

UNIwersytet w Białymstoku

Wydział Matematyczno-Fizyczny

Instytut Matematyki

Barbara Urynowicz

SKŁAD DOKUMENTÓW  
ZAWIERAJĄCYCH  
DANE ZGROMADZONE  
W BAZIE DANYCH MySQL

*Praca dyplomowa napisana  
pod kierunkiem  
dr Małgorzaty Prażmowskiej*

Białystok 2003

Składam serdeczne podziękowania  
mgr Mariuszowi Żynelowi  
za pomoc przy pisaniu  
pracy licencjackiej.

Barbara Urynowicz

# Spis treści

Wstęp	1
<b>1 Implementacja</b>	<b>2</b>
1.1 Rodzaje wydruków . . . . .	2
1.2 Szablon w L <sup>A</sup> T <sub>E</sub> X-u . . . . .	3
1.3 Rozbudowa modułu zarządzania . . . . .	4
1.4 Generowanie dokumentu . . . . .	5
1.5 Algorytm doboru szerokości kolumn . . . . .	8
1.6 Biblioteka <code>brprint</code> . . . . .	11
1.7 Biblioteka <code>print</code> . . . . .	12
<b>2 Korzystanie z aplikacji</b>	<b>13</b>
2.1 Krok po kroku . . . . .	13
2.2 Przykłady . . . . .	14
<b>A Szablon <code>brprint.tex</code></b>	<b>18</b>
<b>B Biblioteka <code>print</code></b>	<b>21</b>
<b>C Biblioteka <code>brprint</code></b>	<b>24</b>
Spis literatury	28

# Wstęp

Niejednokrotnie próbowaliśmy drukować strony internetowe. Aby to zrobić wystarczy wybrać przycisk *Drukuj*, który znajduje się w oknie przeglądarki. Ewentualnie można zmienić właściwości wydruku, czyli układ strony, ilość kopii, itp. W przypadku bardziej złożonych stron, wygląd uzyskanego wydruku odbiega od dokumentu wyświetlanego w oknie przeglądarki. Coraz częściej na stronach pojawia się możliwość wyboru strony specjalnie przygotowanej do wydruku na drukarce.

Na wydziale Matematyczno-Fizycznym używana jest baza danych oparta o MySQL, w której zgromadzone są informacje o pracownikach, publikacjach, seminariach, planie studiów i rozkładach zajęć. Posiada ona swój interfejs WWW, zwany RDM, do przeglądania i aktualizacji tych danych. Często pojawia się potrzeba wykonywania dobrej jakości wydruków z bazy danych do celów sprawozdawczych. Zastosowanie opcji wydruku z przeglądarki w tym wypadku nie daje dobrych rezultatów. Najlepszą jakość oraz możliwość automatyzacji zadań przygotowywania wydruków daje L<sup>A</sup>T<sub>E</sub>X. System ten oferuje wiele programów, które pozwalają na swobodne manipulowanie wydrukami.

Naturalnym wydaje się rozbudowa istniejącej aplikacji RDM o możliwości wykonywania wydruków, a nie tworzenie odrębnej aplikacji. Celem pracy jest przygotowanie uniwersalnego zestawu bibliotek do tworzenia dokumentów w formacie L<sup>A</sup>T<sub>E</sub>X i PDF na podstawie danych zgromadzonych w bazie MySQL. Ponieważ aplikacja RDM w całości napisana jest w języku Meta-HTML, a biblioteki, które należy napisać mają być jej integralną częścią, więc wybór języka programowania jest oczywisty.

# Rozdział 1

## Implementacja

### 1.1 Rodzaje wydruków

W bazie danych Wydziału Matematyczno-Fizycznego funkcjonują dwa rodzaje wydruków. Można powiedzieć, że jeden z nich to *wydruk szczegółowy*, a drugi to *wydruk ogólny*. Wydruki szczegółowe nazywa się też raportami i ta nazwa nawet lepiej oddaje ich znaczenie. Raporty są to po prostu odpowiedzi bazy danych na zadane kwerendy. Na ogół raporty są to wydruki o z góry zadanym formacie, kwestią jest tylko wypełnienie ich danymi z bazy. Raporty programuje się oddzielnie, ze względu na ich indywidualność i specyfikę.

Raport powstaje w następujących krokach:

- sformułowanie kwerendy do bazy danych,
- wypełnienie stosownego szablonu zgodnie z odpowiedzią jaką uzyskamy z bazy,
- wygenerowanie dokumentu gotowego do wydruku w formacie wybranym przez użytkownika: DVI, PS lub PDF.

O ile w raporcie dane mogą być pobierane z wielu tabel jednocześnie, to w wydruku ogólnym tylko z jednej tabeli. W zasadzie wydruk ogólny jest wydrukiem fragmentu, bądź całości, ale pojedynczej tabeli z bazy danych. W całym procesie przygotowania takiego wydruku można wydzielić następujące kroki:

- wybór tabeli,
- wybór kolumn, które chcemy na wydruku,
- odpowiednie wypełnienie szablonu danymi z bazy,
- wygenerowanie ostatecznego dokumentu w formacie przez nas wybranym: DVI, PS lub PDF.

W tej pracy zajmujemy się wyłącznie wydrukami ogólnymi. W zasadzie większość makr, które umieściliśmy w bibliotece `print` (str. 21), jest na tyle ogólna, że może być wykorzystana przy generowaniu dowolnych wydruków.

## 1.2 Szablon w $\LaTeX$ -u

Z jednej strony zależy nam na wydrukach wysokiej jakości, z drugiej zaś chcemy aby wydruki te przygotowywane były automatycznie. Wymagania te doskonale spełnia  $\LaTeX$ -system zautomatyzowanego składu dokumentów. Ponadto używanie tego oprogramowania nie wymaga opłat licencyjnych.

Przygotowanie dokumentu do wydruku zawierającego informacje z bazy danych sprowadza się do sporządzenia odpowiedniego tekstu w  $\LaTeX$ -u. Ponieważ rozbudowywany jest istniejący interfejs WWW do bazy danych napisany w Meta-HTML, naturalnym wydaje się zastosowanie Meta-HTML do tworzenia dokumentów  $\LaTeX$ -owych. Wbudowane w Meta-HTML funkcje pozwalają to zadanie zrealizować.

W zależności od rodzaju wydruku, tzn. od tego czy jest to raport, czy wydruk tabeli, w generowanych dokumentach  $\LaTeX$ -owych powtarzają się większe bądź mniejsze fragmenty. Bardzo często powtarza się preambuła. Dlatego też, zamiast generować kompletny dokument za każdym razem, modyfikujemy tylko przygotowany wcześniej szablon. Zwróćmy uwagę, że każdy raport – wydruk szczegółowy – ma swój własny szablon. Natomiast w przypadku wydruku ogólnego jest jeden szablon dla wszystkich tabel. Jest on maksymalnie uproszczony i wydruk ogólny sprowadza się do przygotowania tabeli w  $\LaTeX$ -u.

