrammapakket te gebruiken om het programma te bouwen .



# Technische aspecten van de werking van printers en plotters

## 1. Classificatie van printers

### 1.1. Dot matrixprinters (impactprinters)

In een matrixprinter wordt de afbeelding gecreëerd door een printkop die is uitgerust met een reeks pinnen (meestal 9, 24 of 48) die tegen een met inkt doordrenkt lint slaan terwijl de printkop heen en weer voor het papier wordt bewogen. De naalden zijn in één of meerdere verticale rijen gerangschikt.

Inna, mniej znana technologia, nazywana Comb Matrix lub Shuttle Matrix posuguje si poziomym rzdem igie (optymalnie jedna igła dla każdego znaku poziomego), ponieważ pojedynczy rzęd punktów potrafi lepiej odwzorować znak. W przypadkach, gdy nie ma tylu igie, ile jest znaków w poziomych, grzebie porusza się nieznacznie do przodu i do tyłu, aby wydrukować niezbędne punkty. Przy tej technice papier może przesuwany być prawie bezustannie i dlatego jest ona częściej używana w szybkich drukarkach mozaikowych. Technologia ta jest znacznie sprawniejsza, niż drukowanie pojedynczych symboli.

Uderzeniowe drukarki mozaikowe są wszechstronne, tzn. matryca z igłami może być ustawiana do drukowania czcionek, a nawet grafiki.

Drukarki mozaikowe mogą być używane do drukowania kopii razem z oryginałami, ponieważ obrazy tworzone są poprzez uderzanie w taśmę barwniczą. W drukarkach tych stosuje się papier cichy, chociaż w większych modelach może być użyte pojedynczych kartek oraz etykiet.

Ze względu na metodę drukowania, drukarki mozaikowe są znacznie głośniejsze niż inne modele (nawet w tzw. trybie cichej pracy).

Prędkość drukowania, podawana zwykle w cps (znaki na sekundę), jest różna, w zależności od używanych czcionek (Draft/LQ) oraz różnych wartości rozdzielczości cpi (punkty na cal) [16], [15], [8].

### 1.2. Drukarki Daisywheel i maszyny do pisania

W drukarkach typu Daisywheel (drukarki rozetkowe) wydruk tworzony jest za pomocą

bona, na którą rym wytoczono wszystkie litery i symbole. Głowica przesuwa się po papierze w przód i w tył, a elektromagnes zaczyna pracować, gdy bon jest ustawiony w pozycji potrzebnej litery.

Drukarki Daisywheel popularne były ze względu na jakość wydruku lepszą, niż we wcześniejszych modelach, jak np. w dziewięciu owych drukarkach mozaikowych.

Głównie ich wady to: mała szybkość drukowania, ograniczona ilość symboli na bonie, oraz brak możliwości tworzenia grafiki.

Drukarki Daisywheel należą do typu drukarek uderzeniowych. Umoliwia to drukowanie kopii jednocześnie z oryginałem. Drukowanie jest możliwe na papierze ciągłym, jak również na pojedynczych arkuszach oraz etykietach.

Prędkość drukowania podaje się zwykle w cps (znaki na sekundę) [8].

### 1.3. Drukarki atramentowe

Drukarki atramentowe działają podobnie jak mozaikowe, które tworzą wydruk przez układanie punktów na papierze. W drukarkach atramentowych krople atramentu rozpryskiwane są bezpośrednio na papier. Znacznie głośniejsze, niż w drukarkach mozaikowych, wzrzą punktów pozwala uzyskać wydruk lepszej jakości.

W nowszych modelach atramentowych atrament nie jest wyrzucany elektrostatycznie lub poprzez drobne pompki, lecz zamieniany w parę przez podgrzewanie i, pod ciśnieniem rozszerzającej się pary, наносzony jest na kartkę.

Modele atramentowe nie należą do typu drukarek uderzeniowych, więc kopie drukuje się oddzielnie. Drukowanie jest możliwe na papierze ciągłym, jak w przypadku fotokopiarki. Niektóre modele, dla lepszego wydruku wymagają specjalnej jakości papieru, chociaż dla większości drukarek wystarczy papier standardowy.

W niektórych drukarkach można używać papieru ciągłego i można na niego drukować etykiety, pod warunkiem, że one wykonane z papieru dobrze wchłaniającego atrament.



rys. A HP DeskJet 600 C

dem zamaczania. Może to mieć znaczenie, jeżeli wydruk ma być użyty do prezentacji w środowisku o dużej wilgotności.

Drukarki atramentowe są cichsze i znacznie szybsze od drukarek uderzeniowych. Parametr szybkości wydruku podaje się w cps (znaki na sekundę), czasami także, dla różnych częstotliwości w ppm (strony na minutę).

Pomimo, że atramenty tych drukarek są rozpuszczalne w wodzie, różni się bardzo pod względem

Niektóre rodzaje atramentu nie są odporne na działanie wody. Do takich atramentów zalicza się: wszystkie Canon-y, czarny DEC 520ic oraz atramenty TI Micromarc. Natomiast czarny atrament HP DeskJet (rys.1) jest prawie tak wodoodporny jak wydruk laserowy. Najbardziej wodoodporne wydają się być czarne i kolorowe atramenty Epsona: litery rozpraszają się, ale nie rozmazują pod wpływem wilgoci.

Wydruki kolorowe (atramentowe) niestety nie są tak długotrwałe jak fotografie. Te z nich, które się wytrzymałe na wiatro, czysto niszczy wilgoć, podczas kiedy inne szybko powoją w jasnym świetle. Trwałość wydruku może być przedłużona przez laminowanie. Niestety, nowoczesne kolorowe drukarki atramentowe nie są w stanie dostatecznie długo, by zbadać długowieczność najnowszych atramentów [8], [10], [11].

## 1.4. Drukarki laserowe i drukarki LED

Modele laserowe i LED nie należą do typu drukarek uderzeniowych, tak więc kopie muszą być drukowane oddzielnie. Używane są w nich zwykle pojedyncze arkusze papieru. Można w nich drukować etykiety wytrzymałe na podgrzewanie podczas wypalania tonera.

Aby powstał wydruk w drukarce laserowej, specjalny bęben zostaje naładowany elektrostatycznie, a rysunek strony powstaje przez rozładowywanie bębna promieniem lasera lub linii LED (ang. Light Emitting Diode) w odpowiednich punktach. Do rozładowanych punktów przylega toner. Punkty te są lustrzanym odbiciem miejsc, w których nośnik ma się ukazać na wydruku końcowym.



rys.B HP LaserJet 4V

Papier mijając drut Corona, który nadaje mu ładunek elektryczny przeciwny do ładunku punktu w bębnie, następnie przechodzi przez wałek, z którego toner jest zatrzymywany na jego powierzchni. Ostatecznie, przesuwany jest pod elementem ciepłym, który stapiając toner, łączy go z papierem.

Wśród obecnych drukarek laserowych nie posiada drutu Corona, a papier elektryzowany jest za pomocą kłosa. Drukarki laserowe i LED tworzą bardzo wyraźny, precyzyjny wydruk, o rozdzielczości powyżej 300 dpi (osiąga nawet 1200 dpi). W niektórych drukarkach zastosowano różne techniki, jak np. technologia rozszerzonej rozdzielczości (ang. Resolution Enhancement) w celu kontroli rozmiarów punktu dla wygładzenia konturów.

Drukarki laserowe i LED są stosunkowo ciche. Wydruk modeli laserowych i LED jest mierzony w ppm (strony na minutę) w pozycji copy mode (tzn., obraz strony, który jest już w pamięci drukarki, kopiowany jest na papier możliwie szybko) [8], [10], [11].

## 1.5. Drukarki kolorowe

W drukarkach kolorowych zastosowano kilka metod służących do tworzenia kolorowego wydruku.

Niezależnie od metody używanej do produkcji kolorowego wydruku, istnieją trzy różne sposoby wyznaczania kolorów:

RGB (ang. red-green-blue): metoda standardowa, używana we współpracy z urządzeniami takimi, jak monitory i telewizory. Jest to dodawczy proces mieszania kolorów (kolor biały uzyskuje się przez wymieszanie wszystkich trzech kolorów z tą samą intensywnością).

CMYK (ang. cyan-magenta-yellow-black): metoda używana szczególnie przy drukowaniu dla celów handlowych. Jest to odejmujący proces mieszania kolorów (kolor czarny uzyskuje się przez wymieszanie wszystkich trzech kolorów z tą samą intensywnością lub przez posiadanie dodatkowego koloru czarnego, daje go bogatszy, ciemniejszy czerny).

Rozrzucanie (ang. dithering): metoda, w której kolory nie są naprawdę mieszane, lecz przez umieszczenie punktów różnych kolorów w różnych ciętych miejscach uzyskuje się obraz sprawiający wrażenie ujęcia w czterech kolorach. Metoda ta jest podobna do metody otrzymywania szarości przez ustawienie więcej lub mniej czarnych punktów.

W kolorowych drukarkach mozaikowych stosuje się kolorową taśmę. W drukarkach atramentowych znajdują się pojemniki z kolorowym atramentem. Niektóre z nich tworzą kolor czarny poprzez mieszanie kolorów (HP DeskJet 600 C rys.1), w innych znajduje się oddzielny pojemnik z czarnym atramentem (HP DeskJet 660 C).

W drukarkach termicznych stosuje się film z czterema kolorami wosku. Drukarka przesuwając papier cztery razy, jeden raz dla każdego koloru. Ponieważ każdy kolor jest przenoszony na papier, powstaje pełny kolorowy obraz.

Niektórzy producenci wprowadzili ostatnio na rynek kolorowe drukarki laserowe, posiadające cztery oddzielne pojemniki na toner, jeden na każdy kolor podstawowy.

W kolorowych drukarkach laserowych stosuje się zwykle cztery kolory tonera: trzy główne odejmujące (cyan, magenta i yellow) oraz czarny.

Widzialne spektrum jest ciągłe, można więc tylko przybliżyć kolory widzialne przez wymieszanie barw podstawowych, odejmujących (mieszając razem tonery cyan, magenta i yellow uzyskuje się właściwy rodzaj szarości, a nie mocną czern). Dodanie czarnego tonera pozwala uzyskać więcej kolorów oraz znacznie wyraźniejszą czern. We wszystkich kolorowych drukarkach laserowych zakres kolorów tworzy się przez mieszanie tonerów na jeden z dwóch podstawowych sposobów:

1) W drukarkach o tonowaniu ciągłym można zmieniać ilość tonera dla każdego koloru w każdym punkcie. Drukarki te są droższe, ale umożliwiają jednocześnie reprodukcję o jakości zbli-



onej do fotografii. Dla drukarki kolorowej 32-bitowej, o tonowaniu ci g ym ka dy piksel mo e mie 4,294,967,296 r nych kombinacji tonera. Drukowanie ich wszystkich nie jest konieczne, gdy np. kombinacja mocnej czerni z C, M i Y wygl da identycznie, a u ytkownik ponosi jedynie straty tonera. Ponadto, przej cie z 24-bitowej palety RGB na 32-bitow palet CMYK nie jest zazwyczaj oczywiste, u ytkownik ma wi c w a ciwie dost p do 16,777,216 kolor w.

Nie wszystkie kolory, które mogłyby przedstawiane na ekranie w paletcie RGB powstają przez zmieszanie czterech tonów. Niektóre z nich leżą poza skalą i mogą być osiągnięte przez zastąpienie kolorem drukowalnym, postrzeganym podobnie. W ten sam sposób będzie możliwe drukowanie kolorów, których nie da się odtworzyć na ekranie RGB.

2) W drukarkach o tonowaniu stopniowym nie ma na zmianę ilości tonera przypadają tego na kolor w obrębie każdego piksela. Stosunkowo ta sfera, ale ich wydruk jest dużo niżej jakości, szczególnie przy odtwarzaniu obrazów z rzeczywistości. Każde z czterech kolorów może znajdować się, lub nie, w każdym pikselu, tak więc każdy punkt może mieć tylko 16 różnych kombinacji tonera. Ponadto, kolor czarny, zmieszany z jakimkolwiek innym, będzie wyglądał jak czarny, więc 8 z tych kombinacji będzie wyglądać tak samo, i każdy piksel może w rzeczywistości przybrać 9 różnych kolorów. Kolory, które nie mogą być reprezentowane bezpośrednio, są pozorowane przez roztrzaskanie (ang. dithering), co powoduje, że oko ludzkie odbiera kolory pośrednie w taki sam sposób, jak przy tonowej fotografii w gazetach. Drukowanie nie jest możliwe w 16,777,216 kolorach prezentowanych przez 24-bitowe RGB. Można drukować tylko 9 różnych kolorów, a więc konieczne jest użycie procesu roztrzaskania dla pokazania 24-bitowych kolorów. Roztrzaskanie pozwala na przedstawienie kolorów pośrednich, w sposób porównywalny do tonacji tego. Do drukarek o tonowaniu stopniowym można dostarczać 24-bitowe dane, co nie wpływa na możliwość mieszania różnych ilości każdego tonera dla danego zakresu kolorów każdego punktu [8].

