
WWW.COREEDITOR.PL

Karol Wierzchołowski

**CoreEditor - Zaawansowane środowisko do
tworzenia aplikacji internetowych oraz
redagowania dokumentów w języku \LaTeX**

Olsztyn 2008

Streszczenie

CoreEditor to uniwersalny edytor tekstu, wykorzystywany głównie przy tworzeniu serwisów internetowych, składaniu dokumentów za pomocą \LaTeX -a, czy też programowaniu w takich językach jak C#, C/C++ czy Asembler.

Praca składa się z 3.

W rozdziale pierwszym przedstawiono sylwetkę programu. Pokazano proces jego instalacji oraz typowego użytkownika. Pokazano także, jak skorzystać z niektórych niestandardowych możliwości programu.

Rozdział drugi poświęcony jest mechanizmom i funkcjom, w jakie program został wyposażony, aby wspomagać proces tworzenia stron internetowych oraz składania dokumentów w \LaTeX -u.

Rozdział trzeci zawiera informacje na temat *Interpretera CoreEditor*. Jest on jądrem całego programu pozwalającym na pełną kontrolę nad nim oraz stwarzającym możliwość dokonywania różnorodnych obliczeń matematycznych.

Spis treści

Wprowadzenie	4
1. Opis programu	6
1.1. Instalacja programu	6
1.2. Interfejs programu	7
1.3. Struktura katalogów	10
1.4. Podstawowe operacje	14
1.4.1. Nowy dokument	14
1.4.2. Otwieranie i zapisywanie dokumentów	16
1.4.3. Typ dokumentu i system kodowania	18
1.4.4. Praca z szablonami i projektami	19
1.4.5. Edycja dokumentu	22
1.4.6. Edycja dokumentów w praktyce	25
1.4.7. Sprawdzanie pisowni, Thesaurus i autokorekta	27
1.4.8. Tryb VIM	28
1.5. Ustawienia program	28
1.6. Matematyka w programie	30
1.6.1. Okno Oblicz wyrażenie	31
1.6.2. Obliczenia wewnątrz dokumentu	31
1.7. Inne ciekawe funkcje	33
2. Wspierane języki	36
2.1. Wstęp	36
2.2. Języki HTML i XHTML	37
2.3. Język CSS	41
2.4. Język JavaScript	43
2.5. Język PHP	44

2.6. Język \LaTeX	46
3. Interpreter CoreEditor	48
3.1. Przeznaczenie interpretera	48
3.2. Wprowadzanie instrukcji	49
3.3. Liczby zespolone	50
3.4. Macierze	50
3.5. Operatory arytmetyczne	51
3.6. Operatory logiczne i porównania	52
3.7. Zmienne	53
3.8. Wbudowane funkcje matematyczne	54
3.9. Definiowanie funkcji	55
3.10. Pochodne, całki, długości łuków i miejsca zerowe funkcji .	56
3.11. Wbudowane funkcje systemowe	57
3.12. Bazy danych a interpreter CE	59
3.13. Dodatki	61
3.13.1. Rysowanie wykresów	61
3.13.2. Fraktale	64
Zakończenie	68
Bibliografia	69
Spis tabel	73
Spis rysunków	74

Wprowadzenie

CoreEditor to uniwersalny edytor tekstu, kierowany głównie do programistów, webmasterów oraz osób składających dokumenty przy pomocy L^AT_EX-a. Jest następcą dość popularnego kiedyś programu EdHTML, stworzonego przez tego samego autora. Można wręcz stwierdzić, że pomimo iż został napisany zupełnie od nowa, jest pewną kontynuacją myśli poprzednika.

Prace nad edytorem EdHTML rozpoczęły się już w 2001 roku. Ostatnia wersja tego programu została opatrzona numerem 5.01 i była wydana 20.03.2003 roku¹. Coraz liczniejsza konkurencja jaka się wówczas zaczęła pojawiać wśród narzędzi tego typu, jak również pewne ograniczenia, jakie stwarzała struktura programu spowodowały, że prace nad nim zostały przerwane, a autor opracował, zaprojektował i zaprogramował program zupełnie od nowa.

Pierwsza wersja programu CoreEditor ujrzała światło dzienne w 2006 roku. W porównaniu do poprzednika, wraz z całkowicie na nowo napisanym silnikiem, zmieniła się również nazwa programu, jego oprawa graficzna jak również licencja. Od tej pory CoreEditor przestał być oprogramowaniem darmowym, a korzystanie z niego wiązało się z zakupieniem odpowiedniej licencji. Dystrybucją programu zajmuje się firma MBBS sp. z o.o. (wcześniej Cyberbajt sp. z o.o.).

Obecnie najnowsza wersja programu nosi numer 1.4. Program zdobył dość duże uznanie wśród swoich użytkowników. Licencje edukacyjne nabyło wiele uczelni, szkół i innych instytucji naukowych. O programie pisano również w prasie, np. PC WORLD Computer², Magazyn INTERNET³, Komputer Świat Ekspert czy Komputer Świat. W numerze 3/2006

¹Na podstawie systemu pomocy programu EdHTML

²Konkurencja dla Pajęczka - <http://www.pcworld.pl/news/91373.html>

³Numer 4/2006 oraz http://serwis.magazynyinternetowe.pl/arttykul/187,1,151,coreeditor_1_0.html

czasopisma Komputer Świat Ekspert program został umieszczony na płycie CD wraz z darmową licencją do użytku niekomercyjnego. Z promocji tej skorzystało blisko 4 tysiące osób⁴.

Program stał się na tyle popularny i uznany, że doczekał się własnej strony w Wikipedii⁵, oraz otrzymał rekomendacje serwisu Algorytmy.pl⁶, redakcji serwisu WebInside.pl⁷ oraz DobreProgramy.pl⁸.

⁴Dane pochodzą z panelu administracyjnego zarządzającego sklepem CoreEditor

⁵<http://pl.wikipedia.org/wiki/CoreEditor>

⁶System pomocy programu CoreEditor

⁷<http://www.webinside.pl/warsztat/55>

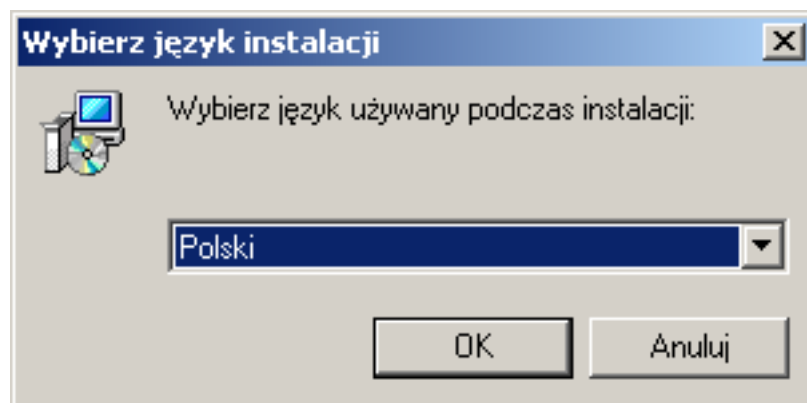
⁸<http://dobreprogramy.pl/index.php?dz=15&n=3520&Promujemy+polskich+autorow!++CoreEditor>

Opis programu

W rozdziale przedstawione zostały informacje na temat instalacji programu, jak również sposobu jego użytkowania. Omówiony został interfejs programu, struktura katalogów oraz podstawowe operacje możliwe do wykonania przy jego pomocy. Na koniec przedstawiono różne mechanizmy w jakie program został wyposażony, aby ułatwić tworzenie stron internetowych oraz redagowanie dokumentów w języku \LaTeX .

1.1. Instalacja programu

Najnowszą wersję programu można zawsze uzyskać na jego stronie domowej pod adresem <http://www.coreeditor.pl>. Po pobraniu aplikacji można uruchomić instalator w sposób standardowy dla aplikacji w systemie Windows - dwukrotnie klikając na nim lewym przyciskiem myszy. Ukaże się wówczas okno jak na rysunku 1.1.

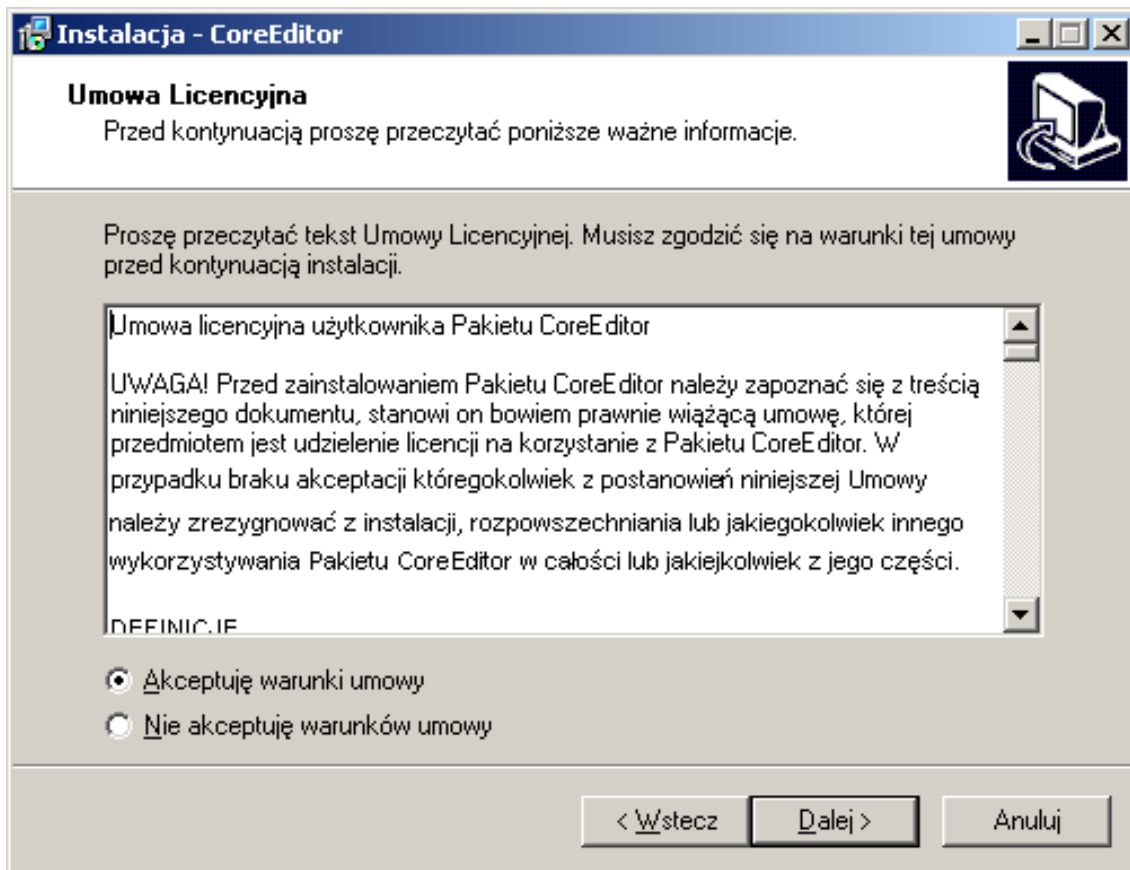


Rysunek 1.1. Instalator - Wybór języka

Należy w tym miejscu wybrać wersję językową, jaka nas interesuje¹ oraz kliknąć przycisk **OK**. Zobaczmy okno powitalne, w którym klikamy

¹CoreEditor dostępny jest w dwóch wersjach językowych: polskiej i angielskiej.

przycisk **Dalej**. Ukaże się wówczas okno z licencją programu. Należy się z nią zapoznać, a następnie zaakceptować i kontynuować. Etap ten prezentuje rysunek 1.2.