Niestety  $\LaTeX$  nie jest doskonałym narzędziem do pracy z tabelami. W naszym jednak przypadku tabele są relatywnie proste, rzadko się zdarza, że zawierają one połączone komórki. Ponieważ na ogół długość jednej tabeli przekracza rozmiar papieru A4, stosujemy pakiet `longtable` [2] pozwalający składać długie tabele, przekraczające jedną stronę.

Tak więc rozbudowę aplikacji RDM należało zacząć od napisania szablonu w  $\LaTeX$ -u. W szablonie tym umieszczamy *znaczniki*, które będą wyszukiwane i zastępowane odpowiednimi fragmentami kodu w  $\LaTeX$ -u poprzez aplikację Meta-HTML. Aby wyraźnie oddzielić znaczniki od reszty tekstu  $\LaTeX$ -owego i umożliwić aplikacji niezawodną identyfikację tych znaczników przyjęto, że:

- w preambule, gdy znacznik nie jest argumentem makra, to ma postać

`%ZNACZNIK%`

- wewnątrz makra (jako argument makra) znacznik ma postać

`*ZNACZNIK*`

Okazało się, że w aplikacji Meta-HTML, potrzebna jest ochrona znaku \*, ponieważ odgrywa on specjalne znaczenie w regularnych wyrażeniach. W tym celu znak \* poprzedzamy znakiem \ w kodzie Meta-HTML.

W szablonie dokumentu ogólnego można wyróżnić następujące trzy części

- nagłówek,
- część główna,
- stopkę.

W nagłówku umieszczona jest nazwa wydziału oraz nazwa drukowanej tabeli. Natomiast w stopce znajduje się bieżąca data z godziną ułatwiająca identyfikację wydruku oraz numer strony wydruku i ilość wszystkich stron.

### 1.3 Rozbudowa modułu zarządzania

Ze względu na bezpieczeństwo bazy danych Wydziału Matematyczno-Fizycznego dostęp do zgromadzonych danych wymaga autoryzacji użytkownika. Do tej pory były cztery poziomy dostępu:

- czytanie,
- wstawianie,
- aktualizacja,
- usuwanie.

Każdemu użytkownikowi przypisana jest kombinacja tych czterech poziomów, do każdej tabeli bazy z osobna. Lista użytkowników bazy danych umieszczona jest w tabeli `USERS`, natomiast przypisane użytkownikom prawa dostępu zgromadzone są w tabeli `USERPRIV`. Są to tabele systemowe dostępne wyłącznie administratorowi.

System autoryzacji został rozbudowany o dodatkowy poziom, a mianowicie

- drukowanie.

Oznaczało to modyfikację tabeli `USERPRIV` i zmianę typu kolumny `permissions` z `set('s','u','i','d')` na `set('s','u','i','d','p')`, gdzie właśnie 'p' odpowiada przywilejowi drukowania. Zmiana ta pociąga również za sobą konieczność dostosowania tej części interfejsu, która odpowiada za nadawanie praw użytkownikom. Sprawdzenie, czy dany użytkownik posiada przywilej drukowania z danej tabeli wykonywane jest przy pomocy makra `AuthPrint`, które zostało umieszczone w bibliotece `auth.mhtml`.

W zależności od tego, czy użytkownik posiada uprawnienia do drukowania, w przeglądarce tabel bazy danych pojawia się przycisk *Drukuj* lub nie.

## 1.4 Generowanie dokumentu

Omówimy teraz szczegółowo jak wygląda tworzenie dokumentu w przypadku wydruku ogólnego:

1. Autoryzacja. W momencie kliknięcia drukuj w okienku przeglądarki rekordów (rys. 1.1) następuje załadowanie strony na której ustala się parametry wydruku (rys. 1.2). Pierwszą rzeczą przy ładowaniu tej strony jest sprawdzenie, czy zalogowany użytkownik jest uprawniony do wykonywania wydruków z danej tabeli. Jeśli takich uprawnień nie ma, wyświetlone jest ostrzeżenie i nie możliwe jest kontynuowanie wydruku. Jest to nadmiarowa kontrola, gdyż przycisk drukuj w przeglądarce rekordów jest dostępny tylko wtedy, gdy użytkownik ma prawo do dokonywania wydruku.
2. Po ustaleniu parametrów wydruku, tzn. wyborze kolumn, sposobu ustalania szerokości kolumn, formatu dokumentu, orientacji oraz układu papieru i wybraniu przycisku *OK* (por. rys. 1.2) następuje wywołanie funkcji `Execute` z biblioteki `brprint` odpowiedzialnej za tworzenie dokumentu. Na początku przesyłana jest do przeglądarki informacja o tworzeniu dokumentu. Realizuje się to przy pomocy makra *PushInfo* znajdującego się w bibliotece `print.mhtml`.
3. Przygotowanie dokumentu do wydruku rozpoczyna się od wypełnienia szablonu. Wypełnienie szablonu odbywa się w ten sposób, że do czytania otwierany jest plik szablonu. Natomiast do pisania, w tym samym czasie, otwierany jest plik tymczasowy, który w nazwie zawiera nazwę tabeli, bieżącą datę i godzinę. Jest to dokument  $\text{\LaTeX}$ -owy dlatego też posiada rozszerzenie `.tex`.

Plik szablonu czytany jest linijka po linijce. W miejsce wyróżnionych znaczników wstawiane są dane dotyczące drukowanej tabeli, takie jak:

- nazwa instytucji w miejsce `*BUSINESS*`,
- nazwa tabeli w miejsce `*TITLE*`,
- bieżąca data i godzina zamiast `*CURDATE*`.

W przypadku, gdy wybierzemy wydruk w orientacji poziomej (`landscape`), w proambule szablonu za znacznik `%LANDSCAPE%` podstawiana jest komenda  $\text{\LaTeX}$ -a `\Landscape`, która decyduje o zmianie orientacji wydruku. Gdy natomiast wybierzemy orientację pionową (`portrait`), za ten znacznik podstawiany jest pusty ciąg znaków.