## 1.6. Inne rodzaje drukarek

Istnieją inne technologie druku np. technologia drukarek liniowych, wykorzystywana szczególnie w systemach mainframe. Drukowanie następuje za pomocą obracającego się cylindra lub taśmy zawierającej wszystkie symbole.

W przypadku zastosowania cylindra rotacyjnego wszystkie kolumny zawierają zestaw znaków, cylinder obraca się w kierunku przesuwania się papieru, a papier jest zatrzymywany między tymi cylindrami, kiedy symbol do wydrukowania znajduje się we właściwej pozycji. Jeden obrót cylindra drukuje całą linię tekstu.

W przypadku zastosowania a cucha rotacyjnego zestaw symboli jest powt rzony na a cuchu kilka razy. Projektant drukarki ustala ilo powt rze ka dego symbolu, w zale no ci od tego jak cz sto pojawia si on w tek cie. W ameryka skim angielskim jest to sekwencja ETAON.... Je li

używane symbole powtarzane są częściej, w ten sposób drukowanie całości jest szybsze. a cucha obraca się prostopadle do kierunku przesuwania się papieru, a papier jest zatrzymywany między taśmą a cuchem, kiedy symbol do wydrukowania znajduje się we właściwej pozycji. Wariantem drukarki a cuchowej jest drukarka typu Belt Printer. Jako wydruku jest zbliżona do jakości wydruku drukarki a cuchowej, przy jednoczesnym poborze mocy i mniejszym natężeniu hałasu. W standardowej drukarce belt używa się wysokiej jakości zestawu w a cucha. W drukarkach IBM 3203 i 1403 każde uderzenie powoduje wydrukowanie trzech symboli z a cucha. Użycie a cucha wyrażone jest celowo: trzy symbole na moduł podczas obrotu a cucha następują po sobie.



rys.C IBM 3203

W drukarce belt symbole ustawione są osobno na palcach (ustawienie podobne do symboli na daisywheel) umieszczonych na gumowym pasie. Podobnie jak w drukarce a cuchowej, nie wszystkie symbole są powtarzane z tą samą częstotliwością. W odróżnieniu od drukarek cylindrowych i a cuchowych, pas jest zatrzymywany między taśmą a papierem.

Prędkość drukowania dla tych drukarek podawana jest zwykle w lps (linie na sekundę) lub nawet w pps (strony na sekundę). Drukarki belt, drukarki a cuchowe oraz drukarki z cylindrem rotacyjnym (do 70 lps) polecane są dla zwiększenia szybkości wydruku.

Istnieją również drukarki mozaikowe termiczne, umożliwiające drukowanie symboli i grafik (np. barcodes) przez podgrzewanie papieru (punkt w): miejsca podgrzane stają się czarne.

W dziedzinach związanych z tworzeniem wydruków wykorzystuje się jeszcze kilka innych egzotycznych technologii np. papier pokryty aluminium stosowany przez Sinclair'a, a także technologia rozładowania iskry (ang. spark discharge). Są one jednak bardzo rzadkie i w większości przestarzałe [8].

## 2. Klasyfikacja ploterów

### 2.1. Plotery tablicowe

Plotery tablicowe są urządzeniami o dużych gabarytach, zbliżonych do maksymalnego formatu rysunku. Na przykład, ploter formatu A1 lub A0 ma wygląd ogromnego stołu, na którym arkusze kalki mocowane elektrostatycznie. Po tablicy poruszają się pisaki na stalowych szynach, a wykreślenie najdrobniejszego nawet elementu trwa bardzo długo, ponieważ pisaki muszą się przemieszczać tam i z powrotem przebiegając po szynach [11].

## 2.2. Plotery pisakowe (b bnowe)

Znacznie doskonalsze s plotery pisakowe (rys. 4). Urz dzenia te zajmuj o wiele mniej miejsca, a arkusz kalki mocowany jest ju tylko w dwu punktach.



rys.D Ploter pisakowy

Zasad dzia ania mo na najpro ciejsz przedstawi w postaci ramy z rozpi t kalk poruszaj c si po rolkach w p aszczy nie pionowej, podczas gdy karetk z pisakami porusza si w p aszczy nie poziomej. Wymiary gabarytowe plotera pisakowego musia y by dostosowane wy cznie do szeroko ci formatu. By y to urz dzenia znacznie mniejsze, cichsze i szybsze.

Szybko pracy by a stale zwi kszana dzi ki optymalizacji programowej ruchu karetki z pisakiem.

W ploterach pisakowych nie uda o si przewyci y nast puj cych wad: mo liwo ci kolorystyczne ogranicza y si do wymiany pisak w, a wype nianie p l sprowadza o si do ich zakreskowania. Rysowanie bitmap, by o mo liwe tylko teoretycznie. Jako rysunku zale a a od szeregu czynnik w, jak np. jako tuszu i pisaka oraz szybko przesuwu pisaka. W przypadku skomplikowanych, wielowarstwowych rysunk w wi kszych format w wyst powa y niedok adno ci rz du milimetr w, spowodowane kumulowaniem si mechanicznych odchyle przesuwu pisaka.

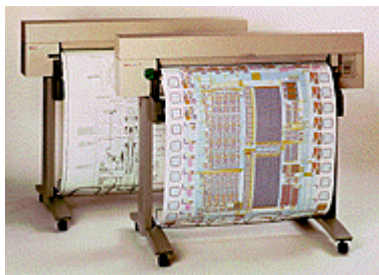
Rozw j ploter w pisakowych (laboratoria firmy Houston Instruments) doprowadzi z czasem do powstania plotera z g owic skanuj c . Zamontowanie takiej g owicy przekszta ca o ploter w wielkoformatowe urz dzenie skanuj ce [11].

## 2.3. Plotery atramentowe

R wnolegle z ploterami pisakowymi rozwija y si rastrowe plotery atramentowe. Urz dzenie to dzia a na zasadzie rozpylania na kalce kropelek tuszu, podobnie jak w drukarce atramentowej. Plotery atramentowe s znacznie dok adniejsze i atwiejsze obs uduje, za efekty ich pracy s bardziej przejrzyste.

Obecnie wiele firm wprowadza na rynek wielkoformatowe plotery rastrowe. Najnowszym produktem tego rynku jest produkt firmy Hewlett-Packard DesignJet 750C. Jest to ploter formatu A0, kolorowy, dysponuj cy niezwykle wysok rozdzielczo ci true 600 dpi w systemie monochromatycznym. Jak wi kszo nowoczesnych wyrob w Hewlett-Packarda, jest to urz dzenie ca kowicie bezobs ugowe. Po za adowaniu rolki papieru i w o eniu na swoje miejsce naboi

atramentowych ploter mo na w czy . DesignJet (rys.5) ju samodzielnie troszczy si , by wszystkie



rys.E DesignJet 350 C

dysze prawid owo dzia a y, by papier ustawiony by w a ciwie, sam sprawdza parametry no nika i dostosowuje si do podanego formatu.

Oryginalnym wk adem Hewlett-Packarda w rozw j koncepcji plotera rastrowego jest zastosowanie tzw. tuszu pigmentowego. Do tej pory g wnym argumentem przeciwko ploterom atramentowym, by o zjawisko zasychania dysz.

W HP zadbano, by produkt sam sprawdza , czy jaka dysza si nie zatka a, i sam pr bowa awari usun . Dopiero zastosowanie tuszu pigmentowego przynios o radykalne efekty i przyczyni o si do podniesienia jako ci wydruku. Tusz ten, w odr nieniu od innych stanowi zawiesin cia a sta ego. Naboju z takim tuszem nie mo na ju uzupe nia tzw. refilami.

Efektem zastosowania tuszu pigmentowego jest zwi kszenie wyrazisto ci wykre lonej linii b d powierzchni. Napylna na papier warstewka substancji zabezpiecza go przed wilgoci i podnosi trwa o oraz odporno rysunku.

Drobnodziarnista struktura tuszu umo liwia uzyskanie rozdzielczo ci rz du 600 dpi.

Innym ciekawym rozwizaniem zastosowanym w modelu 750C jest kompensacja luzu na z batkach toru karetki. Usuwa ono efekt poszarpanych linii pionowych, kt ry w przypadku niekt rych innych urz dze , zw aszcza przy rysowaniu cienkimi liniami du ych element w tworzy ma o estetyczny efekt wizualny.

DesignJet 750C jest urz dzeniem niezwykle cichym i szybkim. Cho przy prostych rysunkach z zasady nie mo e konkutowa z ploterami pisakowymi, w przypadku skomplikowanych, z czasem rysowania formatu A1 rz du 4 minut, przewy sza je.

Du ym udogodnieniem jest mo liwo wyposa enia plotera w tzw. kart HP JetDirect. Jest to karta sieciowa, dzi ki kt rej ploter widziany jest w sieci jako urz dzenie dost pne dla wszystkich pracownik w i nie wymaga ustawienia dodatkowego serwera. 8 MB RAM, w kt re standardowo wyposa ony jest DesignJet, pozwala na kre lenie bardzo du ych rysunk w (nawet o wielko ci kilkunastu megabajt w).

Sterowniki rozwizano w taki spos b, by drukowanie odbywa o si w trybie ci g ym w pamci komputera, r wnolegle z przetwarzaniem wektor w na bitmap . Dodatkowa pamci potrzebna jest dopiero w wczas, gdy rozbudowujemy system o funkcje postscriptowe [11].

### 3. J zyki sterowania drukarek i ploter w

Nowoczesne drukarki mog tworzy bardzo z o one rysunki, niekiedy konami co

Nowoczesne drukarki mogą tworzyć bardzo duże rysunki, niekiedy konkurując z fotografiami lub z dokumentami otrzymywanymi z drukarni. Sposób przesyłania danych od drukarki jest sam w sobie formatem zapisu obrazu w pliku. Istnieją dwa ogólne typy formatów plików wydruku dla drukarek: rozszerzone formaty tekstowe (ang. extended text format) i języki opisu strony (ang. page description languages).

### 3.1. Rozszerzone formaty tekstowe

Formaty takie włączają informacje graficzne do konwencjonalnego strumienia danych tekstowych. Zwykle tekst jest drukowany, a sekwencje sterujące wprowadzają elementy nietekstowe. Szeroko rozpowszechnionym formatem jest Printer Control Language (PCL). Stał się on standardem dla drukarek laserowych niskiej i średniej wydajności [6], [15], [16], [8].

#### 3.1.1. IBM ProPrinter

Język drukarki IBM ProPrinter początkowo używany do drukarek IBM (IBM Graphics Printer 5152, IBM ProPrinter XL 4201/4202, IBM ProPrinter X24/XL24 4207/4208), jest też używany obecnie przez wiele drukarek mozaikowych jak w przypadku Epsona ESC/P i ESC/P2. Funkcjonalność jest różna w zależności od zastosowanej wersji lub wyselekcjonowanej podczas ustawienia drukarki (XL, XL24 lub AGM z przykładami takich wersji).

Język sterowania IBM ProPrinter w drukarkach od różnych producentów może się nieznacznie różnić co do funkcjonalności. Zwykle różnice są dodatkowymi funkcjami nieuwzględnionymi w standardzie.

#### 3.1.2. Epson ESC/P, ESC/P2

ESC/P (Epson Standard Code).

ESC/P2 (Epson Standard Code, Level 2).

Język drukarki ESC/P został stworzony przez Epsona do stosowania we wczesnych drukarkach mozaikowych. Obecnie używany jest także przez atramentowe i laserowe drukarki Epsona jak również przez wiele innych drukarek mozaikowych znajdujących się na rynku. ESC/P2 jest ulepszeniem ESC/P, posiada np. nowe funkcje do skalowania czcionek, dla drukowania grafiki rastrowej, itp.

Drukarki ESC/P lub ESC/P2 różnych producentów mogą się nieznacznie różnić w swojej funkcjonalności. Zwykle różnice te, to dodatkowe funkcje niedostarczone z wersji oryginalnej Epsona.