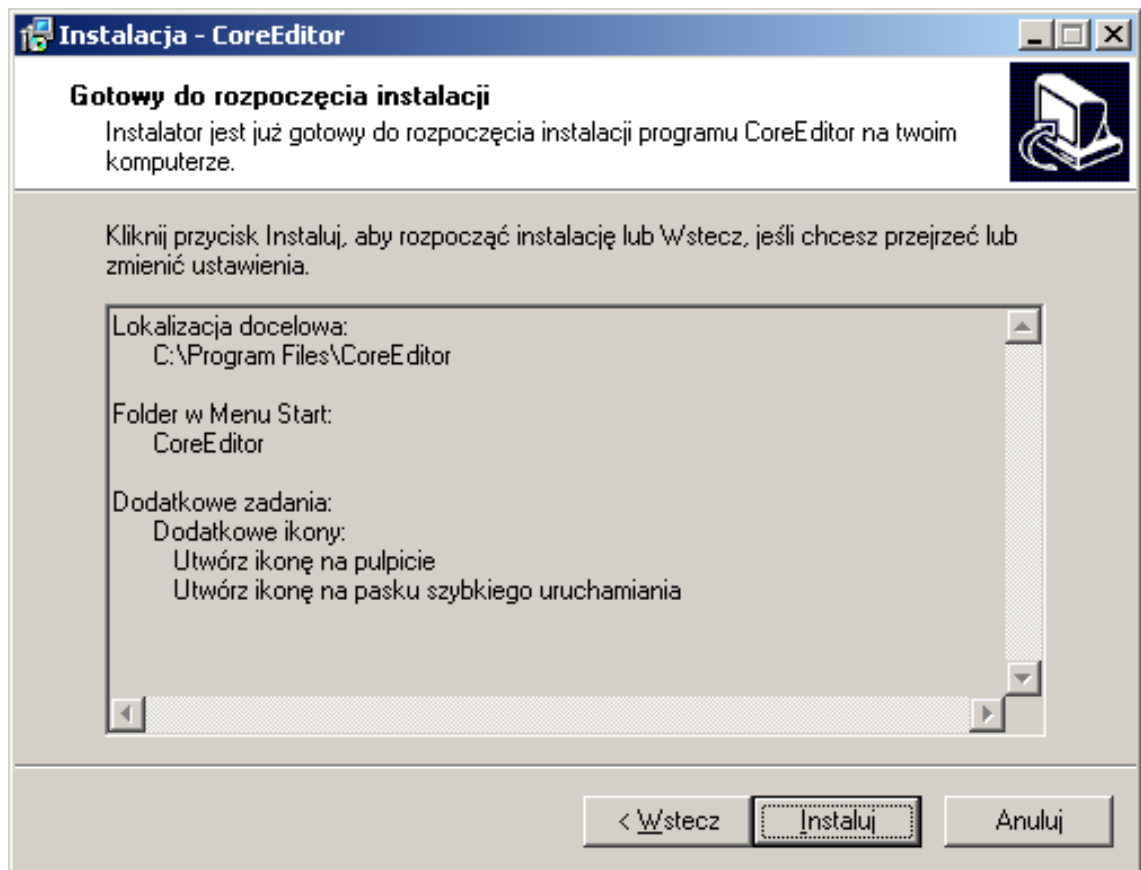
Rysunek 1.2. Instalator - Akceptacja warunków licencji

Kolejny krok to wybór katalogu, w którym program ma zostać zainstalowany, nazwa grupy tworzonej na pasku Start oraz wybór miejsc, w których mają być tworzone skróty do programu. Po tych czynnościach widoczne będzie okno jak na rysunku 1.3. Należy kliknąć przycisk **Instaluj** - aby rozpocząć proces instalacji.

1.2. Interfejs programu

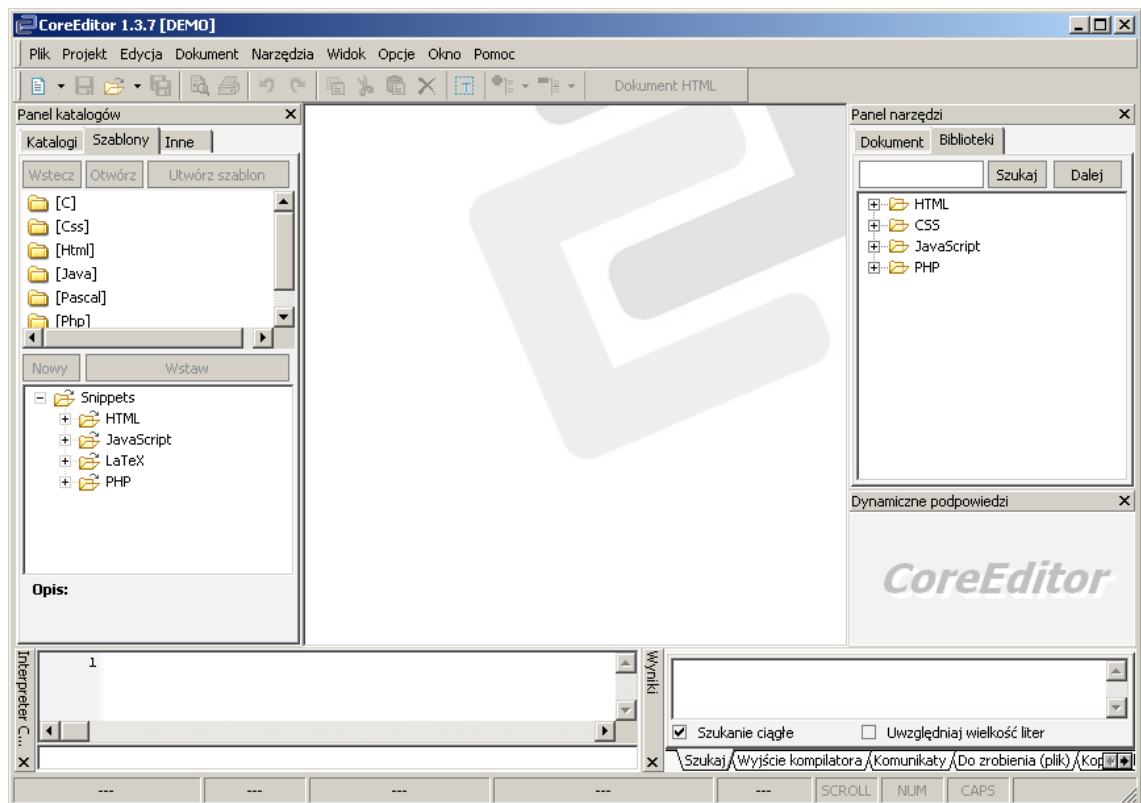
Po zainstalowaniu programu i jego uruchomieniu naszym oczom ukaże się ekran - jak na rysunku 1.4.

Interfejs CoreEditor-a składa się z następujących elementów:



Rysunek 1.3. Instalator - Podsumowanie instalacji

- **Menu** - znajdujące się na samej górze okna programu - zawiera elementy, poprzez które można sterować pracą programu, np. wykonywać operacje zapisywania, otwierania czy drukowania dokumentów, wstawiania gotowych kodów do dokumentu, zarządzania oknami, systemem pomocy itp.
- **Paski narzędzi** - widoczne poniżej menu - zawierają skróty do najczęściej wykorzystywanych elementów z menu.
- **Panel katalogów** - znajdujący się po lewej stronie okna programu - zawiera zakładki, w których umieszczone są informacje związane ze strukturą plików i katalogów na dysku, strukturą projektów, szablonami, gotowymi wstawkami (ang. *snippets*) itp.
- **Panel narzędzi** - widoczny po prawej stronie okna programu - zawiera różne dodatkowe narzędzia wspomagające i ułatwiające edycję



Rysunek 1.4. CoreEditor - Interfejs programu

dokumentów, np. dostęp do dokumentacji związanej z aktualnie używanym językiem programowania, dostęp do Inspektora znaczników i sekcji z bloków kodu CSS (ang. *Cascading Style Sheets*), strukturę dokumentów, okno dynamicznych podpowiedzi itp.

- **Interpreter CoreEditor** - widoczny w lewym dolnym rogu okna programu - zapewnia interfejs do rdzenia programu CoreEditor, a więc do interpretera, który zarządza całym programem.
- **Wyniki** - okno widoczne w prawym dolnym rogu programu - składa się z kilku zakładek prezentujących informacje różnego typu np.: zadania do wykonania z danym projektem lub plikiem, dostęp do listy kopii aktualnie edytowanego dokumentu, przechwycone komunikaty z zewnętrznych kompilatorów itp.
- **Pasek statusu** - widoczny na samym dole okna programu - zawiera różne informacje związane ze stanem CoreEditor-a np.: położenie

kursora wewnątrz dokumentu, typ dokumentu na jakim aktualnie pracujemy, sposób kodowania polskich znaków itp.

- **Część robocza** - znajduje się w centrum okna programu - tworzone są w nim okna edytowanych dokumentów, przeglądanych plików graficznych lub filmowych itp.

Położenie wszystkich tych elementów może być zmieniane przez użytkownika metodą *przeciągnij-i-upuść* (ang. *drag-and-drop*). Również wielkość tych okien jak i ich widoczność może być dowolnie modyfikowana. CoreEditor obsługuje dowolnie wiele profili konfiguracyjnych, w których zapamiętywane jest położenie, wielkość i widoczność okien. Zarządzanie profilami odbywa się za pomocą menu **Opcje**→**Profile**.

Program CoreEditor potrafi obsługiwać kilkadziesiąt różnych typów dokumentów, z którymi związane są różne menu i paski narzędzi. Dlatego zastosowano w nim mechanizm *auto-ukrywania*, który polega na tym, że w zależności od edytowanego dokumentu pojawiać się mogą dodatkowe elementy menu jak również nowe paski narzędzi.