Najważniejszym znacznikiem jest `*DOCBODY*`. W jego miejsce wstawiany jest wynik makra *FillBody* z biblioteki `brprint` (por. dodatek C). Odpowiada ono za generowanie tabeli. W przypadku wydruku ogólnego



Skocz do:

Szukaj:  w polu:  Autor

Sortuj wg: Autor

w wynikach  uwzględnij wielkość liter  całe słowa

rosnąco  malejąco

1 – 10 z 10

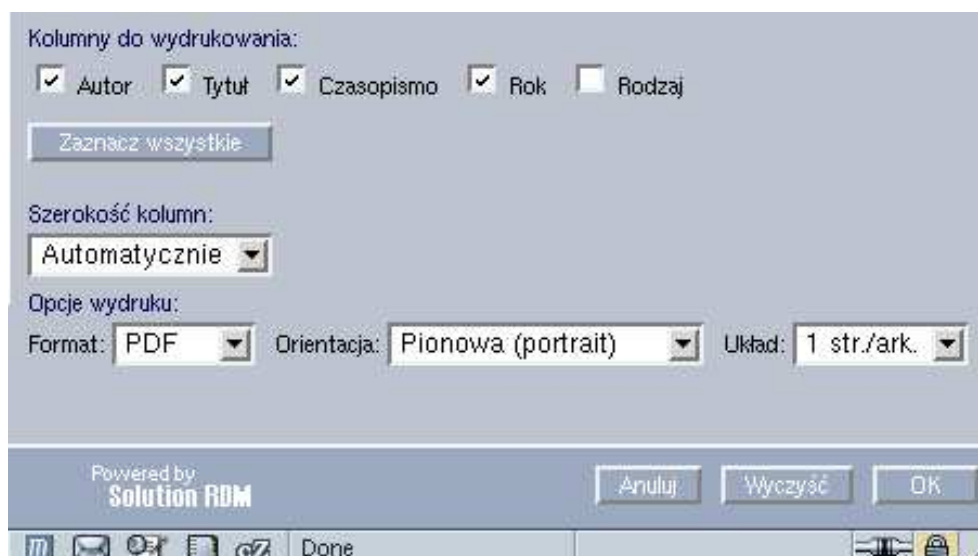
Lp.	Autor	Tytuł	Czasopismo	Rok
1.	Anna Gomolińska	A comparative study of some generalized rough approximations	Fund. Inform.	2002
2.	Anna Gomolińska	Derivability of rules from rule complexes	Logic Log. Philos.	2002
3.	Czesław Bagiński and Marcin Łuba	O klasyfikacji skończonych grup prostych	Wiadom. Mat.	2002
4.	Ezbieta Majewska	O pewnych modelach aukcji w odniesieniu do rynku papierów war...		2002
5.	Jan Krempa and Izabela Malinowska	Groups of $p$ -automorphisms for finite $p$ -groups	Publ. Math. Debrecen	2002
6.	Jurij Aminov	An expression for the volume of asymptotic parallelepiped		2002
7.	Jurij Aminov	On isometric immersions of Lobachevsky spaces into Euclidean ...		2002
8.	Krzysztof Prażmowski and Mariusz Żynel	Automorphisms of spine spaces	Abh. Math. Sem. Univ. Hamburg	2002
9.	Ryszard R. Andruszkiewicz	On iterated maximal essential extensions of rings	Algebra Colloq.	2002
10.	Ryszard R. Andruszkiewicz	Generalized iterated maximal essential extensions of rings	Algebra Colloq.	2003

Mierze: 15 Filtr: Br. i ubiegły

Powered by Solution RDM

Drukuj Wstawił Odśwież Zamknij

Rysunek 1.1: Przeglądarka tabeli.



Rysunek 1.2: Okno wyboru parametrów wydruku.

cała tabela musi być przygotowana z poziomu Meta-HTML dlatego, że zmienia się ilość kolumn w tabeli oraz ich szerokość (szczegóły algorytmu doboru szerokości kolumn przedstawione są w 1.5).

Generowanie tabeli w makrze *FillBody* odbywa się w ten sposób, że wysyłane jest do bazy MySQL zapytanie, to samo, które było użyte w okienku przeglądarki rekordów (rys. 1.1) i odbierana jest odpowiedź bazy. Odpowiedź bazy ma zawsze postać tabeli, a jej przekazanie do interpretera Meta-HTML odbywa się wiersz po wierszu. Po przeczytaniu pojedynczego wiersza danych wejściowych z bazy, generowany jest odpowiedni wiersz tabeli w  $\text{\LaTeX}$ -u, tak że wybierane są tylko te kolumny, które były zaznaczone. Proces pobierania wiersza danych i generowania wyniku jest powtarzany, aż do wyczerpania danych wejściowych z bazy.

Przeczytane z szablonu linijki tekstu, po podstawieniu za znaczniki, są sukcesywnie zapisywane w pliku tymczasowym. Tak więc plik ten jest dokładną kopią szablonu z dokładnością do wyróżnionych znaczników.

4. Utworzenie dokumentu wynikowego. W momencie, kiedy plik tymczasowy jest już uzupełniony możemy zacząć przetwarzać go dalej. Makro, które jest tutaj wykonywane to *CreateTargetDoc*. Znajduje się ono w bibliotece `print` (por. dodatek B).

W pierwszym kroku z katalogu, w którym gromadzone są dokumenty wynikowe, usuwane są te, które zostały utworzone wcześniej niż jeden dzień od bieżącej daty. Ponieważ dokumenty tworzone są dynamicznie i każde wywołanie funkcji *Drukuj* powoduje utworzenie nowego dokumentu, więc ich ilość może dość szybko rosnąć. Z drugiej strony, dokumenty

te, ze względu na swoją specyfikę dostępne są dla użytkownika tylko na krótki moment tuż po wygenerowaniu. Zatem nie warto przechowywać ich wszystkich na dysku.

Gdy docelowym formatem jest LaTeX, to w makrze *CreateTargetDoc* nie jest wykonywane nic więcej.

Jeśli jako format docelowy wybrano DVI, PS lub PDF plik tymczasowy przetwarzany jest dwukrotnie przy pomocy L<sup>A</sup>T<sub>E</sub>X-a, tak aby powstała wersja DVI dokumentu ze wstawionymi odnośnikami typu `\ref`, `\cite` oraz `LastPage`.

W przypadku, gdy docelowy format to PS lub PDF, utworzony wcześniej plik DVI przetwarzany jest przy pomocy programu `dvips`, innymi słowy na podstawie DVI generowany jest PS. Jeśli wśród opcji do drukowania znajdzie się opcja broszura (booklet) na plik PS wołany jest program `psbook`. Jeśli wśród tych opcji wystąpi opcja dwóch stron na jednym arkuszu wołany jest program `psnup` z parametrem `-2`. Ostatnim krokiem, o ile ma być utworzony dokument PDF, jest zawołanie programu `ps2pdf`. Wszystkie pliki tymczasowe, pomocnicze (`.log`, `.aux`) oraz pośrednie są usuwane, pozostaje tylko dokument docelowy. Jeśli jesteśmy w trybie "debugging" wtedy żadne pliki nie są usuwane, aby móc zdjagnozować ewentualne problemy.