Informacje o stosowanej wersji ESC/P i ESC/P2 znajdują się w dokumentacji drukarek poszczególnych producentów. Niektóre pozycje to krótkie wyliczenie dostępnych funkcji z przyk-

adami w BASIC u. Można te znaleźć tam tablice z zestawem symboli i tablice z szerokościami czcionek.

ESC/P2 Reference Manual firmy Epson, zawiera wykaz kodów w ESC/P i ESC/P2, jak również kompletny opis również w rozkazach dla drukarki. Najnowsza wersja pochodzi z sierpnia 1992.

## 3.2. Języki opisu strony

Innym podejściem do sterowania drukarką jest zdefiniowanie całkowicie nowego języka opisu wydruku. W przeszłości używano kilku takich języków, ale obecnie standardem stał się PostScript [8], [6].

### 3.2.1. PostScript

PostScript jest językiem opisu strony produkowanym przez Adobe Systems Inc. od początku lat 80-tych. Adobe został utworzony w 1982 r. przez dr John E. Warnock i dr Charles M. Geschke. Dostarcza on instrukcje do opisu strony informacji. Ponieważ wymaga więcej pamięci, niż większość języków w opisie strony, był pierwszym, szeroko dostępnym produktem, który kontrolował wielkość liczb czcionek i grafiki.

Pierwsza wersja opublikowana w 1985 r. jest nazywana Level I, obecne ulepszenie jest nazywane Level II (nie należy mylić z: wersją PostScript 47.0 lub 2011.110, ani z liczbą w linii początkowej jakiegokolwiek wydruku PostScript, jak `!PS-Adobe-3.0`). Poziom PostScript używa i rozszerzenie wersji o interpreter, poszerzając możliwości operacje.

W obiegu znajduje się kilka klonów PostScript u, lecz, ze względu na koszty opłaty licencyjnej za Adobe interpreter; najbardziej znanym jest GhostScript. Inne, wbudowane bezpośrednio do drukarek laserowych lub dostosowywane przez dodatkowe kasety (ang. cartridge) to: Phoenix Page, BrotherScript, Page Styler, True Image, Turbo PS, PDL i KPDL. Można się o nich, oczywiście, dowiedzieć kompatybilne z PostScript em, lecz ta kompatybilność dotyczy czasami przy nadawaniu czcionki, przy manipulowaniu czcionkami (tj. wstawianiu tabeli metrycznej lub nowych symboli) oraz przy innych operacjach. Przy użyciu klonów nie ma jednak problemu z drukowaniem prostego tekstu czy grafiki.

Zestaw symboli PostScript jest stosowany do pisania programu PostScript, a nie symboli drukowanych czcionek PostScript u.

Adobe zaleca używanie tylko drukowalnego podzestawu symboli ASCII w programie PostScript u, spacje, tabulatory i symbole CR oraz LF. PostScript nie zabrania używania symboli poza tym zestawem, lecz stosowanie ich może spowodować problemy z przesyłaniem (np. transfer 8-bitowych symboli przez 7-bitową linię seryjną do drukarki). Dla przedstawienia 8-bitowych symboli poza ciągiem należy użyć formuły `\ddd`.

Plik przeznaczony dla drukarki jest w rzeczywistości programem komputerowym w języku

PostScript, który rysuje dany obraz. PostScript używany jest do wykonywania operacji cięć związanych z generacją obrazu.

Programy PostScriptowe rysują grafik zasadniczo na dwa sposoby. Pierwszy sposób polega na narysowaniu mapy pikseli. Jeden z podstawowych operatorów w PostScriptu wczytuje sekwencję pikseli i wyświetla je w prostokątnym obszarze na stronie. To podejście nadaje się do dołączania do dokumentu obrazów zeskanowanych lub pobranych z obrazu monitora, lub dowolnych innych obrazów, które już istnieją w formacie mapy bitowej.

Alternatywnym sposobem użycia PostScriptu jest tworzenie rysunku w formacie wektorowym lub metapliku. Istnieją operatory do rysowania linii, okręgów, krzywych, prostokątów itp., i z tych elementów można budować grafik.

Mogą również być te dwa podejścia. Dla przykładu: operatory linii, prostokątów i krzywej mogą być używane do wyznaczenia obszaru, który zostanie następnie użyty jako maska wycinania do sterowania wyświetlaniem mapy bitowej [8], [11].

### 3.2.2. HP PCL i PCL

Język PCL został stworzony przez firmę Hewlett-Packard na potrzeby produkowanych przez siebie drukarek (laserowych i atramentowych). Wersje języka PCL są numerowane od 1, a do obecnej wersji 5e.

Krótką historią języka PCL (na podstawie HP's Printer Language Technical Reference Manual):

PCL 1 - Funkcjonalność druku i przestrzeni jest bazą dla funkcji dostarczonych dla prostego, wygodnego stanowiska pracy pojedynczego użytkownika.

PCL 2 - Funkcjonalność EDP (ang. Electronic Data Processing - elektroniczne przetwarzanie danych), jest ulepszeniem PCL 1. Zostały dodane funkcje dla celów ogólnych, system drukowania dla wielu użytkowników.

PCL 3 - Funkcjonalność Office Word Processing jest ulepszeniem PCL 2. Zostały te dodane funkcje dla podwyższenia jakości wydruku, obróbka dokumentów w biurach (drukarki z rodziny HP DeskJet).

PCL 4 - Formatowanie strony jest ulepszeniem PCL 3. Zostały dodane funkcje nowych możliwości drukowania strony (drukarki: HP LaserJet, HP LaserJet IIP (PCL 4,5))

PCL 5 - Funkcjonalność Office Publishing jest ulepszeniem PCL 4. Nowe możliwości publikowania w tym skalowanie czcionki i grafik HP-GL/2 (Drukarki HP LaserJet III, HP LaserJet 4 (PCL 5e)).

Wersje PCL różnią się pod względem funkcjonalności (np. typ czcionek, czcionki mapy bitowej, czcionki skalowania (IntelliFont, TrueType), metody kompresji grafik, ugniatanie grafik przez HP

czcionki skalowanie (metrics, true type), metody kompresji grafiki, wsparcie grafiki przez HP LaserJet III).

PCL jest najbardziej rozpowszechnionym językiem drukarek na obecnym rynku drukarek laserowych. Wisko producent w drukarek laserowych stosuje do nich języki PCL 4 lub PCL 5.

PJL (ang. Printer Job Language) został tak stworzony przez HP, aby dostarczyć metody na zmian parametr w poziomie pracy i stanu odczytu pomiędzy drukarką, a głównym komputerem. PJL może być stosowany na początku drukowania do ustawienia niektórych parametrów, jak język drukarki (PCL, PostScript lub inne), rozdzielczość (300 lub 600 dpi), ilość kopii, itp.

PJL jest obecnie stosowane w następujących drukarkach HP: LaserJet III Si, rodzina LaserJet 4, PrintJet XL 300 oraz DesignJet.

PJL jest również stosowane w serii 5 drukarek LaserJet [11], [7], [10].

### 3.3. Inne języki sterowania drukarek

Na rynku znajduje się wiele innych unikalnych języków w sterowania drukarkami. Następująca lista nie jest więc kompletna (intencją jest tylko wspomnienie o nich, a nie dokładny opis). Kolejno wymienionych języków w niej nie ma nic wspólnego z ich wagą na rynku.

Advanced Function Printing (AFP): jest używany w IBM Mainframes dla drukarek stronicowych. Jest to funkcja prezentacji zestawu Mixed Object Document Content Architecture (MO:DCA), który jest częścią IBM System Application Architecture.

Właściwie nie drukuje się za pomocą MO:DCA, stosuje się IPDS (ang. Intelligent Printer Data Stream).

Informacje obejmują PTOCA (ang. Print Text Object Content Architecture), GOCA (ang. Graphic Object Content Architecture), IOCA (ang. Image Object Content Architecture) jak też inne.

IPDS jest językiem druku IBM SAA. Działają z różnymi czcionkami bitmapowymi, z prostymi podstawowymi grafikami, oraz z wizerunkami bitmapowymi. Ze względu na prostotę wyobrażenia modelu, może być użyty do kierowania obsługą szybkich drukarek laserowych.

Diablo 630: początkowo używany z drukarkami rozetkowymi i maszynami do pisania. Toleruje tylko sekwencje tabulacji, odstępy linii i symboli, selekcję atrybutów (ang. bold, double-strike, underline - pogrubiony, podwójne uderzenie, podkreślonny), ruchy poziome w obu kierunkach, proporcjonalne odstępy i automatyczne wyśrodkowanie i justifyfikację - między innymi. Ten język jest czasami używany przez innych producentów jako baza dla ich specyficznych emulacji.

CaPSL: (ang. Canon Printing System Language) był poprzednim językiem standardowym dla laserowych drukarek Canon. Inną występującą nazwą to LIPS (ang. Laser-beam Image System). Ze CaPSL krąży się pewna historia. Canon produkuje silniki do drukarek dla HP



systemy. Za CaPSL kryje się pewna historia - Canon produkuje silniki do drukarek dla HP, nie mia jednak licencji na stosowanie HP PCL w swoich własnych drukarkach. Stąd powstała potrzeba znalezienia własnego języka drukarek. Lasery Canona tradycyjnie zawierały CaPSL, IBM ProPrinter, ESC/P i emulacje PostScript, ale nie PCL). Ta część kontraktu między Canonem a HP najwidoczniej uległa przedawnieniu, ponieważ obecnie Canon oferuje drukarki z PCL 4 i PCL 5.

LIPS wspomaga Diablo 630 (ustawienie fabryczne dla trybu polecenia - ang. command mode), tryb ISO (dla drukowania tekstu i grafiki rastrowej) oraz tryb VDM (dla grafiki wektorowej i drukowania symboli)..

RENO: jest to standardowy język sterowania dla drukarek Agfa (P400, P3400, itp.). RENO jest rodzajem języka opisu strony. Jego funkcjonalność jest ogromna: oprócz drukowania tekstu z różnymi skalami czcionek, można rysować linie, zawiera ikony (Windows) wzorami, umożliwia wyrażenie programowania (if-then-else, repeat-until, set, use and print variables, operacje push i pop), umożliwia drukowanie swoich własnych symboli i przenosi dane do pamięci operacyjnej drukarki lub na twardy dysk, czy dyskietkę jeśli jest dołączona.

Prescribe: jest to język opisu strony stworzony przez Kyocera. Jego zaletą jest fakt, że można go osadzić w innej obecnej emulacji drukarki na urządzeniach Kyocera. Drukarki Kyocera wspomagają HP PCL, klon HP-GL nazywany KC-GL, Epson ESC/P (tryb LQ-850), IBM ProPrinter X24E, Diablo 630, ogólną emulację drukarki liniowej, i jako opcję KPDL, klon PostScript.

DEC: DEC posiada swoje własne unikalne języki dla drukarek laserowych (LN03, LN06).

ANSI: Dan McGowan z Mannesmann Tally twierdzi, że: Drukarki Mannesmann Tally, które posiadają ANSI są podstawą dla notacji ANSI 3.64. Jest to raczej tylko specyfikacja pokrywająca ogólne funkcje peryferyjne. Drukarki produkowane w U.S.A., w większości posiadają wszystkie rozkazy ANSI 3.64 należące do funkcji drukarki. Drukarki produkowane w Niemczech seryjne drukarki głowicowych (ang. flying serial head printers). Posiadają MTPL (ang. Mannesmann Tally Printer Language), który jest oparty na ANSI 3.64 lecz zawiera dodatkowe polecenia [8].

### 3.4. Języki sterowania ploter w

Nazwa HP-GL (ang. Hewlett-Packard Graphics Languages - język graficzny firmy Hewlett-Packard) oznacza język stosowany pierwotnie do obsługi ploterów w firmie Hewlett-Packard. Z datą powstania tego języka przyjmuje się rok 1976, w którym pojawił się na rynku wspomniane plotery. Wraz z udoskonaleniem tych urządzeń język HP-GL został wzbogacony o nowe polecenia, a obecnie jego druga wersja jest oznaczona symbolem HP-GL/2.

Język HP-GL posiada także obsługę kres w wektorowych, co oznacza w praktyce możliwość

ci sterowania urządzeniem, podobne do tych jakie daje PostScript. Dużą częścią języka HP-GL/2 została włączona do poziomu języka PCL, w którym istnieją specjalne polecenia przebieg sterowania z klasycznego PCL na język HP-GL i odwrotnie.