1.3. Struktura katalogów

Domyślnym katalogiem, w którym instaluje się program CoreEditor jest ścieżka **c:\Program Files\CoreEditor**. W ramach tego katalogu tworzona jest dość rozbudowana struktura plików, w których program przechowuje różne informacje. Znaczenie tych katalogów i plików jest następujące:

Tabela 1.1: CoreEditor - Wspierane typy plików

<i>Numer</i>	<i>Nazwa</i>
0	Dokument <i>normalny</i> ;
1	Dokument HTML;
2	Dokument WML;

- 3 Dokument WMLScript;
- 4 Dokument CSS;
- 5 Dokument JavaScript;
- 6 Dokument VBScript;
- 7 Dokument PHP;
- 8 Dokument PERL;
- 9 Dokument SQL;
- 10 Dokument XML;
- 11 Dokument C/C++;
- 12 Dokument Pascal;
- 13 Dokument Asembler;
- 14 Dokument Visual Basic;
- 15 Dokument Fortran;
- 16 Dokument Java;
- 17 Dokument Python;
- 18 Dokument AWKScript;
- 19 Dokument TCL/TK;
- 20 Dokument T_EX;
- 21 Dokument Progress;
- 22 Dokument Cobol;
- 23 Dokument Foxpro;
- 24 Dokument BAT;
- 25 Dokument DFM;
- 26 Dokument INI;
- 27 Dokument internetowy (XHTML/CSS/JS/PHP);
- 28 Dokument Unix Shell;
- 29 Dokument C#;
- 30 Dokument Nemerle;
- 31 Dokument XSL;

- **CoreEditor.exe** - plik główny aplikacji CoreEditor;
- **CEUpdate.exe** - plik główny aplikacji CE Update zapewniającej pobieranie z Internetu aktualizacji do programu;
- **Licencja.txt** i **License.txt** - pliki z treścią licencji programu CoreEditor zarówno w języku polskim jak i angielskim.
- **lib*.dll** - pliki DLL zapewniające dostęp do różnego rodzaju baz danych programowi CoreEditor, np. MySQL-a czy PostgreSQL-a.
- **ssleay32.dll** i **libeay.dll** - biblioteki zapewniające możliwość szyfrowanego połączenia SSL, wykorzystywanego w momencie wysyłania raportów o występujących błędach w programie.
- **ceupdate.ini** - zawiera zapisany stan rejestracji programu; Plik zakodowany jest w ten sposób, że przeniesienie go na inny komputer sprawi, iż CoreEditor wykryje nieprawidłowość klucza licencyjnego.
- **example** - katalog zawierający różne przykłady skryptów pisanych w języku CE lub CEPascal.
- **plugins** - katalog zawierający różne biblioteki DLL rozszerzające możliwości programu, np. dodające nowe kreatory, obsługę plików dźwiękowych, analizator plików INI itp.
- **data** - katalogów, w którym umieszczone są wszystkie bazy programu CoreEditor, cała jego konfiguracja, system pomocy, kopie bezpieczeństwa, wersje językowe, szablony itp.
- **data\accounts** - zawiera opis kont FTP dodanych w programie. Hasła zapisywane w pliku są szyfrowane;
- **data\backup** - zawiera kopie bezpieczeństwa edytowanych dokumentów;

- **data\charset** - zawiera definicje różnych systemów kodowania polskich znaków²;
- **data\convert** - zawiera definicje do różnych wbudowanych systemów konwersji;
- **data\help** - zawiera system pomocy do programu oraz różne kursy i poradniki do wspieranych języków programowania;
- **data\highlighters** - zawiera definicje różnych systemów kolorowania składni, dla blisko 80. różnych języków; Pliki zapisane są w formacie XML;
- **data\lang** - zawiera różne informacje językowe, jak np. tłumaczenie interfejsu, słowniki, wersje językowe systemów pomocy itp.;
- **data\macros** - zawiera pliki z zapisanymi markami użytkownika;
- **data\menu** - zawiera własne definicje nowych elementów menu; Pliki zapisane są w formacie INI.;
- **data\pallets** - zawiera definicje różnych palet kolorów, np. *palety bezpiecznych kolorów* (ang. *safe colors*);
- **data\profile** - zawiera opis różnych profili konfiguracyjnych;
- **data\projects** - zawiera opis różnych standardowych projektów;
- **data\scripts** - zawiera kody pomocnych skryptów w językach JavaScript i PHP;
- **data\sfiles** - zawiera szereg podkatalogów numerowanych liczbami całkowitymi; Każda liczba całkowita odpowiada określonemu typowi plików, a katalog o tej nazwie, zawiera różne informacje związane z obsługą przez program CoreEditor danego typu pliku, np. informacje

²W wersji podstawowej CoreEditor obsługuje systemy ISO-8859-2, Windows 1250, Mazowie oraz UTF-8 i UTF-16

do systemu podpowiedzi, systemu autokorekty itp. Numery oraz nazwy wspieranych typów plików zebrane zostały w tabeli 1.1.

- **data\snippets** - różne wstawki kodu (ang. *snippets*);
- **data\templates** - szablony dokumentów;
- **data\sfiles.ini** - jeden z głównych plików konfiguracyjnych programu; Zapisany jest w formacie INI. Informacje w nim zawarte określają, jakie formaty plików są wspierane przez program CoreEditor. To w tym pliku przypisane są określonym typom liczby, które wskazują katalogi z informacjami szczegółowymi. Poprzez edycję tego pliku można rozszerzać możliwości programu o obsługę nowych języków programowania, języków skryptowych itp.

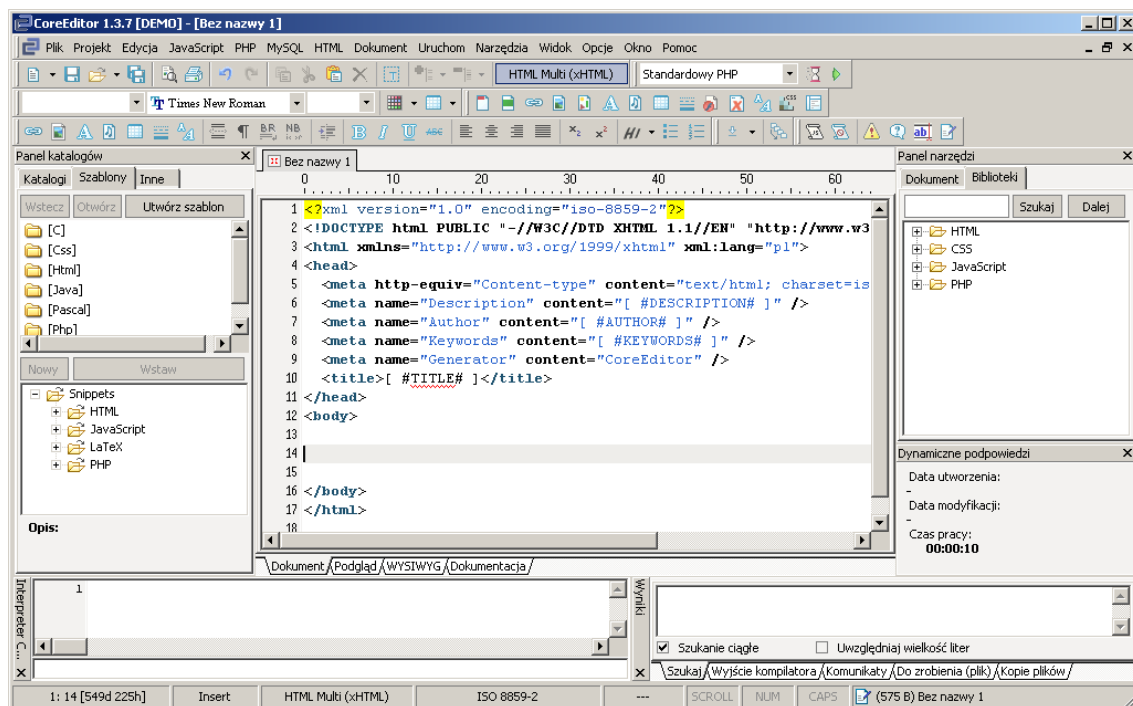
1.4. Podstawowe operacje

Posługiwanie się edytorem CoreEditor nie powinno nikomu sprawić większych trudności. Jest ono bowiem bardzo intuicyjne i podobne do innych edytorów, jak chociażby programów serii Microsoft Word.

1.4.1. Nowy dokument

Chcąc rozpocząć pracę z nowym dokumentem należy z menu **Plik** wybrać **Nowy**→**Dokument** lub wcisnąć kombinację klawiszy CTRL+N lub też wybrać pierwszą ikonkę z głównego paska narzędzi. Po każdej z tych czynności w obszarze roboczym ukaże się okno nowego, pustego dokumentu. Częściej jednak, zamiast tworzenia pustego dokumentu, wykorzystuje się w programie CoreEditor gotowe szablony, które oprócz wstawienia często typowej struktury do tworzonego dokumentu, ustawiają również środowisko pracy CoreEditor do danego języka. Na przykład, planując utworzyć nową stronę internetową w języku XHTML należy wybrać

z menu **Plik** opcję **Nowy**→**Dokument XHTML**. Wówczas okno naszego programu wygląda jak na rysunku 1.5.



Rysunek 1.5. CoreEditor - Nowy dokument XHTML

Należy zwrócić uwagę, że wraz z pojawieniem się okna nowego dokumentu, pojawiają się także nowe paski narzędzi oraz nowe elementy w menu. Jak wspomniano już wcześniej, w programie zastosowano mechanizm *auto-ukrywania* i dzięki niemu w zależności od typu edytowanego dokumentu pokazują się i ukrywają różne elementy.

W programie CoreEditor można pracować na wielu dokumentach jednocześnie. Domyślnie włączony jest tzw. *tryb zakładkowy*, co oznacza, że powyżej okna nowego dokumentu widoczne są zakładki, które pozwalają na przełączanie się pomiędzy dokumentami. Jednakże minimalizując okno dokumentu lub też wybierając jedną z opcji menu **Okno**, można zmienić sposób wyświetlania okien na *tryb kaskadowy*, tzn. pozwalający na widok kilku okien jednocześnie.

W *trybie zakładkowym*, na zakładkach napisana jest nazwa danego pliku. Obok nazwy znajduje się ikona, która pozwala na zamknięcie danej

zakładki (dokumentu). Jej kolor wskazuje również na stan danego dokumentu i jeśli jest czerwony, oznacza to, że dokument uległ modyfikacji.

1.4.2. Otwieranie i zapisywanie dokumentów

Aby otworzyć zapisany wcześniej dokument należy z menu **Plik** wybrać jedną z opcji:

- **Otwórz** - jeśli chcemy otworzyć dokument w trybie edycji tekstowej,
- **Otwórz binarnie** - jeśli chcemy otworzyć dokument w edytorze binarnym, wyświetlającym kody ASCII (ang. *American Standard Code for Information Interchange*) w systemie szesnastkowym,
- **Ostatnio otwarte** - jeśli chcemy otworzyć plik, na którym ostatnio pracowaliśmy.

Domyślnie w programie CoreEditor ustawiona jest opcja która sprawia, że jeśli program zostanie zamknięty i w momencie jego zamykania otwarte są jakiegokolwiek dokumenty, to po ponownym uruchomieniu programu dokumenty te zostaną automatycznie wczytane.

Dokumenty można otwierać również korzystając z trzeciej ikonki na głównym pasku narzędzi.

Chcąc zapisać edytowany dokument można posłużyć się jedną z opcji menu **Plik**:

- **Zapisz** - powoduje zapisanie dokumentu,
- **Zapisz jako** - powoduje zapisanie dokumentu pod nową nazwą,
- **Zapisz wszystkie** - powoduje zapisanie wszystkich otwartych dokumentów.

Możemy również posłużyć się kombinacją klawiszy CTRL+S lub drugą ikoną na pasku narzędzi.

W tabeli 1.2 zebrano opisy jeszcze innych niż omówione powyżej elementów menu **Plik**.