5. Przygotowanie dokumentu wynikowego do pobrania przez przeglądarkę. Funkcja ta realizowana jest w makrze *OpenPDF* w bibliotece `print.mhtml`. Otwierane jest nowe okno przeglądarki, natomiast okienko dialogowe, w którym ustalane były parametry wydruku, zostaje zamknięte.

Jeżeli do wybranego typu dokumentu w przeglądarce zainstalowano odpowiednie oprogramowanie, to w tym nowym okienku zostanie otwarty oczekiwany dokument. W przeciwnym razie przeglądarka zaproponuje wybór aplikacji do obsługi tego pliku bądź zapis pliku na dysku.

## 1.5 Algorytm doboru szerokości kolumn

To czy otrzymamy czytelną i przejrzystą tabelkę zawierającą potrzebne nam informacje, zależy od tego jak dobrana zostanie szerokość poszczególnych kolumn. Nasza aplikacja przewiduje trzy możliwości jeśli chodzi o ustalanie szerokości kolumn:

**automatyczne** — szerokość kolumn wyliczana jest na podstawie ilości znaków w każdej z kolumn,

**równomierne** — szerokość strony dzielona jest na ilość kolumn tak, więc wszystkie kolumny mają tę samą szerokość,

**zadana szerokość** — szerokość kolumn zadawana jest podczas projektowania interfejsu do bazy danych.

Schemat algorytmu automatycznego podziału wygląda następująco:

1. Wyliczamy *maksymalny rozrzut*. Współczynnik ten potrzebny jest, aby były zachowane proporcje między najszerszą i najwęższą kolumną. Stosunek najszerszej do najwęższej kolumny nie powinien przekraczać maksymalnego rozrzutu. W konsekwencji jego zastosowania, poszczególne kolumny posiadają takie wymiary, że na wydruku tabela jest przejrzysta i czytelna.

Wartość współczynnika została dobrana eksperymentalnie. Przy jego wyborze kierowaliśmy się następującymi regułami:

- współczynnik nie może być stałą liczbą dla wszystkich tabel,
- powinien zależeć od ilości i szerokości kolumn,
- im więcej kolumn tym rozrzut musi być mniejszy, więc powinien dostosowywać się automatycznie.

W efekcie doszliśmy do następujących wniosków:

- dla 12 i więcej kolumn maksymalny rozrzut wynosi 3,
- w pozostałych przypadkach liczymy go następująco:

$$\frac{11 - k}{2} + 3, 5,$$

gdzie  $k$  to ilość kolumn.

Przyporządkowanie maksymalnego rozrzutu do kolumn przedstawia tabela:

$\geq 12$	3
11	3,5
10	4
9	4,5
8	5
7	5,5
6	6
5	6,5
4	7
3	7,5
2	8
1	8,5

Formalnie maksymalny rozrzut (**MaxRoz**) wyliczany jest następująco:

```

<if <le 12 <array-size list::Fields>>
  <set-var glob::MaxRoz=3.0>
  <set-var glob::MaxRoz=<add 3.5 <mul 0.5
    <sub 11 <array-size list::Fields>>>>>>

```

2. Ustalenie szerokości poszczególnych kolumn. W każdej z kolumn bierze się maksimum po długości wpisów w danej kolumnie w bazie danych.

```

<IssueQuery <get-var browser::Query>>
  <foreach Field list::Fields iter=i>
    <set-var list::Size[i]=<max list::Size[i]
      <string-length <get-var <get-var Field>>>>>
  </foreach>
</IssueQuery>

```

3. Uwzględnienie szerokości napisów w nagłówkach kolumn.

```

<foreach Field list::Fields iter=i>
  <set-var list::Size[i]=<max list::Size[i]
    <string-length
      <FindValue browser::Fields <get-var Field>>>>>
</foreach>

```

4. Zastosowanie maksymalnego rozrzutu do wyliczenia ostatecznej szerokości kolumn. Zaczynamy od ustalenia najszerszej i najwęższej kolumny.

```

<set-var Max=<apply max <subst-in-string
  <get-var list::Size[]> "\n" " ">>>
<set-var Min=<apply min <subst-in-string
  <get-var list::Size[]> "\n" " ">>>

```

Następnie wyliczamy stosunek (**Roz**) najszerszej do najwęższej i jeśli przekroczy on ustalony wcześniej maksymalny rozrzut, to modyfikowana jest odpowiednio szerokość kolumn.

```

<set-var Roz=<div <get-var Max>.0 Min>>
<when <lt glob::MaxRoz Roz>>
  <set-var Wsp=<div glob::MaxRoz Roz>>
  <foreach Field list::Fields iter=i>
    <if <lt glob::MaxRoz <div <get-var
      list::Size[i]>.0 Min>>
      <set-var list::Size[i]=<mul Min Wsp
        <div <get-var list::Size[i]>.0 Min>>>>>
  </foreach>
</when>

```

Szerokości kolumn, o których mówimy powyżej wyrażone są w znakach, przy czym ignorowana jest różnica szerokości pomiędzy znakami, w szczególności między "i" oraz "m".

## 1.6 Biblioteka brprint

Biblioteki `brprint` i `print` stanowią integralną część rozwijanej aplikacji RDM. W ich kodzie zawarte są nie tylko makra Meta-HTML, ale również makra pochodzące z bibliotek tej aplikacji. Niezbędne biblioteki podłączane są przy pomocy makra `require` (por. str. 24).

`FillBody` — makro odpowiedzialne za wygenerowanie kompletnej tabeli w  $\text{\LaTeX}$ -u. Brany jest tutaj pod uwagę wybór dokonany przez użytkownika dotyczący sposobu doboru szerokości kolumn. W przypadku doboru automatycznego aplikowany jest algorytm opisany w 1.5. Gdy natomiast wybrano podział równomierny szerokości wszystkich kolumn ustawiane są na jeden. Jeżeli wybrano zadaną szerokość to szerokości kolumn są takie jak podano w dokumencie opisującym daną tabelkę. Niezależnie od wyboru algorytmu, tablica `list::Size` zawiera szerokości poszczególnych kolumn wyrażone w znakach. Innymi słowy szerokość kolumny to długość najdłuższego wpisu w tej kolumnie.

Dalsze przetwarzanie można opisać następująco:

1. Wyliczamy sumę szerokości kolumn.

```
<set-var tmp::TSize=0>
<foreach Size list::Size>
  <set-var tmp::TSize=<add tmp::TSize Size>>
</foreach>
```

2. Ustalamy szerokość znaku w milimetrach. W tym celu dzielimy szerokość strony pomniejszonej o marginesy (czyli 180 mm lub 256 mm w zależności od wyboru orientacji strony) przez uzyskaną wcześniej sumę.

```
<set-var CSize=<div <if
  <match <get-var posted::iPrOptOri> landscape>
    256.00 180.00> <get-var tmp::TSize>>>
```

3. Ostateczną szerokość kolumny tabeli w  $\text{\LaTeX}$ -u wylicza się mnożąc wyliczoną szerokość znaku przez szerokość kolumny.
4. Wypełnienie tabeli danymi z bazy.