Składnia języka HP-GL/2 jest prosta. Wszystkie instrukcje są dwuznakowymi skrótami ich nazw. Po skrótach następują parametry. Mogą one być obowiązkowe lub opcjonalne. Jako separator oddzielający od siebie parametry może służyć odstęp lub przecinek. Preferowanym separatorem jest przecinek. Każda instrukcja kończy się terminatorem. Terminatorami mogą być: średnik, pierwszy znak następnej instrukcji lub odstęp [10], [7], [6].

# Systemy t umacz ce

## 1. Konwersja między typami plików

Ponieważ istnieje wiele formatów języka sterowania drukarką, jest rzeczą oczywistą, że często występuje potrzeba przetwarzania formatu pliku wydruku na inny. Zależy od rodzaju formatu źródłowego i docelowego, może to być zadanie trywialne, ale może być niemożliwe do wykonania. Obrazy zapisane w pliku wydruku mogą mieć różną postać. Może to być obraz bitmapowy, wektorowy lub wręcz zwykły tekst ASCII. Uwzględniając format języka rozumianego przez drukarkę, może zachodzić potrzeba konwersji formatu obrazu zapisanego w pliku wydruku [6].

### 1.1. Mapa bitowa na mapę bitów

Przetworzenie jednego typu mapy bitowej na inny jest zwykle łatwe. Jeśli przebrniemy już przez szczegóły kodowania pliku, będziemy mieć do czynienia z pikselami, które są niemal zawsze takie same, więc konwersja jest prosta.

Jeśli przebiega z formatu bardziej do mniej deskryptywnego, np. z kolorowego do czarno-białego lub ze stopniowaniem szarości, istnieją znane metody przekształcania z zachowaniem najlepszej możliwej jakości obrazu.

### 1.2. Format wektorowy na wektorowy

Podstawowym zagadnieniem przy przekształcaniu pomiędzy formatami wektorowymi jest dostosowanie niekorzystnych semantyk poszczególnych formatów i tak, w pewnym stopniu, układ w sposób odpowiedni. W najprostszym przypadku rozkazy są tłumaczone jeden do jednego, np. polecenie rysowania linii na takie samo polecenie w pliku wynikowym. Problemy pojawiają się, gdy dwa formaty nie mają odpowiadających sobie rozkazów. Jeśli format oryginalny ma polecenie rysowania elipsy, a w formacie docelowym ono nie występuje, to trzeba zastosować jedną z dostępnych metod translacji. Można przybliżyć elipsę okręgiem lub łańcuchem krótkich odcinków. Jeśli format przeznaczenia ma możliwość oddzielnego skalowania osi x i y, to dobrą metodą może być ustalenie na nich różnych jednostek i narysowanie okręgu, który zostanie przeskalowany w elipsę.

### 1.3. Format wektorowy na mapę bitów

Przetwarzanie obrazu z postaci wektorowej do mapy bitowej zwane jest rastryzacją. Polega ono na znalezieniu zbioru pikseli odpowiadającego każdemu wektorowi obrazu oryginalnego. Podstawowy algorytm rastryzacji jest znany od 1963 roku, kiedy to został opublikowany przez Bresenham; okręgi i łuki mogłyby być rysowane przy użyciu tej samej metody.

Ten rodzaj konwersji dotyczy takich samych zagadnień jak rysowanie obrazu wektorowego na ekranie rastrowym czy drukarce laserowej, więc często daje się przystosować kod wyświetlania do konwersji na mapę bitową.

## 1.4. Mapa bitowa do formatu wektorowego

Konwersja mapy bitowej na postać wektorową jest daleko bardziej złożona niż dowolne z poprzednich przekształceń. Obecnie istniejące zadowalające algorytmy wykrywania krawędzi (ang. edge detection). Mogą one posłużyć do znalezienia linii w mapie bitowej, ale tylko w najprostszych przypadkach. Problem znajdowania linii w zeskanowanym obrazie (np. w dokumencie przesłanym faksem) pozostaje nierozwiązany.

# 2. Programy konwertujące

## 2.1. Zamiana PostScriptu na inne standardy

Najbardziej znanym programem do interpretacji PostScriptu jest GhostScript. Przez wywołanie GS z opcją -help można uzyskać informacje o dostępnych urządzeniach. W GhostScript możliwa jest emulacja następujących systemów: epon, eponc, necp6, laserjet, ljetplus, ej2p, ljet3, paintjet, bj10e, djet500, djet500c, pjetxl, lbp8 (są to nazwy używane w programie). Można także dodać do programu nowe sterowniki w [8].

## 2.2. Zmiana innych standardów na PostScript

W przypadku tekstu ASCII zmiana standardu na PostScript nie jest skomplikowana. Istnieje kilka ogólnodostępnych narzędzi służących do tego celu. Najprostszym z nich jest program o nazwie a2ps.

W przypadku tekstów w innych od ASCII (ISO 8859-1 lub PC 437), lub szczególnych sekwencji kontrolnych drukarki (ang. printer specific control sequences), konwersja może być skomplikowana. Do zmiany HP PCL na PostScript służy narzędzie lj2ps, które jest bardzo pomocne w przypadku czcionek nieproporcjonalnych (jak dla HP LaserJet II). Problem komplikuje się jednak przy konwersji grafiki. Istnieje również konwerter dla HPGL (hpgl2ps), ale jest on rzadko spotykany. Do zmiany Epsona na PostScript służy filtr epon2ps [8].



## Aplikacja EmuLator

Program EmuLator jest przeznaczony dla środowiska Windows od wersji 3.1. O wyborze tego systemu zdecydowano głównie ze względu na prostotę w jego użyciu i komunikacji z użytkownikiem za pośrednictwem bogatego interfejsu graficznego. Istotną zaletą programu jest możliwość jego uruchomienia dzięki zastosowaniu bogatego interfejsu graficznego. Istotną zaletą programu jest możliwość jego uruchomienia dzięki zastosowaniu bogatego interfejsu graficznego. Istotną zaletą programu jest możliwość jego uruchomienia dzięki zastosowaniu bogatego interfejsu graficznego.

System Windows zapewnia niezależność od sprzętu. Ten sam program może wyświetlać informacje na różnych monitorach (EGA, VGA itd.) i drukować na różnych drukarkach, od mozaikowych do laserowych.

Z punktu widzenia programisty, Windows oferuje wiele gotowych elementów w interfejsie użytkownika, jak np. przyciski ekranowe, menu, okienka dialogowe, listy i okienka edycyjne.

Windows zawiera obszerny interfejs urządzeń graficznych GDI (ang. Graphics Device Interface), służący do wyświetlania tekstu i grafiki. W szczególności, interfejs ten pozwala rysować w dowolnym układzie współrzędnych.

Jako język programowania został wybrany C++ w wersji dla najnowszego w chwili tworzenia programu, kompilatora Borlanda 4.52 [4]. To narzędzie pozwala na obiektowe podejście do projektowanej aplikacji [2] oraz umożliwia wykorzystanie bibliotek klas OWL i CLASSLIB [1].

Zasadniczym przeznaczeniem biblioteki klas OWL (ang. Object Windows Library) jest dostarczenie programistom kompletnego szkieletu aplikacji dla Windows. Wykorzystana została wersja 2.5 OWL Borlanda, która pozwoliła na szybkie stworzenie interfejsu użytkownika z atrakcyjnymi, graficznymi elementami sterującymi.

Biblioteka kontenerowa Borlanda CLASSLIB dostarcza wygodnych obiektów do przechowywania oraz przetwarzania danych.

## 1. Opis programu

Program ma za zadanie rozwiązanie problemu interpretacji obrazu graficznego zawartego w pliku wydruku na ekranie komputera. Dużą liczbą języków w sterowaniu drukarkami i ploterami oraz rozmaite formy ich zapisu skłania programistę do interpretacji wybranego języka drukarki. Takie podejście można zaobserwować np. w programie GhostScript interpretującym język PostScript. Praca ta polega na napisaniu programu, który będzie interpretował dane i zuki

roscript. Praca ta polega na napisaniu programu który będzie interpretował dane j zyki sterowania.

Przyjmując założenie, że plik wydruku zawiera różne rozkazy powodujące akcje graficzne typu: wypisanie tekstu lub narysowanie grafiki oraz sekwencje zmieniające parametry drukarki. Wyniki można zobaczyć na wydrukowanej kartce, a więc można je tak przedstawić na ekranie komputera. Poszczególne języki sterowania powodują to samo działanie przy pomocy różnych kombinacji kodów. W celu zasymulowania pracy drukarki lub plotera, program zawiera zbiór podstawowych operacji wykonywanych przez te urządzenia, oraz umożliwia wykonanie tych operacji.

Istotnym problemem było wykorzystanie przez program dowolnego języka drukarki w sposób przystępny dla użytkownika, a jednocześnie nie pozwalający w każdej chwili na dodawanie nowych języków sterowania. EmuLator otwiera sterownik drukarki, który jest plikiem tekstowym w ustalonym formacie, sprawdzając również czy nie zawiera on błędów w zapisie. Plik ten zawiera także ustawienia początkowe dla danej drukarki lub plotera. Użytkownik może sam stworzyć sterownik dla każdej drukarki, korzystając z zawartych w programie operacji, bez potrzeby rekompilacji programu. Jeżeli dana drukarka używa specyficznych dla siebie mechanizmów, nieujętych w omawianym programie, sprawny programista potrafi szybko rozbudować program o nowe funkcje.

W aplikacji wykorzystane są mechanizmy MDI (ang. Multiple Document Interface), które pozwalają na przeglądanie kilku dokumentów jednocześnie. Program pozwala interpretować wydruki tylko dla jednego rodzaju drukarki w danym momencie.

Efekt końcowy można oglądać na ekranie w skali 1:1 (przy uwzględnieniu niedokładności ekranu monitora), w podglądzie wydruku całej strony, lub wydrukować na dowolnej drukarce. Wykorzystując niezależno sprzeczny kontekst urządzenia GDI, program pozwala na dokonanie konwersji znanego mu formatu drukarki lub plotera na format dowolnego urządzenia dostępnego w systemie Windows.

## 1.1. Budowa

Aplikacja powstała w oparciu o architekturę MVC (ang. Model-View-Controller), powszechnie wykorzystywaną w języku Smalltalk-80 z uwzględnieniem specyficznych możliwości biblioteki OWL 2.5. Architektura MVC dzieli aplikację na trzy warstwy:

1. Model oznacza warstwę aplikacji, w której umieszczone są wszystkie obiekty, zależne od tej aplikacji. W EmuLatorze są to wszystkie obiekty reprezentujące warstwę danych oraz operacje na nich wykonywane. W szczególności, może być to programowalny symulator, umożliwiający umieszczenie kodu w pliku wydruku na operacje graficzne, oraz klasa przetwarzająca dane pliku wydruku na obraz graficzny, wykorzystany w warstwie prezentacji.

2. View jest tzw. warstwą prezentacyjną, która odpowiada za prezentację danych. Warstwa prezentacyjna odczytuje odpowiednie informacje z warstwy aplikacji i wyświetla je na ekranie. Warstwa ta zajmuje się również oknami graficznego interfejsu użytkownika.
3. Controller jest tzw. warstwą sterującą, pośredniczącą w przekazywaniu informacji od urządzeń wejściowych (klawiatura, myszka) do pozostałych dwóch warstw.

Model ten w trakcie realizacji ulega nieznacznym modyfikacjom, oraz by podporządkowany nowym technikom znajdującym się w OWL 2.5, jak np. Document/View, w zależności od rozwiązywanego problemu.