Tabela 1.2: CoreEditor - Menu Plik

<i>Nazwa</i>	<i>Opis</i>
Nowy	Tworzy nowy dokument;
Otwórz	Otwiera zapisany dokument;
Otwórz binarnie	Otwiera dokument przy pomocy wbudowanego edytora binarnego;
Ostatnio otwarte	Lista ostatnio otwartych dokumentów;
Otwórz z sieci Web	Pobiera dokument z sieci Web oraz otwiera go w trybie edycji;
Otwórz z serwera FTP	Otwiera dokument ze zdalnego serwera FTP;
Zapisz na serwerze FTP	Zapisuje dokument na zdalnym serwerze FTP;
Zapisz	Zapisuje dokument;
Zapisz jako	Zapisuje dokument pod nową nazwą;
Zapisz wszystkie	Zapisuje wszystkich otwarte dokumenty;
Zamknij	Zamyka dokument;
Zamknij wszystkie	Zamyka wszystkie otwarte dokumenty;
Eksportuj do	Eksportuje dokument do różnych typów, np. do pliku HTML czy T _E X;
Ustawienia drukarki	Ustawianie konfiguracji drukarki;
Ustawienia strony	Ustawienie parametrów strony wydruku, np. marginesów;
Podgląd wydruku	Wyświetla podgląd wydruku;
Drukuj	Drukuje dokument;

Wyślij dokument	Wysyła dokument na wybrany adres e-mail;
Właściwości pliku	Wyświetla różne informacje o dokumencie np.: data i czas otwarcia, liczba słów, zdań itp.;
Minimalizuj	Minimalizuje program CoreEditor do systemowego schowka;
Uruchom kopie	Uruchamia kopię programu CoreEditor;
Koniec	Kończy pracę z programem CoreEditor

1.4.3. Typ dokumentu i system kodowania

Jak wspomniano już wcześniej, w programie CoreEditor rozróżnia się różne typy plików. W zależności od danego typu, program zmienia schemat kolorowania składni, wyświetla dodatkowe elementy, uruchamia różne systemy ułatwień. Tworząc nowy dokument na podstawie szablonu CoreEditor z reguły od razu ustawia właściwy typ dokumentu. Podobnie rzecz się dzieje w momencie otwierania jakiegoś pliku. Czasami może się jednak zdarzyć, że program niepoprawnie rozpozna typ pliku lub też chcemy ręcznie zmienić sposób traktowania wczytanego dokumentu na inny. Może tak się zdarzyć na przykład w sytuacjach, kiedy w jednym dokumencie znajdują się fragmenty zapisane w różnych językach.

Na pasku narzędzi widoczne jest pole **Typ pliku**, w którym wyświetla się nazwa aktualnego typu. Klikając to pole, wyświetlimy menu z listą wszystkich wspieranych typów. W ten sposób można zmienić tryb pracy CoreEditor-a.

Typ pliku można również zmienić wybierając z menu **Dokument** opcję **Rodzaj dokumentu**.

Program CoreEditor podczas zapisywania plików zapamiętuje również w swojej własnej bazie danych nazwę zastosowanego typu. Dzięki temu przy ponownym wczytywaniu danego dokumentu przywróci właściwe ustawienia.

Podobnie jak ważną sprawą jest ustawienie poprawnego typu pliku, tak jeszcze istotniejszą sprawą jest ustawienie odpowiedniego systemu kodowania znaków. CoreEditor obsługuje kilka najpopularniejszych systemów kodowania znaków, posiada również wbudowane systemy wykrywające rodzaje kodowania i konwertujące znaki pomiędzy różnymi systemami. Czasami jednak może się zdarzyć, że kodowanie zostanie rozpoznane błędnie (na przykład w sytuacji, kiedy w dokumencie znajdują się znaki charakterystyczne dla różnych systemów kodowania znaków) lub sami chcemy ręcznie zmienić rodzaj kodowania.

Aktualnie stosowany rodzaj kodowania znaków wyświetla się zawsze w pasku statusu. Aby zmienić ten typ na inny, należy posłużyć się menu **Dokument** → **Rodzaj kodowania**.

Należy zwrócić szczególną uwagę na to, aby rodzaj kodowania znaków odpowiadał deklaracji `<meta>` w przypadku dokumentów zapisanych w językach HTML i XHTML.

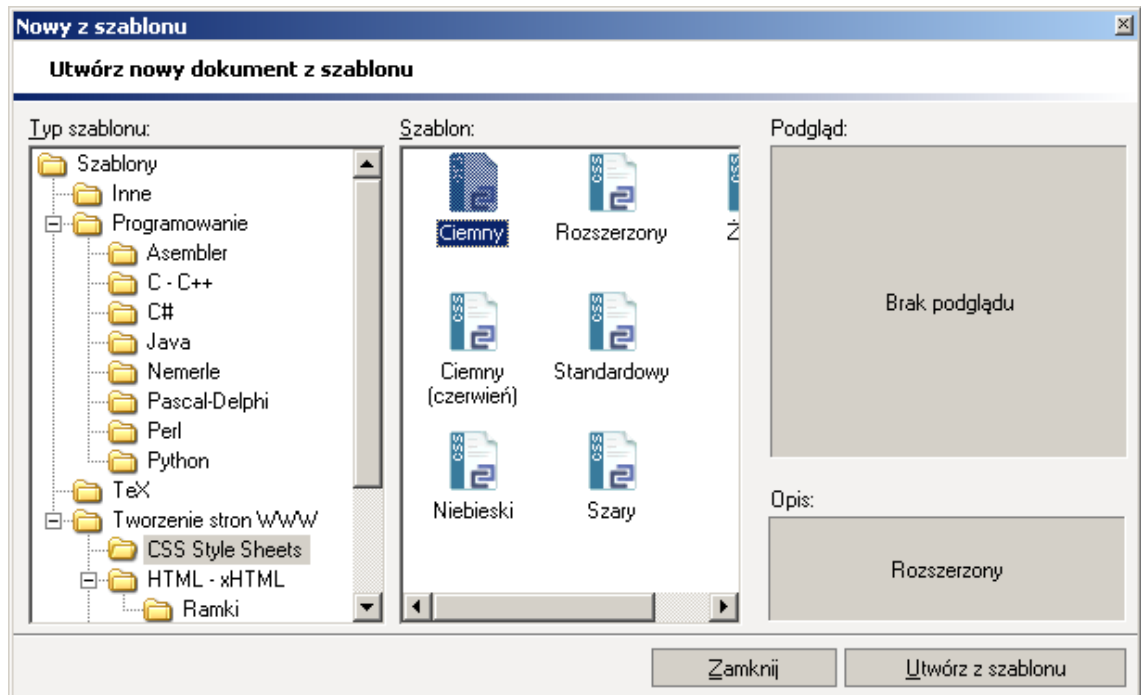
1.4.4. Praca z szablonami i projektami

CoreEditor posiada wbudowane gotowe szablony różnych typów dokumentów. Aby z nich skorzystać, należy z menu **Plik** wybrać **Nowy** → **Z Szablonu**. Ukaże się wówczas okno jak na rysunku 1.6.

Mamy tutaj dostępne szablony do wielu dokumentów, np. kanałów informacyjnych, dokumentów XHTML, XML, CSS, dokumentów języka \TeX , szablonów programów zapisanych w takich językach programowania jak Pascal, Asembler, C#, Java, Nemerle i inne.

Po wybraniu elementu, który chcemy wykorzystać wystarczy kliknąć

na przycisk **Utwórz z szablonu**, a CoreEditor utworzy nowy dokument posługując się wybranym szablonem.



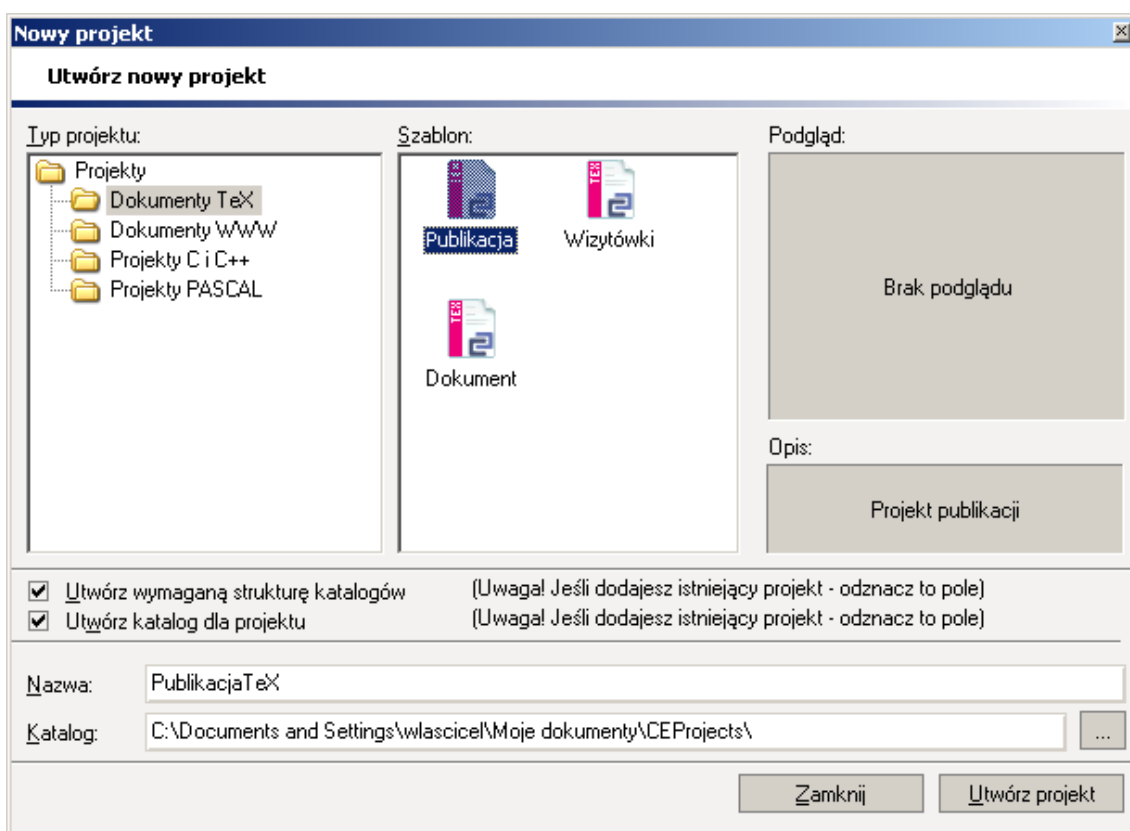
Rysunek 1.6. CoreEditor - Szablony dokumentów

CoreEditor pozwala także na pracę z projektami. Z punktu widzenia środowiska CoreEditor projekt, to zespół powiązanych ze sobą plików. Program wspiera prace na projektach pozwalając m.in. na:

- tworzenie automatycznej kopii bezpieczeństwa całego projektu;
- przechowywanie informacji o autorze projektu, postępach pracy nad nim oraz innych uwagach z nim związanych;
- generowanie historii pracy nad projektem, prowadzenie bazy zadań do wykonania;
- automatyczne otwieranie określonych plików, wskazanie pliku głównego, od którego ma zawsze następować kompilacja projektu;
- synchronizacja projektu z zewnętrznym serwerem FTP. CoreEditor kontroluje, które pliki zostały zmodyfikowane i tylko takie wysyła na zdalny serwer;

Zarządzanie projektami odbywa się poprzez menu **Projekt**. Znaczenie poszczególnych elementów tego menu zebrano w tabeli 1.3.

Chcąc utworzyć nowy projekt, należy z menu **Projekt** wybrać opcję **Nowy projekt**. Ukaże się wówczas okno jak na rysunku 1.7. Po wybraniu szablonu projektu, należy kliknąć przycisk **Utwórz projekt**. CoreEditor utworzy wówczas nowy projekt, stworzy dla niego odpowiednią strukturę katalogów oraz skopiuje do tej struktury odpowiednie pliki związane z projektem.