`Execute` — makro wykonywane po wybraniu *OK* w okienku dialogowym, gdzie ustalane są parametry wydruku (por. rys. 1.2), opisane w 1.4.

**BrowserPrint** — ponieważ każde okienko dialogowe w systemie RDM, który jest tutaj rozbudowywany, złożony jest z dwóch podokienek (tzw. frame-ów); więc muszą być przygotowane trzy strony html:

- strona macierzysta opisująca wygląd podokienek,
- dwie strony, które zostaną do tych okienek załadowane. Właśnie makro **BrowserPrint** odpowiada za stronę macierzystą i wydziela w okienku dialogowym dwie części:
  - okienko właściwe (tzw. Body),
  - stopka (tzw. Foot).

**BrowserPrintBody** — to makro odpowiada za zasadniczą część okienka dialogowego, w którym ustalane są parametry wydruku,

**BrowserPrintfoot** — to makro odpowiada za stopkę okna dialogowego i wyświetla w niej trzy przyciski:

- *Anuluj*,
- *Wyczyść*,
- *OK*.

## 1.7 Biblioteka print

W bibliotece **print**, poza wymienionymi wcześniej makrami, znajdują się jeszcze następujące makra:

**SafeTex** — to makro jako argument pobiera dowolny tekst i ochrania zawarte w nim znaki o specjalnym znaczeniu w  $\text{T}_{\text{E}}\text{X}$ -u. Znak  $\backslash$  zamieniany jest na makropolecenie  $\backslash\text{textbackslash}$ , natomiast pozostałe znaki specjalne t.j.  $\%$ ,  $\&$ ,  $\{$ ,  $\}$  są poprzedzane znakiem  $\backslash$  tak aby nie powodowały problemów w przetwarzanym dokumencie  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -owym.

**CleanWS** — to makro w dowolnym tekście ”kompresuje” spacje i znaki nowej linii.

**PrintOptions** — to makro odpowiada za elementy formularza służące do wyboru formatu, orientacji oraz układu tworzonego dokumentu.

**HTMLPage** — jest to odpowiednik makra **OpenTargetDoc** z tym, że zamiast otwierania nowego okna w celu załadowania tworzonego dokumentu wykonywane jest to w okienku bieżącym.

# Rozdział 2

## Korzystanie z aplikacji

### 2.1 Krok po kroku

Do każdej tabeli w bazie danych Wydziału Matematyczno-Fizycznego użytkownik ma dostęp przez interfejs WWW w postaci przeglądarki rekordów (rys. 1.1). Z jej poziomu można daną tabelę przeszukiwać i sortować. Do tej pory nie było możliwe łatwe przeniesienie oglądanych danych na papier, natomiast dość często potrzebne były wydruki do różnego rodzaju sprawozdań.

Przeglądarka rekordów została wzbogacona o nową funkcję, a mianowicie funkcję drukowania. Informuje o tym nowy przycisk *Drukuj* na dolnej listwie okna przeglądarki. Jeżeli użytkownik zdecyduje się na drukowanie i wybierze *Drukuj* to wyświetli się okno (rys. 1.2) zawierające informacje co i w jakiej formie chcemy uzyskać na wydruku.

1. Na samym początku są nazwy kolumn, które zmieniają się wraz z wyborem tabelki:
  - jeżeli nie potrzebujemy całej tabelki, tylko jej fragment to zaznaczamy, które kolumny chcemy wydrukować;
  - jeżeli zaś drukujemy całość to niżej znajduje się przycisk *zaznacz wszystkie*.
2. Następnie wybieramy sposób doboru szerokości kolumn, czyli:
  - automatycznie;
  - równomiernie;
  - zadana.
3. Później wybieramy format. Do wyboru mamy:
  - PDF,
  - PS,



- DVI,
  - L<sup>A</sup>T<sub>E</sub>X.
4. Do wyboru mamy również dwie możliwe orientacje strony:
- pionowa (portrait),
  - pozioma (landscape).
5. Możemy też zdecydować o tym, czy na arkuszu wydruku umieszczona będzie jedna, bądź dwie strony. Do wyboru mamy również układ broszury (booklet).
6. Na samym dole znajdują się trzy przyciski:
- *Anuluj* — zamknięcie okienka i powrót do przeglądarki,
  - *Wyczyść* — przywrócenie domyślnych wartości wszystkim polom formularza,
  - *OK* — rozpoczęcie generowania dokumentu do wydruku.

## 2.2 Przykłady

W niniejszej sekcji przedstawimy przykłady uzyskanych wydruków przy pomocy naszej aplikacji, w zależności od wyboru algorytmu doboru szerokości kolumn. Zobaczmy jak wygląda ostateczny wydruk tej samej tabeli. Na rys. 2.1 widzimy rezultat działania algorytmu automatycznego doboru szerokości kolumn. Natomiast na rys. 2.2 wszystkie kolumny poza pierwszą mają tę samą szerokość. Do jej wydruku użyto opcji "równomiernie". Kolejny rysunek 2.3 przedstawia ten sam fragment tabeli wydrukowany z opcją "zadana", czyli gdy szerokość kolumn jest z góry zadana w opisie tabeli.

Na tych przykładach doskonale widać różnice pomiędzy opcjami doboru szerokości kolumn. Wybór pozostawiony jest użytkownikowi, aby sam mógł zdecydować w jakiej postaci chce uzyskać wydruk.

L.p.	Autor	Tytuł	Czasopismo	Rok
1	Anna Gomołńska	A comparative study of some generalized rough approximations	Fund. Inform.	2002
2	Anna Gomołńska	Derivability of rules from rule complexes	Logic Log. Philos.	2002
3	Czesław Bagiński and Marcin Luba	O klasyfikacji skończonych grup prostych	Wiadom. Mat.	2002
4	Elżbieta Majewska	O pewnych modelach aukcji w odniesieniu do rynku papierów wartościowych		2002
5	Jan Krempa and Izabela Malinowska	Groups of $p$ -automorphisms for finite $p$ -groups	Publ. Math. Debrecen	2002
6	Jurij Aminov	An expression for the volume of asymptotic parallelepiped		2002
7	Jurij Aminov	On isometric immersions of Lobachevsky spaces into Euclidean spaces and correspondent nonlinear equations		2002
8	Krzysztof Piazimowski and Mariusz Żynel	Automorphisms of spine spaces	Abh. Math. Sem. Univ. Hamburg	2002
9	Ryszard R. Andruszkiewicz	On iterated maximal essential extensions of rings	Algebra Colloq.	2002
10	Ryszard R. Andruszkiewicz	Generalized iterated maximal essential extensions of rings	Algebra Colloq.	2003

Rysunek 2.1: Szerokość kolumn ustalona automatycznie.