### 1.1.1. Warstwa sterująca

#### 1.1.1.1. Główna klasa aplikacji

*class emuApp: public TApplication* - najważniejsza klasa aplikacji, której implementacja znajduje się w plikach *emuapp.cpp* i *emuapp.h*, tam też zdefiniowano pomocnicze klasy *TFileDrop*, zajmujące się obsługą wczytywanych dokumentów w metodzie *przeniesienie* (ang. drag and drop). Klasa *emuApp* jest bazą dla wszystkich obiektów w aplikacji. W funkcji *InitMainWindow* tworzone są obiekty klas *emuMDIFrame*, *emuMDIClient*, adowane z zasobów w elementy graficzne programu (takie jak ikony i menu), wstawiane w ramki *emuMDIFrame* linia statusu (*TStatusBar*) i pasek narzędzi (*TControlBar*), ustawiane atrybuty głównego okna oraz uaktywniane są przestrzenne elementy dialogu funkcją *EnableCtrl3d(true)*. Oprócz obsługi standardowych komunikatów w Windows klasa *emuApp* jest odpowiedzialna za następujące zadania dla omawianej aplikacji:

obsługa systemu pomocy - wczytanie pliku pomocy w trybie zwykłym lub kontekstowym;

tworzenie nowego dokumentu i widoku - podczas otwierania nowego dokumentu i tworzenia widoku, użytkownik informowany jest, jeśli plik sterownika nie jest wczytany; przy wczytywaniu nowego pliku sterownika zamykane są wszystkie okna;

przekazanie komunikatów sterujących dla widoku dokumentu pliku wydruku (*emuView*) - klasa *emuApp* przekazuje widokowi aktualnego *emuView* dane przebiegu do następnej lub poprzedniej strony informując w pasku narzędzi o jej bieżącym numerze;

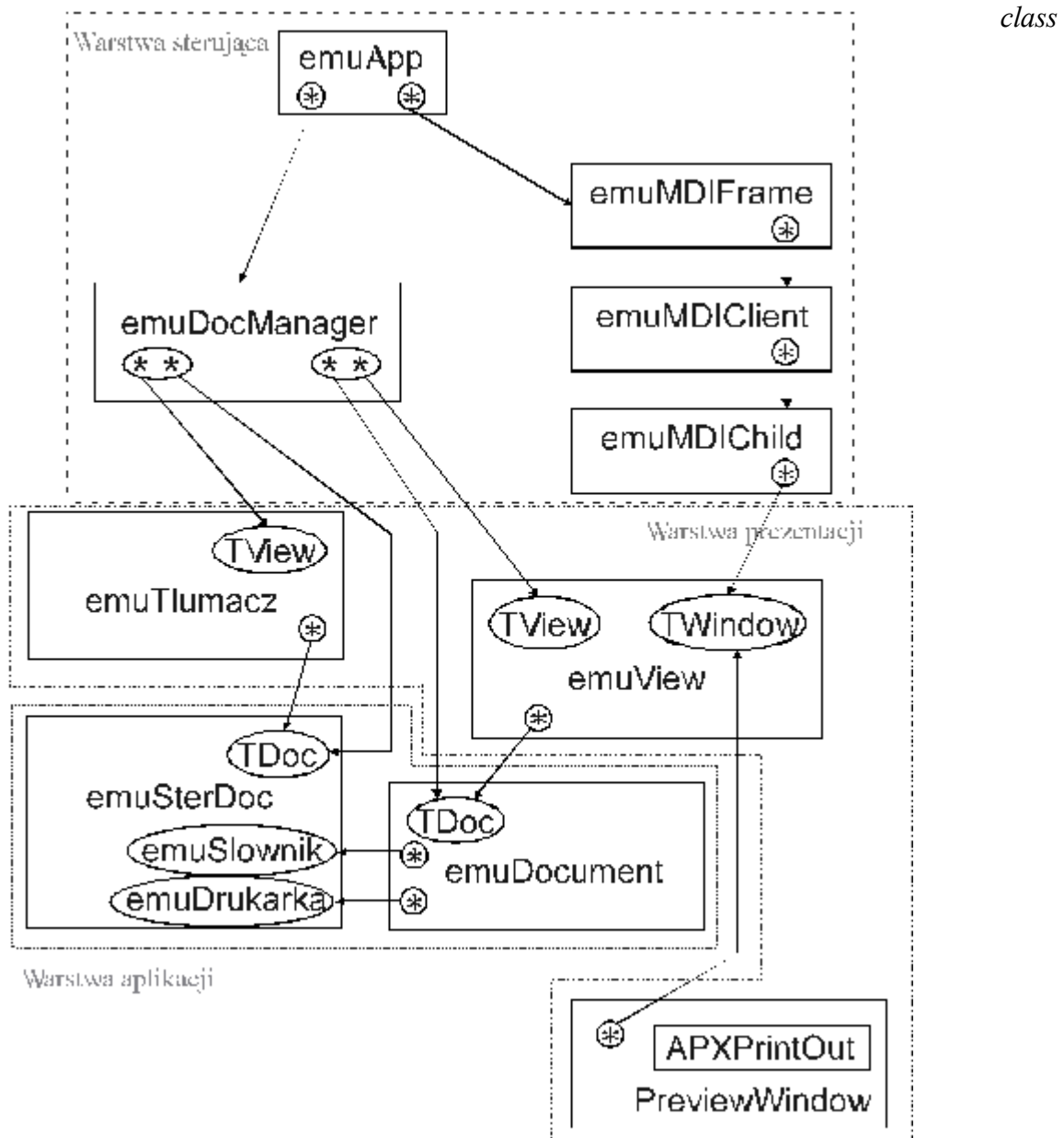
uaktualnianie informacji w linii statusu - klasa *emuApp* informuje o aktualnie wczytanym sterowniku drukarki lub jego braku, a w trakcie wyboru komendy z paska narzędzi lub menu wyświetlana jest odpowiedź.

#### 1.1.1.2. Klasy sterujące aplikacją MDI

*class emuMDIFrame: public TDecoratedMDIFrame* (*emmdifrm.cpp*, *emmdifrm.h*) - zawiera widoczne elementy sterujące oraz obsługę wszystkich komunikatów sterujących przeznaczonych



dla okien potomnych *emuMDIChild* kieruj c je do *emuMDIClient*.



Rys. F Hierarchia g wnych klas aplikacji

*emuMDIClient*: `public TMDIClient` (mmdicnt.cpp, mmdicnt.h) - odpowiada za zarz dzanie obiektami okien *emuMDIChild* w odpowiedzi na dania z menu Okno.

*class emuMDIChild*: `public TMDIChild` (mmdichld.cpp, mmdichld.h) - definiuje podstawowe zachowanie okna interfejsu MDI, zawieraj c w sobie wska nik do obiektu *TWindow*, b d czego widocznym elementem widoku dokumentu.

#### 1.1.1.3. Klasa steruj ca dokument w

*class emuDocManager*: `public TDocManager` (emdcnmngr.cpp, emdcnmngr.h) - obiekt

zarządzający listami, zarejestrowanych szablonów. Makro

*DEFINE\_DOC\_TEMPLATE\_CLASS* tworzy szablony i obiekty dokumentów z widokami, na podstawie których *TDocManager* obsługuje standardowe komendy menu *Plik*, przekazując je odpowiednim dokumentom. Jest również odpowiedzialny za wyświetlenie elementów w interfejsie użytkownika do wyboru pliku i widoku.

## 1.1.2. Warstwa prezentacji

### 1.1.2.1. Klasa prezentacji danych sterownika drukarki

*class emuTlumacz: public TView* (*emutlmcz.cpp*, *emutlmcz.h*) - obiekt pośredniczący w dostępie do danych zawartych w dokumencie. *emuTlumacz* pozwala na uzyskanie wskaźnika do klasy słownika (*emuSownik*) tworzonego w obiekcie dokumentu sterownika drukarki (*emuSterDoc*). Jako jedyny widok w aplikacji, *emuTlumacz* nie posiada skojarzonego ze sobą okienka i jest specjalnie obsługiwany w obiekcie głównym aplikacji. Dla każdego obiektu dokumentu pliku wydruku (*emuDocument*), *emuApp* ustawia wskaźnik do *emuTlumacz* (w danym momencie może istnieć tylko jeden obiekt *emuTlumacz*).

### 1.1.2.2. Klasy prezentacji danych pliku w wydruk w

*class emuView: public TWindowView* (*emuvview.cpp*, *emuvview.h*) - obiekt widoku dokumentu pliku wydruku dziedziczący w asno ci dwóch klas OWL. Jako potomek *TView* obsługuje dostęp do danych *emuDocument*. Dziedzicząc z *TWindow* jest graficzną reprezentacją dokumentu w postaci widocznego okna, którego wskaźnik jest przekazywany obiektowi *emuMDIChild*. W odpowiedzi na danie wyświetlenia kolejnej strony, *emuView* przekazuje obiektowi *emuDocument* kontekst wyświetlania do narysowania obrazu strony wydruku. Aby każda reakcja widoku na komunikat *WM\_PAINT* nie powodowała uruchamiania całego mechanizmu tłumaczenia, interpretacji i tworzenia rysunku z pliku wydruku, używany jest kontekst metapliku, pamiętany w postaci obiektu klasy *TMetaFilePict*.

*class APXPrintOut: public TPrintout* (*apxprint.cpp*, *apxprint.h*) - reprezentuje fizyczny dokument wysyłany do drukarki. Obiekt tej klasy jest odpowiedzialny za wyrysowanie strony na drukarce fizycznej lub w oknie podglądu wydruku.

*class PreviewWindow: public TDecoratedFrame* (*apxprev.cpp*, *apxprev.h*) - tworzy ramkę podglądu wydruku oraz obsługuje wyświetlanie jednej lub dwóch stron jednocześnie w obiekcie *TLayoutWindow*. Jest również odpowiedzialny za wydruk na fizycznej drukarce w wyniku wyboru komendy *Drukuj* aplikacji.

### 1.1.2.3. Inne klasy

*class emuEditView: public TEditView* (*emueditv.cpp*, *emueditv.h*) - klasa widoku

*class emuEditView: public TEditView (emueditvw.cpp, emueditvw.h)* - klasa widoku skojarzona z dowolnym typem pliku (bezpośrednio *TFileDocument*); pozwala na podstawowe operacje edycji, wydruku tekstu i wyszukiwania.

*class TBmpViewWindow (emuabout.cpp, emuabout.h)* - służy do przedstawienia początkowej winiety programu w postaci bitmapy.

*class emuAboutDlg: public TDialog (emabtdlg.cpp, emabtdlg.h)* - korzystając z pomocniczej klasy *ProjectRCVersion*, podaje informacje projektu, wyświetla okienko dialogu o programie.

### 1.1.3. Warstwa aplikacji

#### 1.1.3.1. Klasy dokumentu sterownika drukarki

*class emuSterDoc: public TFileDocument (emustrdc.cpp, emustrdc.h)* - reprezentuje obiekt danych dokumentu i podaje sposób ich interpretacji widokowi. Zawiera szereg metod do obsługi fizycznego pliku na dysku, cznie z obsługą strumieni. Każdy dokument może być skojarzony z kilkoma widokami, dlatego wspomaga komunikację z nimi, tworząc listę bieżących widoków i przysyłając do nich komunikaty o wszelkich zmianach. Dokumenty i widoki posiadają listę własności (*Property*), na podstawie których aplikacja decyduje o sposobie przetwarzania danych. *emuSterDoc*, wśród swoich własności, posiada wskaźniki do stworzonych przez niego obiektów *emuSownik* i *emuDrukarka* w momencie otwierania dokumentu sterownika drukarki.

*class emuSownik: public ContainerType (emuslwnk.cpp, emuslwnk.h)* - obiekt ten wywodzi się od kontenerowej klasy *sownika* tworzonej makrem *typedef TDictionaryAsHashTable<AssociationType> ContainerType*; jest zbiorem asocjacji, wraz z metodami ich dodawania i wyszukiwania. Asocjacja (*typedef THAssociation<emuMyClass, GraphicalObject> AssociationType*;) stanowi parę wskaźników do obiektów w *emuMyClass* i *GraphicalObject*. Obiekt klasy *emuDocument*, znajdując kod sterujący, wykonuje operację graficzną, wywiedzioną z *GraphicalObject*. W trakcie tworzenia *sownika* przez *emuSterDoc*, tworzone są dynamicznie i dodawane tylko te obiekty graficzne, które są potrzebne do interpretacji danego urządzenia.

*class emuMyClass (emumycla.cpp, emumycla.h)* - pojedynczy kod drukarki lub plotera, wraz z operatorem porównania pozwalającym na znalezienie go.

*class GraphicalObject (emugrobj.cpp, emugrobj.h)* - bazowa klasa dla operacji graficznych na kontekście urządzenia, z której są wirtualnie wywiedzione obiekty odpowiadające poszczególnym kodom sterującym. Obiekt tej klasy nie wykonuje żadnej operacji, lecz w celach diagnostycznych wypisuje nazwy operacji do wykonania.

*class GraphObjHPGL: public virtual GraphicalObject (emughpgl.cpp, emughpgl.h)* - bazowa klasa dla operacji graficznych specyficznych dla ploterów. Z niej wywodzi się klasa

bazowa klasa dla operacji graficznych specyficznych dla ploter w. Z niej wywodzi si klasy

konkretnych operacji graficznych *class HPGL\_0xXXXX: public virtual GraphObjHPGL*, gdzie 0xXXXX stanowi unikalny numer wewn trzny aplikacji dla danej operacji.