Rysunek 1.7. CoreEditor - Tworzenie projektu - projekt publikacji w \LaTeX

Tabela 1.3: CoreEditor - Menu Projekt

<i>Nazwa</i>	<i>Opis</i>
Nowy projekt	Tworzy nowy projekt;
Otwórz projekt	Otwiera zapisany projekt;
Zapisz projekt	Zapisuje plik projektu na dysku;

Zamknij projekt	Zamyka projekt oraz wszystkie związane z nim dokumenty;
Dodaj plik do projektu	Dodaje aktywny dokument do projektu;
Usuń plik z projektu	Usuwa aktywny dokument z projektu;
Utwórz kopie bezpieczeństwa	Tworzy kopię zapasową projektu i wszystkich plików wchodzących w jego skład;
Zamknij wszystkie	Zamyka wszystkie otwarte dokumenty;
Właściwości	Wyświetla okno właściwości projektu;

Podczas pracy z projektem w **Panelu katalogów** pokazuje się dodatkowa zakładka **Projekt** zapewniająca dostęp do wszystkich plików składowych projektu.

1.4.5. Edycja dokumentu

CoreEditor posiada szereg właściwości znacznie usprawniających edycję dowolnego typu dokumentu. Niektóre z tych funkcji są typowe dla wielu innych edytorów podobnego rodzaju. Są jednak również i takie elementy, które odróżniają program CoreEditor od innych i sprawiają, że po przyzwyczajeniu się to do nich, trudno wrócić do poprzedniego edytora.

Większość z funkcji związanych z edycją dokumentu zebrana została w menu **Edycja** a ich opis prezentuje tabela 1.4. Również w menu kontekstowym, widocznym w momencie kliknięcia prawym klawiszem myszy w obszarze edytora³, znajduje się szereg dodatkowych opcji. Niektóre z nich to:

- **Obliczenia** - menu zawiera dodatkowe opcje związane z wykonywaniem różnych obliczeń na danych zapisanych w dokumencie.

³Niektóre z elementów tego menu pojawiają się w specyficznych sytuacjach, np. po zaznaczeniu jakiegoś tekstu lub po umieszczeniu kursora w obrębie wyrazu itp.

- **Zaznacz** - menu zawiera opcje pozwalające na zaznaczanie różnych obszarów tekstu, np. wyrazu, zdania, bloku znacznika itp.
- **Wstaw** - menu zawiera opcje pozwalające na wstawianie do dokumentu różnych elementów, np. symboli specjalnych, daty i godziny, czy unikalnego numeru GUID.
- **Schowki** - menu pozwalające na kontrolę dodatkowymi schowkami⁴, w jakie wyposażony został program CoreEditor.
- **Zakładki** - menu zawiera opcje pozwalające na zarządzanie zakładkami.
- **Zaznaczanie linii** - menu zawiera opcje pozwalające na zaznaczanie wybranych linii dowolnym kolorem.

Tabela 1.4: CoreEditor - Menu Edycja

<i>Nazwa</i>	<i>Opis</i>
Cofnij	Cofa ostatnio wykonaną operację;
Powtórz	Powtarza ostatnio cofniętą operację;
Kopiuj	Kopiuje zaznaczony tekst do schowka;
Wytnij	Wycina zaznaczony tekst do schowka;
Wklej	Wkleja do dokumentu tekst przechowywany w schowku;
Wklej specjalnie	Opcja związana z dokumentami RTF pozwala na wklejanie zawartości schowka w specjalny sposób, tzn. stosując np. odpowiednią konwersję znaków, wklejając obrazek itp.;
Zaznacz wszystko	Zaznacza całą zawartość dokumentu;

⁴CoreEditor obsługuje 10 różnych schowków. Pozwala również na kontrolę globalnego schowka systemu operacyjnego i przechwytywanie wszystkich informacji, które są kopiowane również w innych programach.

Wytnij i dopisz	Wycina zaznaczony tekst i dokleja go do zawartości schowka;
Kopiuj i dopisz	Kopiuje zaznaczony tekst doklejając go do zawartości schowka;
Konwertuj	Odnosi się do zaznaczonego bloku tekstu. Pozwala na konwersję tego bloku na różne sposoby, np. konwertując odpowiednio wielkość znaków, zamieniając znaki cudzysłowu na apostrofy lub odwrotnie, konwertując niektóre znaki by były widziane w dokumentach HTML itp.;
Sortuj	Sortuje linie dokumentu w kolejności alfabetycznej;
Powiel linie	Powielą aktywną linię dokumentu;
Zamień linie miejscami	Zamienia aktywną i następną linię miejscami;
Połącz linie	Łączy ze sobą aktywną i następną linię;
Rozdziel linie	Rozdziela linie na wiele linii według zadanego separatora;
Przesuń linie do góry	Przesuwa aktywną linię do góry;
Przesuń linie do dołu	Przesuwa aktywną linię do dołu;
Przetwórz	Menu zawiera dodatkowe funkcje dokonujące różnych operacji na tekście, np. usuwanie duplikatów, usuwanie białych znaków itp.;
Szukaj ⁵	Szuka zadanego wzorca;
Szukaj następny	Szuka kolejnego wystąpienia zadanego wzorca;

⁵Podczas wyszukiwania można posługiwać się wyrażeniami regularnymi, które również program CoreEditor obsługuje.

Szukaj poprzedni	Szuka poprzedniego wystąpienia zadanego wzorca;
Zastąp	Odszukuje zadanego wzorca i zastępuje go innym tekstem;

1.4.6. Edycja dokumentów w praktyce

Istnieje kilka różnych elementów, które bardzo usprawniają edycje dokumentów w programie CoreEditor, a których nie posiadają często edytory konkurencyjne. Te elementy i sytuacje to m.in.:

- Często w dokumentach umieszczamy kod odwołujący się do innych plików. Może to być np. instrukcja `include()` w języku PHP, dyrektywa `#include` w języku C/C++ czy też `\include` w języku \LaTeX . Często w takich sytuacjach zachodzi potrzeba utworzenia nowego dokumentu, do którego się właśnie odnosimy. W programie CoreEditor uzyskamy to bardzo szybko. Wystarczy kliknąć prawy klawisz myszy na nazwie pliku, do którego się odnosimy, a następnie z menu kontekstowego wybrać jedną z opcji menu **Plik**.
- Umieszczając kursor przed nawiasem, CoreEditor automatycznie zaznacza nawias do pary, do wybranego nawiasu. Ułatwia to kontrolę nawiasów, gdy jest ich bardzo dużo, jedne w drugich zagnieżdżonych. Podobnie zaznaczane są bloki w różnych językach programowania np. słowa kluczowe `begin` i `end` w języku Pascal, czy też środowiska `\begin` i `\end` w \LaTeX -u.
- Często pracując z dokumentami HTML lub XHTML występuje wiele zagnieżdżonych w sobie bloków `<table>` lub `<div>`. Znajdując się w obrębie jednego z tych bloków i wciskając kombinację klawiszy **CTRL+Q** sprawimy, że CoreEditor automatycznie odszuka znacznik zamykający dany blok i zaznaczy cały tekst objęty tym blokiem.

- Wciskając kombinację klawiszy **SHIFT+ENTER** wyświetlimy automatyczną pomoc kontekstową związaną z miejscem, w którym się aktualnie znajdujemy. Mogą to być np. wszystkie znaczniki języka XHTML, wszystkie możliwe atrybuty dla znacznika, w którym się znajdujemy, wszystkie możliwe wartości atrybutu, opcje związane z językiem CSS, funkcje i zmienne wykorzystane w językach programowania itp.
- CoreEditor podczas zapisywania dokumentu, zapamiętuje również takie informacje jak: pozycja kursora w dokumencie, sposób kolorowania składni, sposób kolorowania poszczególnych linii, ustawienia zakładek, zadania do wykonania na danym dokumencie itp. Dzięki temu ponowne otwarcie dokumentu przywraca wszystkie parametry ostatnio ustawione dla danego pliku.
- CoreEditor tworzy automatycznie kopie bezpieczeństwa dokumentów, na których pracuje. Dostęp do tych kopii możliwy jest za pośrednictwem zakładki **Kopie plików** widocznej w oknie **Wyników**. Użytkownik ma możliwość podglądu pliku na dowolnym jego etapie, jak również może wyświetlić listę zmian, jakie zaszły od tamtego momentu do aktualnej postaci dokumentu.
- Dla różnych języków wspieranych przez program stworzone zostały dodatkowe menu, zawierające szablony oraz różne gotowe funkcje. Takie dodatki przewidziane zostały dla języków XHTML, JavaScript, PHP, \LaTeX itd.
- Tworząc dokumentację do swojego projektu nie musimy sięgać po żadne inne edytory. CoreEditor pozwala na edycję plików RTF (ang. *Rich Text Format*). Możliwe jest osadzanie w takich plikach rysunków, tabel, a nawet wzorów w oparciu o obiekt *Microsoft Equation 3.0*⁶.

⁶Jeśli obiekt ten został zainstalowany w systemie.

1.4.7. Sprawdzanie pisowni, Thesaurus i autokorekta

Program CoreEditor wspomaga pisanie dokumentów w języku polskim. Obsługuje w tym celu takie mechanizmy jak:

- **Sprawdzanie pisowni** - posiada wbudowaną bazę kilkuset tysięcy poprawnie zapisanych wyrazów w języku polskim. Dzięki tak bogatej bazie, CoreEditor w skuteczny sposób dba o ortografię redagowanych dokumentów. Sprawdzanie pisowni może odbywać się w sposób dynamiczny, tzn. na etapie pisania dokumentu poprzez zaznaczanie czerwoną linią nieznanymi lub błędnymi wyrazów lub poprzez włączenie narzędzia *Sprawdzania pisowni*. Po kliknięciu prawym klawiszem myszy na błędnie zapisanym wyrazie w menu kontekstowym pojawia się pozycja *Sugestie*, pozwalająca na wybór poprawnej wersji danego słowa lub też dodanie nowego słowa do bazy.
- **Thesaurus** - narzędzie zawierają bibliotekę⁷ kilkudziesięciu tysięcy wyrazów i ich synonimów. Dzięki temu, program może podpowiadać jakiego innego wyrazu użyć, by nie tworzyć zbyt licznych powtórzeń w dokumencie. Aby przejrzeć listę synonimów do danego słowa, wystarczy kliknąć na nim prawy klawisz myszy. Jeśli jakieś synonimy zostaną odnalezione, pojawi się nowa pozycja *Synonimy* z listą znalezionych zwrotów.
- **Autokorekta** - to narzędzie, które automatycznie poprawia błędnie wpisywane wyrazy. W bazie programu CoreEditor znajduje się kilkaset typowo popełnianych błędów.

Po przełączeniu wersji językowej programu CoreEditor na angielski, wczytywane są również bazy przewidziane dla danego języka, a więc słownik ortograficzny i baza autokorekty.

⁷CoreEditor obsługuje otwartą bibliotekę synonimów dostępną na licencji GNU. Najnowszą wersję biblioteki można znaleźć na stronie <http://synonimy.sourceforge.net/>

1.4.8. Tryb VIM

CoreEditor jako jeden z nielicznych narzędzi tego typu w środowisku Windows wspiera klawiszologię typową dla programów Vi i Vim. Wciskając klawisz **ESC** w pasku statusu pojawia się pozycja **Command** i program CoreEditor oczekuje na komendę. W chwili obecnej jeszcze nie wszystkie komendy są obsługiwane.