L.p.	Autor	Tytuł	Czasopismo	Rok
1	2	3	4	5
1	Anna Gomińska	A comparative study of some generalized rough approximations	Fund. Inform.	2002
2	Anna Gomińska	Derivability of rules from rule complexes	Logic Log. Philos.	2002
3	Czesław Bagiński and Marcin Luba	O klasyfikacji skończonych grup prostych	Wiadom. Mat.	2002
4	Elżbieta Majewska	O pewnych modelach aukcji w odniesieniu do rynku papierów wartościowych		2002
5	Jan Krempa and Izabela Malinowska	Groups of $p$ -automorphisms for finite $p$ -groups	Publ. Math. Debrecen	2002
6	Jurij Aminov	An expression for the volume of asymptotic parallelepiped		2002
7	Jurij Aminov	On isometric immersions of Lobachevsky spaces into Euclidean spaces and correspondent nonlinear equations		2002
8	Krzysztof Piaznowski and Mariusz Żynel	Automorphisms of spine spaces	Abh. Math. Sem. Univ. Hamburg	2002
9	Ryszard R. Andruszkiewicz	On iterated maximal essential extensions of rings	Algebra Colloq.	2002
10	Ryszard R. Andruszkiewicz	Generalized iterated maximal essential extensions of rings	Algebra Colloq.	2003

Rysunek 2.2: Równomierna szerokość kolumn.

Lp.	Autor	Tytuł	Czasopismo	Rok
1	2	3	4	5
1	Anna Gomińska	A comparative study of some generalized rough approximations	Fund. Inform.	2002
2	Anna Gomińska	Derivability of rules from rule complexes	Logic Log. Philos.	2002
3	Czesław Baginski and Marcin Luba	O klasyfikacji skończonych grup prostych	Wiadom. Mat.	2002
4	Elżbieta Majewska	O pewnych modelach aukcji w odniesieniu do rynku papierów wartościowych		2002
5	Jan Krempa and Izabela Malinowska	Groups of $p$ -automorphisms for finite $p$ -groups	Publ. Math. Debrecen	2002
6	Jurij Aminov	An expression for the volume of asymptotic parallelepiped		2002
7	Jurij Aminov	On isometric immersions of Lobachevsky spaces into Euclidean spaces and correspondent nonlinear equations		2002
8	Krzysztof Prazmowski and Mariusz Żynel	Automorphisms of spine spaces	Abh. Math. Sem. Univ. Hamburg	2002
9	Ryszard R. Andruszkiewicz	On iterated maximal essential extensions of rings	Algebra Colloq.	2002
10	Ryszard R. Andruszkiewicz	Generalized iterated maximal essential extensions of rings	Algebra Colloq.	2003

Rysunek 2.3: Zadana szerokość kolumn.

# Dodatek A

## Szablon brprint.tex

```
%
% (c) Copyright 2003 Barbara Urynowicz & Mariusz Zynel
%
% Last modified: May 23, 2003.
%
% This software is FREE. You can use and/or redistribute it for any
% purpose in either, modified, or unmodified form, under the terms of the
% GNU General Public License as published by the Free Software Foundation.
%
% The above copyright notice and this permission notice shall be included
% in all copies or substantial portions of this software.
%
% THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
% EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
% LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

\documentclass[a4paper,10pt]{article}

\usepackage[T1]{poltex}
\usepackage{titlesec,tabularx,lastpage,ltxtable,colortbl}
\usepackage[dvips]{graphicx}

\renewcommand{\rmdefault}{phv}
\renewcommand{\mddefault}{mc}
\renewcommand{\bfdefault}{bc}

\newdimen\tmpdim
\topmargin0mm
\textheight266mm
\textwidth190mm
\marginparwidth0pt
\marginparsep0pt
\headheight12pt
\headsep8pt
\topskip0pt
\footskip\headheight
\addtolength\footskip{\headsep}
\parindent0pt

\def\landscape{
  \setlength\tmpdim{\paperheight}
  \setlength\paperheight{\paperwidth}
  \setlength\paperwidth{\tmpdim}
  \textheight190mm
  \textwidth266mm}
```



```
\def\arraystretch {1.2}  
\setlength\extrarowheight{1pt}  
  
\newcolumntype{S}{>{\columncolor{mgray}\tiny}c}  
  
\normalfont\small  
  
\section{*TITLE*}  
  
*DOCBODY*  
  
\end{document}
```