*class GraphObjPCL: public virtual GraphicalObject* (emugopcl.cpp, emugopcl.h) - bazowa klasa dla operacji graficznych specyficznych dla drukarek atramentowych i laserowych. Z niej wywodzi si klasy konkretnych operacji graficznych *class PCL\_0xXXXX: public virtual GraphObjPCL*, gdzie 0xXXXX stanowi unikalny numer wewn trzny aplikacji dla danej operacji.

*class GraphObjIBMPPro: public virtual GraphicalObject* (plik emugopro.cpp, emugopro.h) - bazowa klasa dla operacji graficznych specyficznych dla drukarek ig owych. Z niej wywodzi si klasy konkretnych operacji graficznych *class IBMPPro\_0xXXXX: public virtual GraphObjIBMPPro*, gdzie 0xXXXX stanowi unikalny numer wewn trzny aplikacji dla danej operacji.

*class emuDrukarka* (emudrkrk.cpp, emudrkrk.h) - klasa zawieraj ca ustawienia pocz tkowe emulowanego urz dzenia, wykorzystana r wnie w momencie tworzenia obrazu graficznego.

#### 1.1.3.2. Klasy dokumentu pliku wydruku

*class emuDocument: public TFileDocument* (emudcmnt.cpp, emudcmnt.h) - obs uguje operacje zwi zane z fizycznym plikiem wydruku. Na danie *emuView* analizuje plik, szukaj c kod w steruj cych w *emuSownik* i wykonuj c operacje graficzne wirtualn funkcj *Draw()* obiektu wywodz cego si od *GraphicalObject*.

*class emuStrona* (emustron.cpp, emustron.h) - klasa wspomagaj ca tworzenie rysunku konkretnej strony. *emuDocument* wykorzystuje zbi r stron (*typedef TlArrayAsVector<emuStrona> emuStronki;*) do pami tania pozycji strumienia pliku wydruku i ustawie drukarki (*emuDrukarka*) dla ka dej ze stron.

## 1.2. Rozbudowa

Program stanowi doskona y szkielet do rozbudowy o nowe funkcje i modu y. Wykorzystuj c technik OWL Document/View, mo na stworzy inne rodzaje widok w i dokument w dla istniej cych lub nowych obiekt w. R wnie stworzone modu y interpretuj ce pliki wydruku mog zosta atwo rozszerzone. Przy rozbudowie programu pomocne b d komentarze zamieszczone w plikach r d owych oraz klasy pomocnicze, kt re nie zosta y wbudowane w ko cowej wersji programu.

### 1.2.1. Dodawanie obiekt w widok w

Aby doda obiekt widoku, wystarczy zdefiniowa now klas wywodz c si od *TView*, w kt rej wykorzystane b d metody zawarte w obiekcie wybranego dokumentu. Je eli w widoku do

prezentacji użyte będą specjalne formy danych, należy zdefiniować je w obiekcie dokumentu. Obiekt widoku może być powiązany z dokumentem, który posiada kilka innych sposobów prezentacji. Aby program mógł je obsługiwać, należy zdefiniować odpowiednie szablony dla *emuDocManager*, np.:

```
DEFINE_DOC_TEMPLATE_CLASS(emuSterDoc, TSterListView, DocType7);
```

```
DocType7 __dvt7("Podgląd sterownika", "*.emu", 0, "EMU", dtAutoDelete);
```

gdzie: *emuSterDoc* jest istniejącą klasą dokumentu, a *TSterListView* jest klasą widoku sterownika. Widok może występować w postaci pokazanego na ekranie okna lub w dowolnej, innej formie interpretacji danych zawartych w dokumencie.

### 1.2.2. Dodawanie obiektów w dokument

Każdy obiekt dokumentu odpowiada za wczytanie danych i prawidłowe podanie ich widokom. Jeżeli dokument posiada kilka widoków, a jeden z nich pozwala na zmianę danych, konieczna jest jeszcze obsługa komunikatów przekazujących informacje o tym fakcie innym widokom. Komunikacja między widokami i dokumentami może przebiegać przez komunikaty, listy wiadomości (*Prospertities*) lub bezpośrednio odwołaniami wskazywanymi. Więcej danych o współpracy dokumentu z widokami można znaleźć w dokumentacji kompilatora Borland C++ 4.52.

### 1.2.3. Rozbudowa funkcji graficznych

W programie *EmuLator*, w trakcie interpretacji pliku wydruku wykorzystana jest funkcja:

```
int emuDocument::Rysuj( TDC& strDC, emuStrona* strona )
```

gdzie: *strDC* jest kontekstem uruchodzenia, w którym ma powstać obraz; *strona* jest wskaźnikiem do obiektu zawierającego bieżące ustawienia strony oraz kopii ustawień drukarki (*emuDrukarka*) z poprzednio stworzonej strony, lub (w przypadku pierwszej strony) ustawienia uruchodzenia przeczytane przez *emuSterDoc*. Jeżeli w strumieniu dokumentu zostanie znaleziony kod sterujący, obiekt klasy *emuSłownik* zwraca wskaźnik do asocjacji, z której jest uzyskiwany wskaźnik do odpowiedniego obiektu graficznego. Operacja jest wykonywana poprzez wywołanie wirtualnej funkcji obiektu wywodzi się od *GraphicalObject*:

```
if (znalazl)
```

```
{
```

```
if (obiekt = znalazl->Value())
```

```
obiekt -> Draw(strDC, strumyk, strona);
```

Funkcja *void GraphicalObject::Draw( TDC& dc, TInStream\* is, emuStrona\* str)* do wykonania zadania jest wywoływana z następującymi parametrami:

*TDC& dc* - referencja do kontekstu uruchodzenia, na którym należy wykonać operację graficzną;

*TInStream\* is* - wskaźnik do strumienia pliku wydruku, z którego mogą być pobierane dalsze

.....

dane do wykonania operacji;

*emuStrona\* str* - wskaźnik do obiektu, w którym można znaleźć ustawienia emulowanego urządzenia (*emuDrukarka \*druka*) oraz, w przypadku wystąpienia błędu lub końca strony należy ustawić jedną z wartości zmiennej *status*:

```
enum {
    Bład=0,
    KoniecStrony,
    KoniecPliku,
    Dalej,
};
```

Aby rozszerzyć program o nowe funkcje graficzne, należy stworzyć obiekt wywoływany wirtualnie wprost lub pośrednio (np. tak jak obiekty *GraphObjHPGL*) z klasy *GraphicalObject*, który wykonuje operację korzystając z wyżej podanych parametrów. Następnie należy wybrać unikalny numer wewnętrzny aplikacji dla operacji graficznej (z zakresu od 0x0000 do 0xFFFF) i umieścić operację dodawania obiektu w *emuSlwnk.cpp*:

```
int
emuSlownik::AddItem(const string& mc, const unsigned long int & mv, const
string& mvs)
{
    int wynik=0;
    switch (mv) {
    case 0xFFFF: {
        AssociationType assoc( new emuMyClass(mc),
        new GraphicalObject(mvs) );
        wynik=Add(assoc);
        break;
    }
}
```

Ta funkcja jest wywoływana przez *emuSterDoc*, podczas czytania pliku sterownika urządzenia z następującymi parametrami:

*mc* - kod sterujący emulowanego urządzenia;

*mv* - kod wewnętrzny aplikacji dla operacji graficznej;

*mvs* - nazwa operacji wykorzystywana w celach diagnostycznych, a pobierana z pliku sterownika.

#### 1.2.4. Rozszerzenie w asno ci emulowanych urz dze

Je eli do wykonania operacji graficznej potrzebne s parametry, kt rych nie mo na znale w definicji *emuDrukarka* (plik *emudrkrk.h*), nale y je doda do tej klasy. Inicjalizacja warto ci nast puje w *bool emuSterDoc::GetDrukarka(TInStream\* is)* gdzie nale y doda procedur czytania parametr w np.:

```
if (klucz.contains("PAGE SIZE")) // Rozmiar strony
{
    wiersz += SkipAll(*is);
    long x = GetLong(*is);
    wiersz++;
    if (!is->good()) goto error;
    wiersz += SkipAll(*is);
    long y = GetLong(*is);
    wiersz++;
    if (!is->good()) goto error;
    drukarka->SetRStr(x*(metric ? mm_pkt : cal_pkt),
    y*(metric ? mm_pkt : cal_pkt));
}
else
```

*klucz* jest napotkanym s owem kluczowym w pliku sterownika drukarki wed ug stosowanej konwencji:

```
!Page Size
```

Nast pnie, dowolne parametry czytane s ze strumienia tekstowego *\*is* odpowiedni dla nich funkcj . Funkcja *SkipAll(\*is)* omija wszystkie znaki w strumieniu do napotkania znaku ! na pocz tku wiersza. Zmienna *wiersz* wskazuje numer wiersza w kt rym wyst pi b d. Konstrukcja *x\*(metric ? mm\_pkt : cal\_pkt)* pozwala ustawi parametry we wcze niej ustalonych jednostkach (obecnie mm lub cal).

#### 1.2.5. Klasy pomocnicze

Podczas uruchamiania projektu mo na skorzysta z do czonych do r de klas diagnostycznych. Nale y w tym celu w czy do czanie kodu dla debugger a w opcjach projektu, oraz zmieni komentarze w pliku *emuapp.cpp*, uaktywniaj c szablony tych widok w. S to nast puj ce klasy:

*class TDumpView : public TListBox, public TView* (*dumpview.cpp, dumpview.h*) -  
 pozwala na skojarzenie z dowolnym typem dokumentu, oraz pokazuje dok adnie jego

zawarto z kodami szesnastkowymi poszczególnych bajt w pliku;

*class TInfoView : public TWindowView* (infoview.cpp, infoview.h) - pozwala na zobaczenie listy w asno ci (*Prosperities*) dowolnego dokumentu;

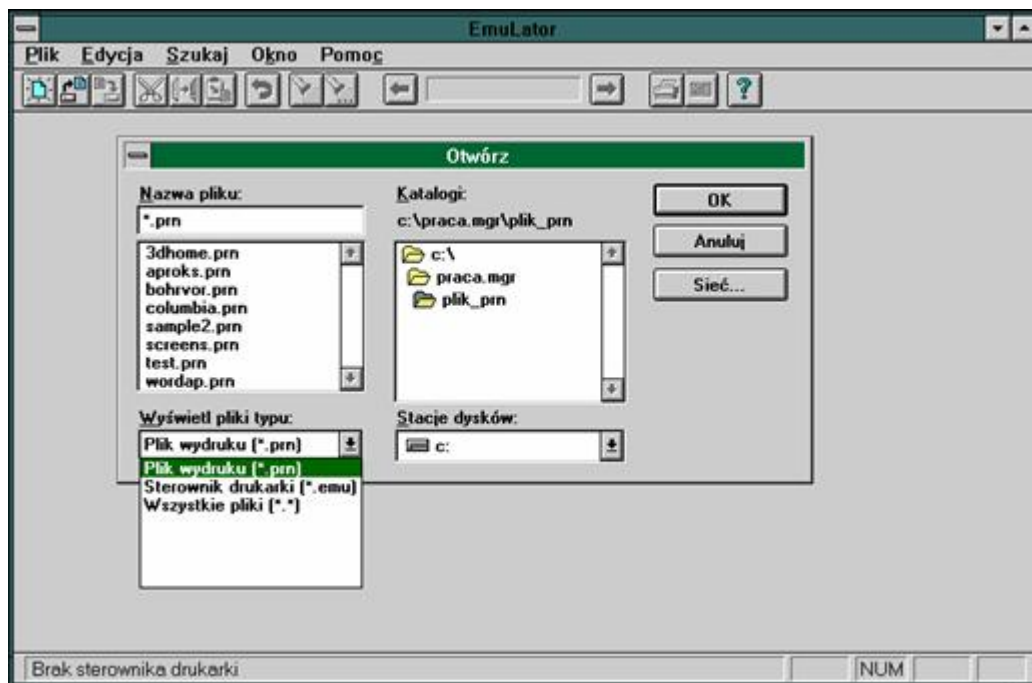
*class TSterListView : public TListBox, public TView* (strlisvw.cpp, strlisvw.h) - widok specjalnie stworzony na potrzeby dokumentu klasy *emuSterDoc*, pokazujący zawartość sterownika zainicjalizowanego plikiem sterownika drukarki.