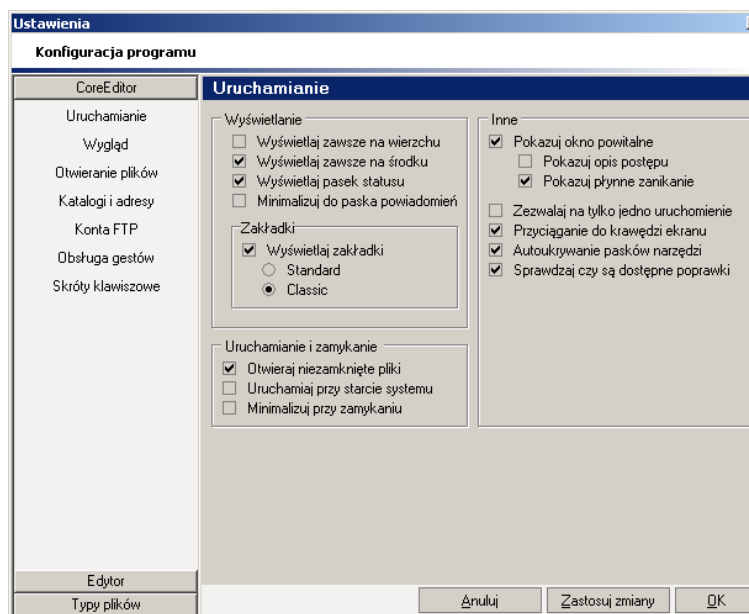
Tabela 1.5: CoreEditor - Komendy w trybie VIM

<i>Komenda</i>	<i>Opis</i>
:w lub :write	Zapisanie dokumentu;
:close	Zamknięcie dokumentu;
:new	Utworzenie nowego dokumentu;
:f lub :sf lub :e lub :edit	Otworzenie dokumentu;
:q lub :quit	Zamknięcie programu;
:q! lub :quit!	Natychmiastowe zamknięcie programu bez zapisywania zmian;
:x lub :wq	Zapisanie zmodyfikowanych dokumentów i zamknięcie programu;

1.5. Ustawienia program

Niemal wszystkie systemy ułatwiające pisanie dokumentów czy też wpływające na wygląd i zachowanie programu - można dowolnie dostosowywać według własnych upodobań. Wszystko to zapewnia okno *Ustawienia*, do którego można przejść wybierając z menu **Opcje** opcję **Ustawienia** → **Ustawienia**. Na rysunku 1.8 pokazano wygląd tego okna.

Okno *Ustawienia* składa się z rozwijanego menu widocznego po lewej stronie, oraz części głównej, w której wyświetlają się opcje związane z wybranym elementem menu, po prawej stronie.



Rysunek 1.8. CoreEditor - Ustawienia programu

Z uwagi na liczbę różnorodnych opcji, wszystkie one zostały pogrupowane i w ramach każdej grupy dostępne są jeszcze dodatkowe podstrony. Dostępne grupy opcji to:

- **CoreEditor** - w grupie tej znajdują się opcje związane z całym programem. Można tutaj określać wygląd programu, tzn. skórkę, typ wyświetlanych zakładek, widoczność niektórych elementów. Można też definiować własne skróty klawiaturowe oraz własne gesty myszy.
- **Edytor** - za pomocą opcji z tej grupy można modyfikować główną kontrolkę edycyjną. Można wpływać na wygląd tej kontrolki, modyfikować schematy kolorów wykorzystywanych przy kolorowaniu składni, czy też definiować parametry związane z modułem sprawdzania pisowni.
- **Typy plików** - w tej grupie znajdują się strony związane z opcjami dla każdego z obsługiwanych typów plików. Dla każdego wspieranego typu można określić rozszerzenia, domyślny schemat kolorowania, czy też zdefiniować charakterystyczne kody, po których program może rozpoznawać dany typ. W oknie tym można także skojarzyć pliki

danego typu z programem CoreEditor przypisując im odpowiednią ikonę. Za pośrednictwem przycisku **Edytuj AUTOSNIPPETS** można edytować bazę automatycznych szablonów dla danego typu. Z kolei wybierając przycisk **Zmień ustawienia specyficzne** można wywołać okno pozwalające na konfiguracje zewnętrznych kompilatorów czy też systemów pomocy związanych z danym typem pliku.

1.6. Matematyka w programie

CoreEditor może służyć jako dość rozbudowane środowisko do różnorodnych obliczeń numerycznych. Zawdzięcza to *Interpreterowi CE*, w który został wyposażony. Więcej informacji na ten temat znalazło się w rozdziale 3. W tym miejscu, omówione zostaną jedynie pokrótce informacje podstawowe związane z matematyką w programie.

CoreEditor pozwala na:

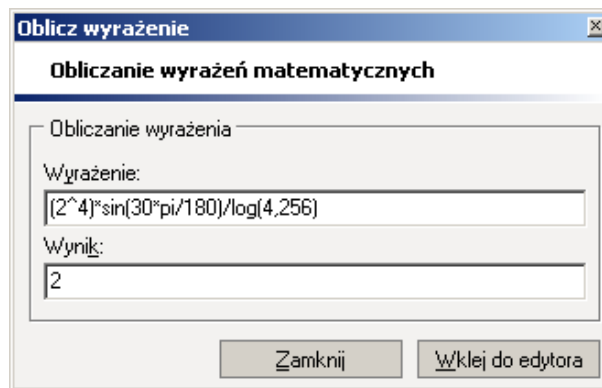
- dokonywanie obliczeń zarówno na liczbach rzeczywistych jak również zespolonych,
- operowanie na macierzach,
- definiowanie dowolnej liczby własnych zmiennych,
- definiowanie dowolnej liczby własnych funkcji,
- obliczanie pochodnych funkcji w punkcie,
- odszukiwanie miejsc zerowych funkcji,
- obliczanie długości łuków,
- obliczanie całek oznaczonych,
- operowanie wyrażeniami logicznymi.

Zadania do obliczenia można wprowadzać do programu na kilka sposobów:

1. Za pośrednictwem okna *Interpreter CoreEditor*. Więcej na ten temat znajduje się w rozdziale 3.
2. Za pośrednictwem okna *Oblicz wyrażenie*, dostępnego w menu **Narzędzia**.
3. Bezpośrednio w tekście dokumentu.

1.6.1. Okno Oblicz wyrażenie

Chcąc dokonać prostych obliczeń w programie CoreEditor można skorzystać z okna *Oblicz wyrażenie*. Aby je wywołać, należy wybrać tę opcję z menu **Narzędzia**. Ukaże się wówczas okno jak na rysunku 1.9.



Rysunek 1.9. CoreEditor - Okno Oblicz wyrażenie

Okno to składa się z dwóch pól tekstowym. W polu **Wyrażenie** należy wprowadzać dowolne wyrażenie arytmetyczne, a w polu **Wynik**, wyświetlać się będzie wynik obliczeń. Należy zwrócić uwagę na fakt, że wynik pojawia się w każdym momencie wprowadzania wyrażenia, więc nie ma potrzeby posługiwania się klawiszem ENTER. Za pomocą przycisku **Wklej do edytora** można wkleić wynik obliczeń do dokumentu.

1.6.2. Obliczenia wewnątrz dokumentu

Inna metoda dokonywania różnorodnych obliczeń, to wprowadzanie ich bezpośrednio w treści redagowanego dokumentu. Po wprowadzeniu ta-

kiego wyrażenia w dokumencie można posłużyć się jedną z opcji wyświetlanych w menu **Obliczenia** pokazującym się po wciśnięciu prawego klawisza myszy. Wybrana opcja odnosić się będzie do:

- linii w której się aktualnie znajduje kursor lub
- zaznaczonego bloku tekstu.

Jeśli zaznaczony jest blok tekstu, wynik obliczeń zostanie wprowadzony w jego miejsce. Jeśli zależy nam, by nie usuwać samego wyrażenia na którym dokonujemy obliczeń, zaznaczany blok tekstu należy zakończyć znakiem równości, tj. =. Wówczas wynik wstawiony będzie po tym znaku.

W menu **Obliczenia** dostępne są następujące pozycje:

- Oblicz,
- Oblicz sumę,
- Oblicz różnicę,
- Oblicz iloczyn,
- Oblicz średnią arytmetyczną,
- Oblicz średnią geometryczną,
- Oblicz średnią harmoniczną,
- Oblicz średnią kwadratową.

Pierwsza z opcji odnosi się do jednego wyrażenia, dla którego CoreEditor próbuje wyznaczyć wartość. Z kolei wszystkie pozostałe odnoszą się do wielu wyrażeń. Każde takie wyrażenie składowe jest obliczane, a następnie na wszystkich uzyskanych wynikach dokonywana jest określona operacja.

Oto przykład kroków, jakie należy wykonać, aby wypróbować wspomniane narzędzie:

1. Wprowadzamy do dokumentu tekst:

Listing 1.1. Obliczanie średniej arytmetycznej

```
1 2 5 sin(30*pi/180)
2 2.3 3^2 5!
3 =
```

2. Zaznaczamy cały ten wprowadzony blok tekstu (również ze znakiem równości);
3. Klikamy prawy klawisz myszy i z menu **Obliczenia** wybieramy **Oblicz średnią arytmetyczną**;
4. W oknie, po znaku równości, umieszczony zostanie wynik, tj. 23.1333333333333.

Średnie w programie CoreEditor wyliczane są w oparciu o następujące wzory [12]:

$$S_a = \frac{\sum_{i=1}^n a_i}{n}$$

$$S_g = \sqrt[n]{\prod_{i=1}^n a_i}$$

$$S_h = \frac{n}{\sum_{i=1}^n \frac{1}{a_i}}$$

$$S_k = \sqrt{\frac{\sum_{i=1}^n a_i^2}{n}}$$

1.7. Inne ciekawe funkcje

CoreEditor to dość rozbudowany program i omówienie jego wszystkich funkcji zajęłoby bardzo dużo miejsca. Dlatego chcąc zapoznać się dokładniej z oferowanymi przez niego możliwościami, warto zajrzeć do *Systemu pomocy*, odwiedzić stronę domową⁸, czy też odwiedzić serwis [35], w którym można znaleźć różne materiały na jego temat.

⁸<http://www.coreeditor.pl>

W tym miejscu natomiast, jedynie w formie punktów wymieniono możliwości programu, o których nie mówiono w tej pracy, a zdaniem autora są na tyle ciekawe, że warto je poznać.