# Dodatek B

## Biblioteka print

```
;;;
;;; print.mhtml - Printing
;;;
;;; (c) Copyright 2003 Barbara Urynowicz & Mariusz Zynel
;;;
;;; Version 4.0
;;;
;;; This software is FREE. You can use and/or redistribute it for any
;;; purpose in either, modified, or unmodified form, under the terms of the
;;; GNU General Public License as published by the Free Software Foundation.
;;;
;;; The above copyright notice and this permission notice shall be included
;;; in all copies or substantial portions of this software.
;;;
;;; THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
;;; EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
;;; LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

<require stdlib>

<defun SafeTeX aText>
  <subst-in-string <get-var aText> "\\\\" "\textbackslash" "%" "\\%" "&"
    "\\&" "{" "\\{" "}" "\\}">
</defun>

<defun CleanWS aText>
  <subst-in-string <get-var aText> "[\t_]+" " " "\n[ ]*\n" "\n">
</defun>

<defsubst PrintOptions &optional aColspan>
  <Cell "Opcje wydruku" <get-var aColspan>>
    Format:&nbsp;<SelectionRaw iPrOptFmt
      "pdf==PDF\nps==PS\ndvi==DVI\ntex==LaTeX"> &nbsp;&nbsp; 
    Orientacja:&nbsp;<SelectionRaw iPrOptOri "portrait==Pionowa (portrait)
      \nlandscape==Pozioma (landscape)">&nbsp;&nbsp; 
    Układ:&nbsp;<SelectionRaw iPrOptNup "plain==1\str./ark.\npsnup==2
      str./ark.\nbookpsnup==Booklet">
  </Cell>
</defsubst>

<defun CreateTargetDoc &optional aFileBN>
  <defvar aFileBN <get-var glob::FileBN>>
  <cgi-exec "/usr/bin/find_<PathToTmp>_mtime_+1_-exec_rm_{}_\";
    "_output=glob::OutPut">

  <when <match <get-var posted::iPrOptFmt> "dvi|ps|pdf">>
    <cgi-exec "cd_/tmp;_/opt/bin/latex_<get-var_aFileBN>;_/opt/bin/latex
```



```

<get-var aFileBN>"_output=glob::OutPut>

<when <match <get-var posted::iPrOptFmt> "ps|pdf">>
  <cgi-exec "cd_/tmp;_/opt/bin/dvips_t_a4" <if <match
    <get-var posted::iPrOptOri> "landscape"> "-t_landscape">
    "<get-var _aFileBN>" output=glob::OutPut>

  <when <match <get-var posted::iPrOptNup> "book">>
    <cgi-exec "cd_/tmp;_/opt/bin/psbook<get-var _aFileBN>.ps
    <get-var aFileBN>-tmp.ps; mv <get-var aFileBN>-tmp.ps
    <get-var aFileBN>.ps" _output=glob::OutPut>
  </when>
  <when <match <get-var posted::iPrOptNup> "psnup">>
    <cgi-exec "cd_/tmp;_/opt/bin/psnup-2<get-var _aFileBN>.ps
    <get-var aFileBN>-tmp.ps;
    mv <get-var aFileBN>-tmp.ps <get-var aFileBN>.ps" _output=glob::OutPut>
  </when>

  <when <match <get-var posted::iPrOptFmt> pdf>>
    <cgi-exec "cd_/tmp;_/opt/bin/ps2pdf" "-sPAPERSIZE=a4_"
    <get-var aFileBN>.ps" _
    <get-var aFileBN>.pdf" _output=glob::OutPut>
  </when>
</when>
</when>

<cgi-exec "mv_/tmp/<get-var _aFileBN>.<get-var _posted::iPrOptFmt>
  <PathToTmp>/" _output=glob::OutPut>

<when <not <get-var posted::Debug>>>
  <cgi-exec "rm_f_/tmp/<get-var _aFileBN>.*" output=glob::OutPut>
</when>
</defun>

<defsubst PushInfo &optional &key aTitle aText>
  <defvar aTitle <get-var glob::Title>>
  <defvar aText "Tworzenie dokumentu ...">
  <server-push>
  <html>
  <head><title><get-var aTitle></title></head>
  <body bgcolor=<get-var form::ColorBody> text=<get-var form::ColorForeground>
    marginwidth=15 leftmargin=15 marginheight=15 topmargin=15>
  <h4><get-var aText></h4>
  </body>
  </html>
  </server-push>
</defsubst>

<defsubst OpenTargetDoc &optional &key aTitle aFileBN>
  <defvar aFileBN <get-var glob::FileBN>>
  <defvar aTitle <get-var glob::Title>>
  <set-var File=<get-var glob::FileBN>.<get-var posted::iPrOptFmt>>

  <script language="javascript">
    window.open("<TmpURL>/<get-var _File>", "_blank",
      "resizable=yes,scrollbars=yes");
  <when <not <get-var posted::Debug>>>
    closeDialog()
  </when>
  window.parent.FootWin.document.MainForm.bOK.value='Zamknij';

  function submitForm () {
    closeDialog()
  }

```

```
</script>

</defsubst>

<defsubst HTMLPage &optional &key aTitle aFileBN>
  <defvar aFileBN <get-var glob::FileBN>>
  <defvar aTitle <get-var glob::Title>>
  <set-var File=<get-var glob::FileBN>.<get-var posted::iPrOptFmt>>
  <html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-2">
    <meta http-equiv="refresh" content="1; URL=<TmpURL>/<get-var File>">
    <meta name="design" content="Mariusz Żynel - www.solution-4u.com">
    <title><get-var aTitle></title>
    <link rel="stylesheet" type="text/css" href="<StyleDir>/main.css">
  </head>
  <body bgcolor=<get-var form::ColorBody> text=<get-var form::ColorForeground>
    marginwidth=15 leftmargin=15 marginheight=15 topmargin=15>
  <center>
    <p class=center>
      Jeśli żądany plik nie został pobrany, kliknij:
    <br><br>
    <a href="<TmpDir>/<get-var File>">Pobierz teraz</a>
    <br><br><br><br>
    <a href="javascript:close()">Zamknij</a>
  </center>
  </body>
  </html>
</defsubst>
```

# Dodatek C

## Biblioteka brprint

```
;;;
;;; brprint.mhtml - Print browser contents
;;;
;;; (c) Copyright 2003 Barbara Urynowicz & Mariusz Zynel
;;;
;;; Version 4.0
;;;
;;; This software is FREE. You can use and/or redistribute it for any
;;; purpose in either, modified, or unmodified form, under the terms of the
;;; GNU General Public License as published by the Free Software Foundation.
;;;
;;; The above copyright notice and this permission notice shall be included
;;; in all copies or substantial portions of this software.
;;;
;;; THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
;;; EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
;;; LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

<require lists>
<require forms>
<require databases>
<require auth>
<require log>
<require print>
<require queries>

<defun BrowserID>
  <get-var glob::BrowserID>
</defun>

<defsubst FillBody>
  ;; Przekształcamy listę rozdzieloną przecinkami na tablicę
  <CommaToArray <get-var posted::iFields> list::Fields>

  <when <string-eq a <get-var posted::iWidthMethod>>
    ;; Wyliczmy maksymalny rozrzut
    <if <le 12 <array-size list::Fields>>
      <set-var glob::MaxRoz=3.0>
      <set-var glob::MaxRoz=<add 3.5 <mul 0.5 <sub 11
        <array-size list::Fields>>>>>

    ;; Policzmy maksima w poszczególnych kolumnach
    <IssueQuery <get-var browser::Query>>
    <foreach Field list::Fields iter=i>
      <set-var list::Size[i]=<max list::Size[i] <string-length
```

```

        <get-var <get-var Field>>>>
    </foreach>
    </IssueQuery>

    ;;; Uwzględnijmy szerokości napisów w nagłówkach kolumn
    <foreach Field list::Fields iter=i>
    <set-var list::Size[i]=<max list::Size[i] <string-length
        <FindValue browser::Fields <get-var Field>>>>
    </foreach>

    ;;; Uwzględnijmy ograniczenia co do ostatecznej szerokości kolumn
    ;;; Aplikujemy wyliczony maksymalny rozrzut glob::MaxRoz
    <set-var Max=<apply max <subst-in-string <get-var list::Size[]>
        "\n" "┐">>>
    <set-var Min=<apply min <subst-in-string <get-var list::Size[]>
        "\n" "┐">>>

    <set-var Roz=<div <get-var Max>.0 Min>>
    <when <lt glob::MaxRoz Roz>>
    <set-var Wsp=<div glob::MaxRoz Roz>>
        <foreach Field list::Fields iter=i>
            <if <lt glob::MaxRoz <div <get-var list::Size[i]>.0 Min>>
                <set-var list::Size[i]=<mul Min Wsp
                    <div <get-var list::Size[i]>.0 Min>>>>
            </foreach>
        </when>
    </when>

    <when <string-eq r <get-var posted::iWidthMethod>>>
        <foreach Field list::Fields iter=i>
            <set-var list::Size[i]=1>
        </foreach>
    </when>

    <when <string-eq z <get-var posted::iWidthMethod>>>
        <foreach Field list::Fields iter=i>
            <set-var list::Size[i]=<FindValue browser::FieldWidth
                <get-var Field>>>
        </foreach>
    </when>

    ;;; Policzmy sumę uzyskanych maksimów, czyli "najdłuższy" wiersz
    <set-var tmp::TSize=0>
    <foreach Size list::Size>
        <set-var tmp::TSize=<add tmp::TSize Size>>
    </foreach>

    ;;; Wyliczamy ile przypada na pojedynczy znak (w mm)
    <set-var CSize=<div <if <match <get-var posted::iPrOptOri> landscape>
        256.00 180.00> <get-var tmp::TSize>>>

    \newcolumnntype{P}{p<get-var Size>mm}

    \begin{longtable}{|r|<subst-in-string "<foreach _Size_ list::Size>p{<sub
        <mul CSize Size> 4>mm}|</foreach>"┐" [ \n]+ "┐">}
        \hline
        \rowcolor{dgray}
        \hfil L.p.
        <foreach Field list::Fields> & \hfil\bfseries <FindValue browser::Fields
            <get-var Field>></foreach>\|
        \hline
        \rowcolor{mgray}
        \multicolumn{1}{|S|}{1}
        <For i 1 <array-size list::Fields>> &
            \multicolumn{1}{|S|}{<add 1 i>}</For>\|
    \end{longtable}