## Obsługa programu

### Instrukcja użytkownika

Program EmuLator przed rozpoczęciem użytkowania wymaga zainstalowania na dysku twardym komputera. Jego pakiet instalacyjny znajduje się na dyskietkach typu HD 3.5". Przebieg instalacji jest typowy dla środowiska Windows.

Po zainstalowaniu program EmuLator posiada w oknie Menadżera Programów swoją grupę pod nazwą EmuLator drukarki i plotera. Ikona głównego programu aplikacji jest ikoną o nazwie EmuLator.

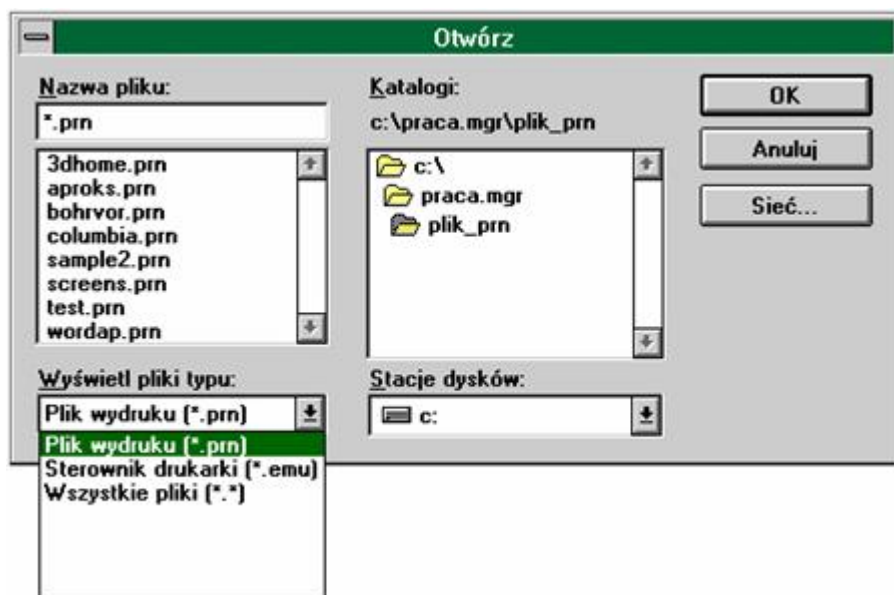


Rys. 6 Okno główne aplikacji

Po uruchomieniu aplikacji na ekranie monitora ukazuje się okno aplikacji o tytule EmuLator (Rys. 7).



Operacje obsługi programu ograniczają się najczęściej do kilku podstawowych:



Rys. H Okno otwórz

Otwieranie dokumentu

Otwarcie dokumentu zawierającego wydruk drukarki musi być poprzedzone wczytaniem sterownika urządzenia dla którego plik został stworzony. Za adowanie

sterownika dokonywane jest poprzez wybranie opcji *Otwórz* z menu *Plik* lub kliknięcie odpowiedniej ikony z paska narzędzi. Po rozwinięciu okna dialogu *Otwórz* należy ustawić typ pliku na *Sterownik Drukarki* z rozszerzeniem \*.emu i w polu dialogu *Nazwa pliku* wpisać nazwę sterownika odpowiadającego danej drukarce. Nazwa urządzenia dla którego zostanie wczytany sterownik ukaże się w linii statusu w dolnej części okna (Rys. 9).

W przypadku bieżącego zapisanego kodu rozkazu w sterowniku, podczas próby wczytywania go program wyświetli informację z numerem linii z bieżącym wpisanym kodem.



Rys. I Linia statusu z wczytanym sterownikiem.

Program jest przygotowany do otwierania pliku wydruku. Postępowanie jest zbliżone do powyższego, z tym, że jako typ otwieranego pliku należy wybrać *Pliki Wydruku* z rozszerzeniem

\*.prn. Przy ponownym otwarciu wczytanego już dokumentu program poinformuje o tym użytkownika i nie pozwoli mu na tę operację. Możliwe jest natomiast dodanie innych widoków w tego samego dokumentu; opcja *Dodaj Widok* z menu *Okno*. Zawartość pliku wydruku zostanie pokazana w oknie MDI o tytule otwartego dokumentu. Ukazanie następnej (poprzedniej) strony dokumentu (jeżeli istnieje) następuje po wybraniu opcji *Strona* w menu *Edycja* lub przycisk w strzałkach w pasku narzędzi (Rys. 10).



Rys. 10 Przeglądanie stron dokumentu

### Zamykanie dokumentu

Aby zamknąć otwarty dokument należy wybrać opcję *Zamknij* w menu *Plik*. W chwili zmiany urządzenia, dla którego pokazywany jest dokument wszystkie widoki (dokumenty) zostaną zamknięte automatycznie.

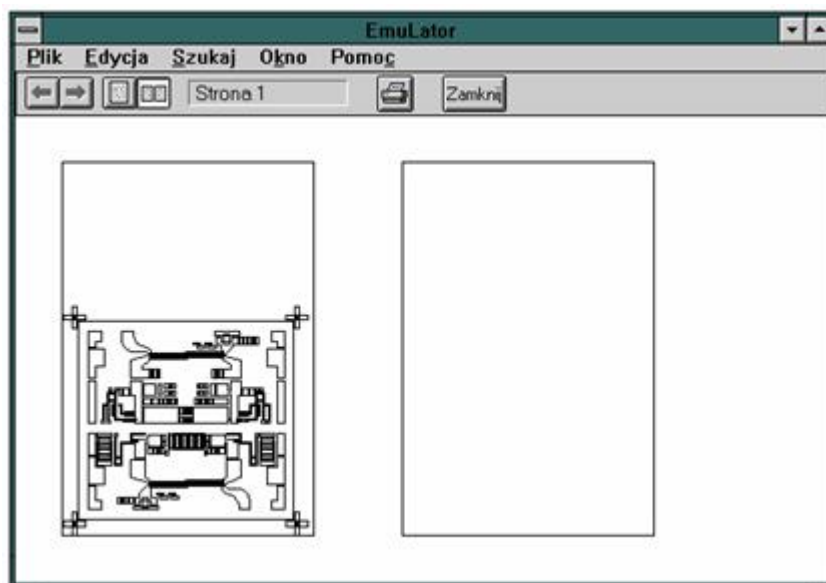
### Podgląd wydruku

Pokazanie dokumentu w takiej postaci, jak będzie on miał na stronie po wydruku można uzyskać poprzez wybór z menu *Plik* opcji *Podgląd Wydruku* lub odpowiedniej ikony z paska narzędzi. Nowy obraz okna dokumentu pokazuje jedną lub dwie strony pliku wydruku. Przegląd następnych (poprzednich) stron następuje po kliknięciu myszką w ikony strzałek na pasku narzędzi (Rys. 11).

### Wydruk dokumentu

Po przejrzaniu dokumentu można go wydrukować na urządzeniu podłączonym do stanowiska. Wydruk następuje po wybraniu opcji *Drukuj* z menu *Plik* lub po kliknięciu myszką w odpowiedni ikon paska narzędzi. Wydruk nastąpi w aktualnych ustawieniach drukarki, które można zmienić w opcji *Ustawienia Drukarki* w menu *Plik*.

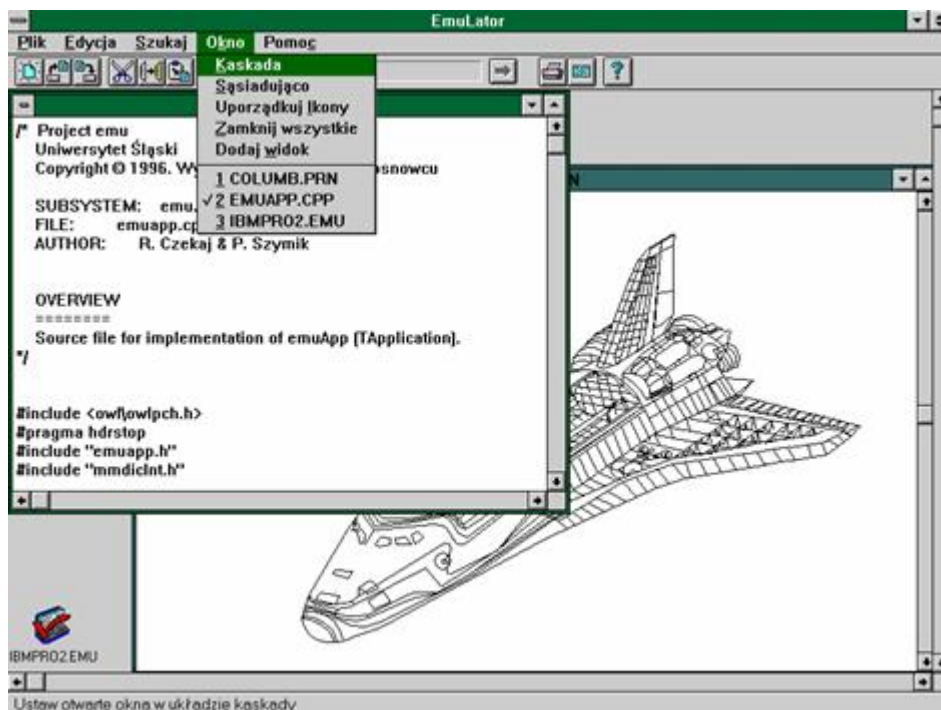
Edycja



Rys. K Podgląd wydruku

sterownik w drukarek

Sterowniki drukarek mogłyby być modyfikowane, w celu ewentualnego dopasowania do używanego urządzenia. Muszą być one wczytane w oknie dialogu *Otwórz* przy wybranym typie pliku *Wszystkie pliki (\*.\*)*. Operacje edycji umożliwiają opcje menu *Edycja*, *Szukaj*, *Okno*. Sterownik nie może być wczytany w trakcie modyfikacji.



Rys. L Edycja plik w tekstowych

## Uzyskanie pomocy

System pomocy dla aplikacji EmuLator można wywołać w menu *Pomoc*. Umożliwia on uzyskanie pomocy na zasadzie szukanego klucza słownego lub szukanego tematu. Po naciśnięciu kombinacji klawiszy Shift+F1, wskaźnik myszy zmienia kształt, pozwalając wskazać element sterujący, na temat którego potrzebna jest informacja.

## Opis sterownika

Format pliku sterownika drukarki został stworzony dla aplikacji EmuLator przy użyciu prostej konwencji zapisu. W odróżnieniu od innych programów, EmuLator wykorzystuje sterowniki, które nie są kompilowanymi programami. Zapisane są one w formacie pliku tekstowego, który nie wymaga innych dodatkowych operacji. Program posiada wbudowaną kontrolę poprawności notacji i w chwili rozpoznania nieprawidłowości użytkownik zostanie poinformowany, w której linii ona wystąpiła. Możliwa do zapisania liczba kodów sterowania drukarką jest w zasadzie nieograniczona, podobnie jak liczba możliwych ustawień początkowych drukarki lub plotera.

Każda linia, rozpoczynająca się od znaków: ! , # , \$ , % , traktowana jest przez moduł czytający kody sterownika jako komentarz. Każda linia rozpoczynająca się od znaku ! jest czytana przez moduł inicjalizujący ustawienia drukarki, a każda wiersz musi rozpoczynać się znakiem komentarza lub zawiera poprawne dane. Od momentu napotkania znaku komentarza program ignoruje kolejne znaki w strumieniu, a do napotkania znaku końca linii.

Pierwszy wiersz sterownika jest niezbyt długi, drugi bez komentarza oznacza nazwę urządzenia emulowanego. Jest podawana w linii statusu programu.

```
EMU Driver File
$ Plik sterownika plotera:
Ploter HP-GL 2
$ dla programu EmuLator
```

Opis rozkazu kopiowany jest przez moduł obsługi strumienia jako cały wiersz.