CoreEditor:

- posiada wbudowanych 16. skórek, dzięki czemu można dostosować jego wygląd według własnych upodobań;
- obsługuje kilka różnych dialektów dla języka SQL, m.in: MySQL, Interbase6, Oracle, Sysbase;
- pozwala na pracę na zdalnych plikach poprzez wbudowany klient FTP;
- pozwala na eksport dokumentów do formatów HTML, RTF czy \TeX ;
- posiada wbudowany edytor dokumentów RTF;
- posiada wbudowany edytor binarny;
- posiada wbudowaną przeglądarkę grafiki obsługującą m.in. takie formaty jak BMP, JPG, GIF, PNG, PCX, PSP, PSD, RGB, TIFF, TGA, WMF, WBMP. Pozwala również na konwersję plików graficznych pomiędzy formatami BMP, JPG, GIF, PNG i WBMP;
- posiada wbudowany odtwarzacz Video wykorzystujący zainstalowane codecki przy odtwarzaniu filmów;
- posiada wbudowany odtwarzacz plików dźwiękowych, obsługujący takie formaty jak WAV czy MP3;
- obsługuje 10 odrębnych schowków, tzw. schowek rotacyjny, czy też pozwala na podgląd schowka globalnego;
- posiada wbudowane narzędzie lupy oraz obsługę syntezerów mowy;
- posiada dodatkowe narzędzia takie jak: *Szukaj/Zastąp w plikach*, *Porównywanie plików*, *Kompresor HTML*, czy *Formatowanie kodu HTML*;

- posiada wbudowany rozbudowany konwerter tekstu;
- posiada narzędzie *Określ zaznaczenie* pozwalające na zapisanie dowolnego fragmentu ekranu w pliku graficznym;
- umożliwia zapisywanie dokumentów z zaznaczaniem końców wierszy w stylu DOS i Unix;
- posiada obsługę (eksperymentalną) zwijania kodu;
- posiada wbudowany, prosty klient bazy MySQL i PostgreSQL;
- pozwala na definiowanie własnych skrótów klawiszowych do dowolnych funkcji;
- pozwala na definiowanie własnych gestów myszki i przypisanie pod te gesty dowolnych funkcji;

Wspierane języki

W rozdziale zaprezentowane zostały informacje na temat mechanizmów i funkcji w jakie został wyposażony program CoreEditor, aby wspierać proces tworzenia stron internetowych, czy też pomagać w składaniu dokumentów za pomocą \LaTeX -a.

2.1. Wstęp

CoreEditor obsługuje kilkanaście różnych języków programowania i języków skryptowych, a składnię potrafi kolorować dla kilkuset różnych języków. Lista wspieranych kompilatorów zebrana została w tabeli 2.1.

Program stwarza również możliwość samodzielnego rozszerzanie możliwości programu o nowe języki. Użytkownik może samodzielnie definiować nowe typy plików, definiować schematy kolorowania składni dla tych języków, a nawet dodawać obsługę nowych kompilatorów, podłączać dokumentacje i zewnętrzne systemy pomocy. Niemniej jednak, niektóre języki zyskały w programie szczególnie istotne znaczenie, dlatego wsparcie dla nich jest wyjątkowo silne. Jako, że program CoreEditor to środowisko głównie to tworzenia stron internetowych oraz redagowania dokumentów w \LaTeX -u, właśnie języki związane tymi technologiami, mają wyjątkowe wsparcie.

Tabela 2.1: CoreEditor - Wspierane kompilatory

<i>Język</i>	<i>Nazwa kompilatora</i>
PHP	Interpreter języka PHP (php.exe lub php-cgi.exe)

PERL	Interpreter języka PERL (<code>perl.exe</code>)
C/C++	Kompilatory GCC i G++ (z pakietu DEV-CPP, MinGW)
C/C++	Kompilator Borland C++ v 5.5
Pascal	Kompilator FPC (Free Pascal Compiler)
Pascal	Kompilator TMT Pascal
Pascal	Kompilator GPC
Pascal	Kompilator Delphi
Asembler	Kompilator A86
Asembler	Kompilator NASM
Basic	Kompilator FreeBasic
Basic	Kompilator GLBCC
Java	Kompilator Sun Java
Python	Interpreter Python (<code>python.exe</code>)
LaTeX	Kompilatory z pakietu MiKTeX
C#	Kompilator Microsoft Visual C# (<code>csc.exe</code>)
Nemerle	Kompilator Nemerle

2.2. Języki HTML i XHTML

Języki HTML (ang. *HyperText Markup Language*) i XHTML (ang. *Extensible HyperText Markup Language*) posiadają wyjątkowo duże wsparcie ze strony programu CoreEditor. Przejawia się ono m.in. następującymi elementami:

- **Automatyczne zamykanie znaczników** - to funkcja polegająca na automatycznym zamykaniu znaczników języków HTML i XHTML. Działa ona w ten sposób, że po rozpoczęciu wpisywania znacznika zamykającego, CoreEditor automatycznie operację tę dokończy.
- **System szablonów** - możliwe jest utworzenie nowego dokumentu w oparciu o szablon HTML lub XHTML. Również wybierając z menu

Plik opcję **Nowy**, a następnie **Z szablonu** sprawimy, że pojawi się kreator, a w nim dostępnych będzie kilka różnych szablonów stron WWW.

- **System pomocy** - w skład systemu pomocy wchodzi kurs języków HTML i XHTML¹. Dostępna jest również pomoc kontekstowa, tzn. po kliknięciu prawym klawiszem myszy na jakimś znaczniku lub atrybucie znacznika, w menu kontekstowym pokaże się pozycja **Szukaj w dokumentacji**, która powoduje odszukanie pomocy do wybranego elementu.
- **Menu oraz paski narzędzi** - po utworzeniu dokumentu HTML lub XHTML pojawiają się nowe paski narzędzi oraz nowe elementy w menu. Zawierają one różne gotowe szablony najczęściej wprowadzanych znaczników i całych partii kodu.
- **Kreatory** - do programu CoreEditor dołączono kilkanaście różnych kreatorów w postaci zewnętrznych bibliotek DLL. Dostęp do tych kreatorów zapewnia jeden z pasków narzędzi oraz menu **HTML**. Kreatory pomagają w takich czynnościach jak: wstawianie hiperłączy, wstawianie rysunków, wstawianie filmów Flash, wstawianie poziomych linii, tworzenie rozbudowanych tabel, tworzenie rozbudowanych sekcji **HEAD** i **BODY**, tworzenie tzw. map odsyłaczy, i wiele innych. Na rysunku 2.2 pokazano wygląd okna edytora *Mapa rysunku*.
- **System podpowiedzi** - to rozbudowany system podpowiedzi aktywowany kombinacją klawiszy SHIFT+ENTER. Użycie tej kombinacji spowoduje, że CoreEditor wyświetli listę dostępnych znaczników. Jeśli użyjemy tej kombinacji będąc kursorem w obrębie jakiegoś znacznika, pojawi się lista dostępnych dla tego znacznika atrybutów. Będąc

¹Kursy udostępnione zostały zostały przez Dariusza Majgiera oraz Sławomira Kokłowskiego.

z kolei w polu wartości dla określonego atrybutu, ukaze się lista dostępnych dla niego wartości.

- **Inspektor obiektów** - pojawia się w postaci dodatkowej zakładki w *Panelu narzędzi*. Kiedy kursor znajduje się w obrębie jakiegoś znacznika, w Inspektorze pojawia się lista wszystkich możliwych dla tego znacznika atrybutów.
- **Dynamiczne podpowiedzi** - wyświetla się w postaci panelu na *Panelu narzędzi*. Wyświetlają tam się różne informacje, np. prostokąt w kolorze, na którym znajduje się aktualnie kursor. Kolor może być zapisaną stałą, np. Red, Blue czy też kodem, np.: #ff00ff.
- **Tryb WYSIWYG** (ang. *What You See Is What You Get*) - dodatkowa zakładka w edytorze, która umożliwia na pracę nad dokumentem HTML w taki sposób, jak w programach typu *Microsoft Word*. Do poprawnej pracy tego modułu niezbędne są dodatkowe biblioteki udostępnione przez firmę Microsoft.
- **Podgląd** - dodatkowa zakładka w edytorze. Wyświetla się w niej podgląd do tworzonego dokumentu. Można się na nią przełączać również za pomocą skrótu klawiaturowego wciskając kombinację CTRL+B. CoreEditor do wyświetlania podglądu może wykorzystywać przeglądarkę internetową Internet Explorer lub Mozilla, po doinstalowaniu dodatkowych bibliotek.
- **Podgląd dynamiczny** - umożliwia redagowanie dokumentu bezpośrednio w kodzie HTML/XHTML i jednoczesny podgląd wynikowy strony we wbudowanej przeglądarce.
- **Paleta kolorów** - rozbudowana paleta kolorów, zawierająca spektra różnych barw, paletę kolorów bezpiecznych, próbnik kolorów itp. Na

rysunku 2.1 zaprezentowano przykładowy widok okna tego narzędzia.

- **Kodowanie znaków** - CoreEditor dba o poprawne kodowanie polskich znaków. Wspiera najważniejsze systemy kodowania znaków, tj. ISO 8859-2 i Windows 1250 oraz obsługuje kodowanie UTF-8 i UTF-16.
- **Obsługa programu Tidy** - podczas instalacji programu, dołączany jest również darmowy program Tidy, opracowany przez W3C. CoreEditor obsługuje w pełni możliwości tego programu poprzez różne opcje w menu **HTML** → **Tidy**. Dzięki Tidy można w prosty sposób sformatować kod języka HTML/XHTML, oczyścić ten kod ze *śmieci*, przekonwertować pomiędzy różnymi standardami, sprawdzić jego poprawność itp.
- **Praca z projektami** - CoreEditor pozwala na tworzenie strony internetowej w postaci projektu. Dzięki takiemu podejściu można w prosty sposób tworzyć kopie bezpieczeństwa całego serwisu lub też aktualizować swój serwis w internecie w błyskawiczny sposób, dzięki zastosowaniu systemu synchronizacji.
- **System AutoSnippets** - działanie tego systemu polega na automatycznym uzupełnianiu kodu po wprowadzonej pewnej sekwencji oraz skorzystaniu z kombinacji klawiszy SHIFT+SPACJA. Na przykład, wpisując słowo *lorem* i wciskając ową kombinację, do dokumentu wstawi się tekst *Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...*



Rysunek 2.1. CoreEditor - Paleta kolorów

2.3. Język CSS

Podobnie jak język HTML/XHTML, również język CSS (ang. *Cascading Style Sheets*) posiada rozbudowane wsparcie ze strony programu CoreEditor. Wsparcie to, to m.in.:

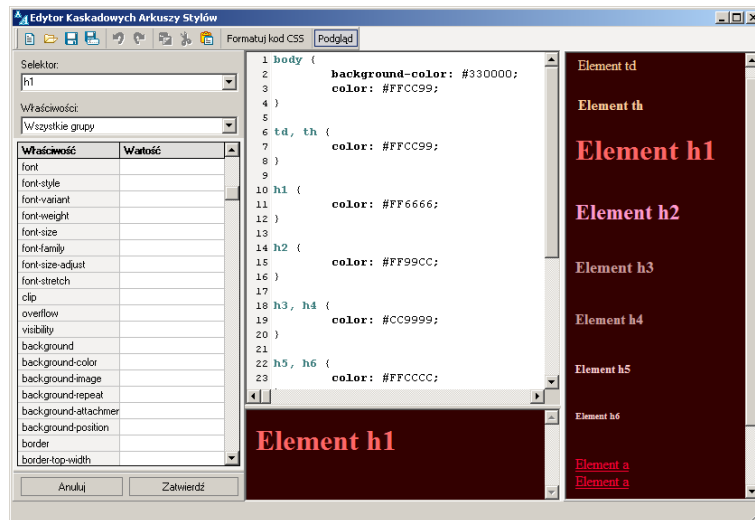
- **System szablonów** - Wybierając z menu **Plik** opcję **Nowy**, a następnie **Z szablonu** sprawimy, że pojawi się kreator, a w nim dostępnych będzie kilka różnych szablonów kaskadowych arkuszy stylów. Można więc za pomocą kilku kliknięć uzyskać schemat jednolitych kolorów dla tła, liter, odnośników i nagłówków na stronie.
- **System pomocy** - w skład systemu pomocy wchodzi kurs języka CSS. Dostępna jest również pomoc kontekstowa działająca w ten sam sposób jak w przypadku języków HTML i XHTML.
- **Edytor Kaskadowych arkuszy stylów** - Rozbudowany edytor usprawniający proces pisania plików CSS. Na rysunku 2.3 zaprezentowano przykładowy zrzut okna edytora.
- **System podpowiedzi** - rozbudowany system podpowiedzi aktywowany kombinacją klawiszy SHIFT+ENTER. Użycie tej kombinacji spo-



Rysunek 2.2. CoreEditor - Edytor map odsyłaczy

woduje, że CoreEditor wyświetli listę dostępnych właściwości lub wartości dla wybranych właściwości.