```

```

\hline
\endhead
\endlastfoot
\endfoot
<set-var i=1>
<IssueQuery <get-var browser::Query>>
  <get-var ^::i>
  <foreach Field list::Fields iter=i>
    & {\raggedright <SafeTeX <get-var <get-var Field>>>}
  </foreach>\
  \hline
  <increment ^::i>
</IssueQuery>
\hline
\end{longtable}
</defsubst>

<defun Execute>
  <set-var
    glob::Title=<match <get-var db::Table> "^t" action=delete>
    glob::FileBN=<get-var _glob::Title>-<date::format-time _"MMDDhhmmi">
    glob::File=<get-var _glob::FileBN>.<get-var _posted::iPrOptFmt>>
  ;;;<set-var posted::Debug=1>

  <PushInfo>

  <DataBaseInit <get-var glob::SQLType> <get-var db::DataBase> nobody ""
    <get-var db::Table> <get-var db::PrimaryKey>>

  <with-open-stream src "<PathToTeX>/brprint.tex" type=file mode=read>
    <with-open-stream dst /tmp/<get-var glob::FileBN>.tex type=file
      mode=write-create>
    <while <stream-readable src>>
      <set-var chunk=<stream-get src>>
      <subst-in-var chunk "\\*BUSINESS\\*" <get-var glob::Business>>
      <subst-in-var chunk "\\*TITLE\\*" <get-var glob::Title>>
      <subst-in-var chunk "%LANDSCAPE%"
        <if <match <get-var posted::iPrOptOri> landscape>
          "\\landscape" "">>
      <subst-in-var chunk "\\*CURDATE\\*"
        <date::format-time "DD.MM.YYYY_r.,_godz._hh:mm">>
      <if <match <get-var chunk> "\\*DOCBODY\\*">
        <subst-in-var chunk "\\*DOCBODY\\*" <FillBody>>>
      <stream-put dst <get-var chunk>>
    </while>
  </with-open-stream>
</with-open-stream>
<CreateTargetDoc>
<OpenTargetDoc>
</defun>

<defsubst BrowserPrintBody>
  <set-var glob::BrowserID=<match <WinID> "p" action=delete>>
  <session-import <BrowserID>browser browser>
  <session-import <BrowserID>db db>
  <session-import <BrowserID>auth auth>
  <AuthPrint <get-var db::Table>>

  <set-var list::WidthMethod[] =
    "a=Automatycznie
    r=Równomiernie">

  <if <get-var browser::FieldWidth[0]> <array-append "z=Zadana">

```

```

list :: WidthMethod>>

<QueryBody "100" aWidth=550>
  <Row>
    <Cell "Kolumny do wydrukowania">
      <Hidden WinID>
      <Hidden iFields>
      <foreach Item browser::Fields>
        <CheckBoxRaw mcbiFields
          aValue="<GetName<get-var Item>>"&nbsp;<subst-in-string
            <GetValue <get-var Item>> " " "&nbsp;   ">
        </foreach>
      <p>
      <input type=button class=button name=all value=" Zaznacz wszystkie "
        onClick="for (var i=0; i<this.form.elements.length; i++)
          { var e = this.form.elements[i]; if (e.type == 'checkbox' &&
            e.name == 'mcbiFields') { e.checked = true } }">
      <br>
    </Cell>
  </Row>
  <Row>
    <Selection "Szerokość kolumn" iWidthMethod list :: WidthMethod>
  </Row>
  <Row>
    <PrintOptions>
  </Row>
</QueryBody>
</defsubst>

<defsubst BrowserPrintFoot>
  <form name=MainForm method=post action=<get-var mhtml::current-doc>>
    <Hidden hCommand "">
    <SubmitPanel "Cancel=Anuluj\nReset=Wyczyść\nOK=OK" 100%>
  </form>
</defsubst>

<defsubst BrowserPrint>
  <frameset rows="*,39" border=0 bordercolor=white>
    <frame name=DestWin src=<FramesDir>/brprint-body.mhtml?WinID=<WinID>"
      toopmargin=0 leftmargin=0 marginheight=0 marginwidth=0 scrolling=auto
      frameborder=0 norsize>
    <frame name=FootWin src=<FramesDir>/brprint-foot.mhtml" topmargin=0
      leftmargin=0 marginheight=0 marginwidth=0 scrolling=no
      frameborder=0 norsize>
  </frameset>
  Frames unsupported.
</frameset>
</defsubst>

```

# Spis literatury

- [1] Lamport L., *LATEXa document preparation system*, Addison Wesley.
- [2] Longtable – Support for tables longer than a page,  
<http://www.ctan.org/tex-archive/help/Catalogue/entries/longtable.html>.
- [3] MySQL Reference Manual,  
<http://www.mysql.com/documentation/index.html>.
- [4] The Meta-HTML Language Reference Manual,  
<http://metahtml.sourceforge.net/documentation/manual/>.