Poniższy fragment zawiera dane dla jednego kodu sterującego drukarką.

```
# Przykład sposobu opisu danego kodu plotera:
2 2 bajty kodu do przeczytania
0x2b 0xff kod sterujący urządzeniem
opis rozkazu opis słowny kodu
0xffff kod programu EmuLator dla operacji (hex od 0x0 do
# 0xffff)
```

Liczby można podawać dowolnie wg konwencji języka C:

```
22 - DEC
0x22 - HEX
022 - OCT
```

oddzielone spacją, np. 0x56 0x53

Rozkazy podzielone s na grupy. Podzia wprowadzony jest tylko ze wzgl du na zwi kszenie przejrzysto ci zapisu sterownika.

```
#####
# KODY ROZKAZ W J ZYKA HP-GL2 #
# GRUPA ROZKAZ W "0A" #
#####
```

W kolejnych wierszach pliku sterownika nast puje ju w a ciwy opis kodu rozkazu. Pierwszy parametr oznacza ilo bajt w kodu sterownika do przeczytania. Nast pny wiersz zawiera w a ciwe kody steruj ce urz dzenia. Po wierszu opisu, znajduje si unikalny numer operacji graficznej aplikacji EmuLator. W poni szym przypadku jest to jednobajtowy kod 0D oznaczaj cy powr t karetki, znajduj cy si w grupie 0A, a oznaczony numerem 11.

```
1
0x0D
powr t karetki
0x0A11
```

Opis jednego kodu steruj cego drukarki musi by oddzielony od kolejnego znakiem komentarza, np.:

```
1
0x0A
przesuw o wiersz LF
0x0A01
#
2
0x56 0x53
wyb r pr dkosci rysowania
0x131B
```

Lista wszystkich dost pnych operacji graficznych, wraz z odpowiadaj cymi numerami, znajduje si w katalogu sterownik w drukarki, w pliku *emu\_nrop.txt*.

Po opisie wszystkich niezbdnych kod w steruj cych drukarki lub plotera, nast puje opis ustawie pocz tkowych emulowanego urz dzenia. Pierwsza z pozycji pojedynczego bloku opisuj cego jedno z ustawie jest s owem kluczowym, kolejne s jego warto ciami.

```
#!!!!!!!!!!!!!!!!!!!!!! USTAWIENIA POCZ TKOWE DRUKARKI !!!!!!!!!!!!!!!!!!!
# Pierwszy wiersz rodzaj ustawien, drugi wartosc
#!!!!!!!!!!!!!!!!!!!!!!
# Rodzaj urzadzenia
!Device
#!0 # nieznany typ
#!1 # drukarka iglowa
#!2 # drukarka atramentowa
#!3 # drukarka laserowa
!10 # ploter
#!!!!!!!!!!!!!!!!!!!!!!
# Stosowana dalej jednostka
```

```
# mm  
# cal inch  
!Unit  
!mm  
#
```

Wiersze, które nie są zaznaczone jako komentarz inicjalizują wartości jako domyślne. Są one zmieniane w momencie wystąpienia w pliku wydruku kodu sterującego, modyfikując dane wartości.









## Podsumowanie

Aplikacja EmuLator stworzona została z myślą o uniwersalnym rozwiązaniu problemu interpretacji pliku w wydruku. Znajduje się ona w wersji instalacyjnej na dyskietkach dołączonych do tej opisowej pracy. Program zawiera zestaw podstawowych funkcji graficznych, pozwalających na interpretację pliku w wydruku drukarki i górowej IBM ProPrinter oraz plotera języka HPGL.

Plik wydruku może zawierać proste dane tekstowe oraz obraz graficzny. Program dokonuje analizy pliku wydruku, a po odnalezieniu kodu sterującego, wywołuje odpowiadającą mu funkcję graficzną, przedstawiając efekt działania na monitorze komputera. Przeszukiwanie pliku pod kątem występowania kodu w sterujących polega na porównywaniu znaków w pobranych ze strumienia pliku wydruku ze znanymi kodami zapisanymi w pliku tekstowym sterownika. Po rozpoznaniu kodu sterującego, następujące bajty danych są traktowane, jako argumenty wywołania operacji graficznej. Nierozpoznana sekwencja znaków w traktowana jest, jako prosty tekst ASCII pokazywany na ekranie. W ten sposób prezentowane są także pliki zapisane w formacie tekstowym.

Program może zostać w prosty sposób przystosowany do emulacji innych rodzajów urządzeń, poprzez edycję tekstowych plików w sterowniku w drukarek w zakresie wbudowanych funkcji graficznych. Istnieje możliwość szybkiego stworzenia następujących obiektów w wykonujących inne funkcje drukarki lub plotera. Wiele si to jednak z pewnością kłopot rekompilacji programu.

W opisywanej postaci program może być przystosowany do czytania dowolnego, nieznanego formatu pliku w wydruku, jak również innych plików zawierających dowolne dane możliwe do pokazania na ekranie monitora (np. graficzne wektorowe i bitmapowe, edytor tekstowy, itp.).

Trzeba przyznać, że uniwersalność stworzonego programu nie wpłynęła pozytywnie na szybkość analizy. Kody sterujące wyszukiwane są, począwszy od najdłuższego słowa znajdującego się w sterowniku. Metoda ta pozwala ujednolicić obsługę różnych formatów sterowania, tzn. można czytać zarówno języki sterowania w formacie tekstowym (np. PostScript), jak również sekwencje sterujące rozpoczynające się od znaku ESC (np. PCL i ESC/P). Jednak, przy plikach zawierających wiele znaków ASCII, szybkość analizy zmniejsza się, wraz ze wzrostem długości słowa sterującego. Możliwa jest modyfikacja programu przez zastąpienie kontenerowej klasy Borlanda (z której wywiedziony jest *emuSłownik*) szybszą strukturą danych z efektywniejszymi metodami wyszukiwania. Innym rozwiązaniem jest stosowanie różnych metod analizy pliku dla każdego formatu. Powoduje to jednak konieczność pisania przez programistę osobnego modułu dla każdego z nich, podobnie jak w tradycyjnych kompilowanych sterownikach i filtrach importu.

Możliwe jest udoskonalenie programu, przez wzbogacenie go o mechanizm wymian

danych przy pomocy Schowka, lub nawet OLE 2. Przez dopisanie odpowiednich funkcji do klas wywiedzionych z *GraphicalObject*, mo liwa mog aby by nawet kontrola i poprawa poszczeg lnych obiekt w, np. skalowanie dla bitmap, dowolne transformacje dla rysunk w wektorowych, lub zmiana rodzaju czy wielko ci czcionki.

EmuLator pozwala na wydruk ogl danych plik w, po uprzednim ich zinterpretowaniu. Wydruk danych mo liwy jest na dowolnej drukarce, dost pnej w rodowisku Windows. Ze wzgl du na niezale no sprz tow kontekstu urz dzenia GDI, program umo liwia dokonanie niejawnej konwersji dowolnego znanego mu formatu, w spos b pozwalaj cy mu przes a wyniki do dowolnego urz dzenia dost pnego w systemie Windows.

Bior c pod uwag powy sze rozwa ania oraz spostrze enia wynikaj ce ze sposobu konstrukcji i dzia ania aplikacji EmuLator, konstruktorzy uwa aj , e cel oraz za o enia tej pracy zosta y w pe ni zrealizowane.



## Literatura

- [1] Barkakati N.: *Grafika i animacja w Windows*. Warszawa, Intersoftland 1994, (t um. z ang.).
- [2] Barteczko K.: *Programowanie obiektowe; Praktyczne wprowadzenie do programowania obiektowego w j zyku C++*. Warszawa, LUPUS 1993
- [3] Dro d ewicz P.: *Programowanie dla Windows w j zyku C*. Warszawa, Lynx-SFT 1994.
- [4] Faison T.: *Borland C++ 4.5 programowanie obiektowe*. Warszawa, Oficyna Wydawnicza READ ME 1996, (t um. z ang.).
- [5] Klein M.: *Przewodnik po bibliotekach DLL i sposobach zarz dzania pami ci* . Warszawa, Intersoftland 1994, (t um. z ang.).
- [6] Levine J.: *Programowanie plik w graficznych w C/C++*. Warszawa, Translator 1994, (t um. z ang.).
- [7] Marciniak A.: *J zyk PCL*. Pozna , NAKOM 1992.
- [8] McCoy B.C.: *Summary - printers FAQ Home Page*. Usenet, comp.periphs.printers Frequently Asked Question (FAQ) List, 06.08.1996
- [9] Osiak S.: *PostScript krok po kroku*. Warszawa, Agencja Wydawnicza M&M 1991.
- [10] Smith N.E.: *Drukarki laserowe*. Warszawa, ZNI MIKOM 1995, (t um. z ang.).
- [11] Sowi ski R.: *Gwiazda plotera*. CADmania Nr 5 (11), listopad 1995
- [12] Wac awek R.: *Programowa obs uga drukarek laserowych*. Warszawa, Komputerowa Oficyna Wydawnicza HELP 1992.
- [13] Wac awek R.: *Windows od kuchni*. Warszawa, Komputerowa Oficyna Wydawnicza HELP 1993.
- [14] Zalewski A.: *Programowanie w j zykach C i C++ z wykorzystaniem pakietu Borland C++*. Pozna , NAKOM 1995.
- [15] *Drukarka mozaikowa D-100MPC*. B onie, Zak ady Mechaniczno-Precyzyjne Mera-B onie 1991.
- [16] *Instrukcja obs ugi drukarki LC-20*. Warszawa, Intersoftland 1991.

## Streszczenie

W omawianej pracy przedstawione zostały techniczne aspekty działania nowoczesnych drukarek i ploterów oraz sposoby sterowania tych urządzeń za pomocą języka w sterowania wydrukiem. Opisany program komputerowy powstał w wyniku analizy poszczególnych formatów języka sterujących.

Aplikacja EmuLator umożliwia emulację działania wybranej drukarki i plotera, oraz pozwala na prezentację graficznych plików w wydruku na ekranie komputera. Do budowy programu wykorzystano środowisko graficzne Windows 3.11 oraz pakiet narzędzi programistycznych Borland C++ w wersji 4.52.

W programie EmuLator znajduje się zestaw podstawowych funkcji graficznych, pozwalających na interpretację plików w wydruku drukarki igłowej IBM ProPrinter oraz plotera języka HPGL. W zakresie tych funkcji, każdy użytkownik może łatwo przygotować aplikację do analizy plików w wydruku dla innych urządzeń poprzez edycję tekstowych plików sterowników drukarek.

Dokładna dokumentacja oraz komentarze w plikach źródłowych programu pozwalają na rozbudowę programu o nowe funkcje graficzne dla innych urządzeń. Obiektywne w szczególności języka C++ umożliwia wzbogacenie aplikacji o funkcje, które nie zostały przewidziane przez twórcę.

Program EmuLator, wraz z systemem pomocy, dostępny jest na dyskietkach w wersji instalacyjnej.



# Spis treści

<b>WST P.....</b>	<b>3</b>
<b>TECHNICZNE ASPEKTY PRACY DRUKAREK I PLOTER W 4</b>	
1. KLASYFIKACJA DRUKAREK.....	4
1.1. Drukarki mozaikowe (drukarki uderzeniowe).....	4
1.2. Drukarki Daisywheel i maszyny do pisanie.....	5
1.3. Drukarki atramentowe.....	5
1.4. Drukarki laserowe i drukarki LED.....	6
1.5. Drukarki kolorowe.....	7
1.6. Inne rodzaje drukarek.....	9
2. KLASYFIKACJA PLOTER W.....	10
2.1. Plotery tablicowe.....	10
2.2. Plotery pisakowe (b bnowe).....	10
2.3. Plotery atramentowe.....	11
3. J ZYKI STEROWANIA DRUKAREK I PLOTER W.....	12
3.1. Rozszerzone formaty tekstowe.....	13
3.2. J zyki opisu strony.....	14
3.3. Inne j zyki sterowania drukarek.....	16
3.4. J zyki sterowania ploter w.....	18
<b>SYSTEMY T UMACZ CE.....</b>	<b>19</b>
1. KONWERSJA MI DZY TYPAMI PLIK W.....	19
1.1. Mapa bitowa na map bitow .....	19
1.2. Format wektorowy na wektorowy.....	19
1.3. Format wektorowy na map bitow .....	20
1.4. Mapa bitowa do formatu wektorowego.....	20
2. PROGRAMY KONWERTUJ CE.....	20
2.1. Zamiana PostScript u na inne standardy.....	20
2.2. Zmiana innych standard w na PostScript.....	20
<b>APLIKACJA EMULATOR.....</b>	<b>22</b>
1. OPIS PROGRAMU.....	22
1.1. Budowa.....	23



<i>1.2. Rozbudowa.....</i>	<i>30</i>
<b>PODSUMOWANIE.....</b>	<b>35</b>
<b>LITERATURA.....</b>	<b>37</b>
<b>STRESZCZENIE.....</b>	<b>38</b>
<b>SPIS TRE CI.....</b>	<b>39</b>