- **Inspektor obiektów** - pojawia się w postaci dodatkowej zakładki w *Panelu narzędzi*. Kiedy kursor znajduje się w obrębie jakiegoś selektora, w Inspektorze pojawia się lista wszystkich możliwych właściwości CSS.
- **Dynamiczne podpowiedzi** - wyświetla się w postaci panelu na *Panelu narzędzi*. Wyświetla się tam podgląd elementów sformatowanych według danej definicji stylów.
- **Podgląd** - dodatkowa zakładka w edytorze. Wyświetla się w niej podgląd do tworzonego dokumentu CSS.



Rysunek 2.3. CoreEditor - Edytor Kaskadowych arkuszy stylów

2.4. Język JavaScript

Wsparcie ze strony programu CoreEditor dla języka JavaScript zapewniają następujące elementy:

- **Szablony funkcji** - W menu **JavaScript** odnaleźć można gotowe szablony funkcji realizujących różne, często spotykane operacje, np.: kontrolowanie poprawności numerów NIP, PESEL, adresów e-mail, skalowaniu obrazków itp.
- **Snippets** - W zakładce **Szablony** w **Panelu katalogów** można również znaleźć szereg gotowych kodów dla języka JavaScript.
- **System podpowiedzi** - rozbudowany system podpowiedzi aktywowany kombinacją klawiszy **SHIFT+ENTER**. Użycie tej kombinacji spowoduje, że CoreEditor wyświetli w zależności o kontekstu listę dostępnych funkcji, składowych wybranego obiektu, listę zmiennych itp.

2.5. Język PHP

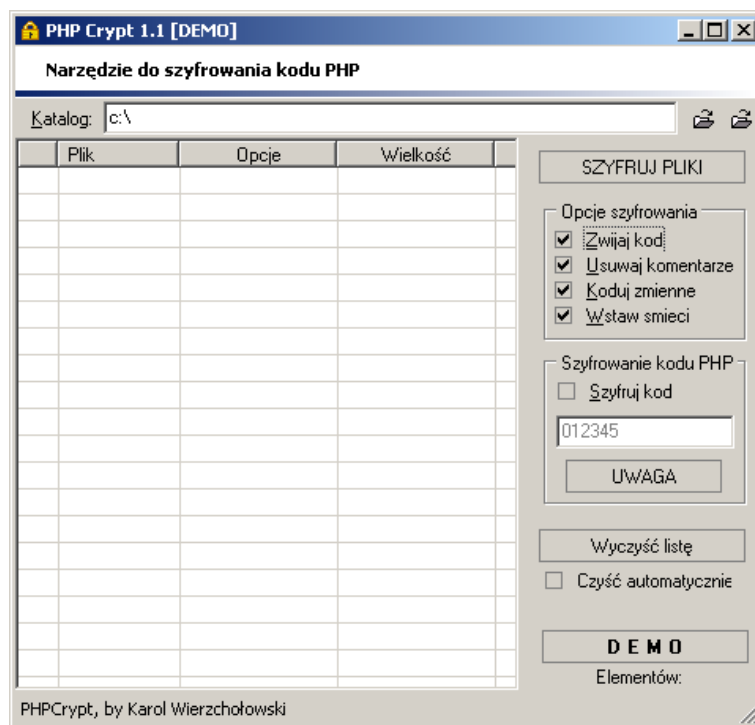
Pisanie skryptów w języku PHP również staje się prostsze z wykorzystaniem programu CoreEditor. Środowisko to zawiera bowiem następujące elementy ułatwiające tę pracę:

- **System pomocy** - w skład systemu pomocy wchodzi kurs języka PHP². Dostępna jest również pomoc kontekstowa, tzn. po kliknięciu prawym klawiszem myszy na jakiejś funkcji, w menu kontekstowym pokaże się pozycja **Szukaj w dokumentacji**, która powoduje odszukanie pomocy do wybranego elementu w dołączonym do programu podręczniku PHP.
- **Menu** - po utworzeniu dokumentu PHP pojawiają się nowe menu. Zawierają one różne gotowe szablony funkcji realizujących przydatne i często wykonywane operacje, np.: upload plików na serwer, skalowanie obrazków, kontrolę różnych numerów itp.
- **System podpowiedzi** - rozbudowany system podpowiedzi aktywowany kombinacją klawiszy SHIFT+ENTER. Użycie tej kombinacji spowoduje, że CoreEditor wyświetli w zależności od kontekstu listę dostępnych funkcji, składowych wybranego obiektu, listę zmiennych itp. Również umieszczając kursor po nawiasie otwierającym argumenty danej funkcji, CoreEditor wyświetli dymek z listą argumentów, jakie dana funkcja wymaga.
- **Podgląd** - mając odpowiednią konfigurację, CoreEditor może wyświetlać podgląd do dokumentów zapisanych w języku PHP, wysyłając je najpierw do interpretera PHP i przechwytyjąc wyniki przez niego generowane.
- **PHPCrypt** - narzędzie dołączone do programu w postaci zewnętrznej

²Kursy udostępnił Leszek Krupiński.

biblioteki DLL. Dostęp do niego dostępny jest poprzez menu **Narzędzia**. PHPCrypt to podprogram służący do zabezpieczania skryptów PHP przed modyfikacją ich przez nieodpowiednie osoby. Kod PHP może być zaśmiecany, zwijany do pojedynczej linii, mogą być w nim podmieniane nazwy zmiennych, wreszcie może być szyfrowany. Pomimo takiego zabezpieczenia, skrypt taki pozostaje nadal w pełni wykonywalny. Na rysunku 2.4 pokazano wygląd okna tego narzędzia.

- **System AutoSnippets** - działanie tego systemu polega na automatycznym uzupełnianiu kodu po wprowadzonej pewnej sekwencji oraz skorzystaniu z kombinacji klawiszy SHIFT+SPACJA. Na przykład, wpisując słowo `dowhile` i wciskając ową kombinację, do dokumentu wstawi się szablon instrukcji `do..while`.



Rysunek 2.4. CoreEditor - Narzędzie PHPCrypt

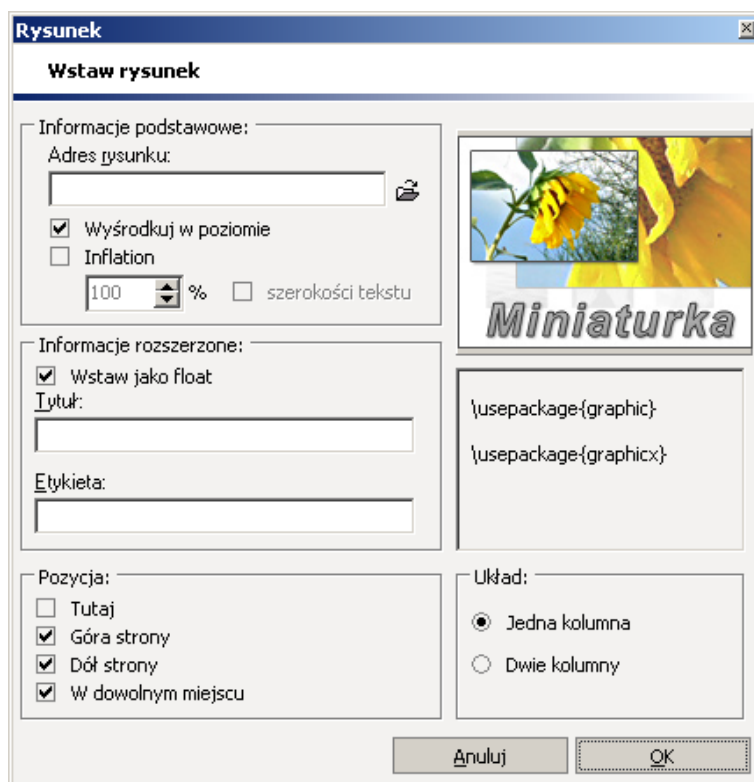
2.6. Język \LaTeX

Składanie dokumentów \LaTeX -a w programie CoreEditor staje się dużo łatwiejsze. Program ten bowiem zapewnia wsparcie dla tego języka w następujący sposób:

- **Pełna obsługa MiKTeX-a** - w pełni obsługuje kompilatory MikTeX-a. Wystarczy dokonać typowej instalacji tego środowiska, aby CoreEditor odnalazł wszystkie niezbędne pliki. Wówczas generowanie dokumentów PDF, DVI czy PS sprowadza się do jednego kliknięcia w ikonę. Również podgląd wygenerowanego pliku wymaga kliknięcia w jedną ikonę lub wciśnięcia klawisza F5.
- **Szablony i projekty** - CoreEditor zapewnia gotowe szablony dokumentów napisanych w języku \TeX . Umożliwia też pracę z takimi dokumentami na zasadzie projektów. W ramach modułu gotowych projektów znaleźć można szablony wizytówek oraz szablon gotowej publikacji.
- **Kolorowanie składni i zaznaczanie bloków** - Program koloruje składnię dla języka \LaTeX i automatycznie zaznacza środowiska `\begin` i `\end` do pary, tzn. po umieszczeniu kursora w obrębie polecenia `\begin` podświetlona zostanie odpowiadająca mu komenda `\end`.
- **Paski narzędzi i menu** - edytując dokument zapisany w języku \LaTeX ukazują się dwa dodatkowe paski narzędzi oraz menu **TeX** zapewniające dostęp do różnych gotowych komend tego języka.
- **System pomocy** - W skład systemu pomocy wchodzi dokumentacja do \LaTeX -a dostępna poprzez zakładkę **Dokumentacja** oraz poprzez menu kontekstowe jak również kurs pisania dokumentów w \LaTeX -u opracowany i udostępniony przez Ryszarda Pawła Kosteckiego³.

³W miarę krótki i praktyczny kurs \LaTeX -a w π^e minut.

- **Kreatory** - dostępne są dodatkowe kreatory ułatwiające wstawianie do dokumentów obrazków czy też definiowanie kolorów. Na rysunku 2.5 zaprezentowano wygląd okna *Rysunek*.
- **Podgląd spisu treści** - Podczas pracy programu na zakładce **Dokument** w panelu **Panel narzędzi** wyświetlany jest dynamicznie spis treści. Poprzez ów spis można w błyskawiczny sposób skoczyć do danej sekcji edytowanego dokumentu.
- **Generowanie podglądu** - Wybierając zakładkę **Podgląd** podczas edycji dokumentu możemy sprawić, że wygenerowane zostaną pliki JPG prezentujące poszczególne strony redagowanego pliku.



Rysunek 2.5. CoreEditor - Narzędzie wstawiania rysunków w języku \LaTeX

Interpreter CoreEditor

Sercem programu CoreEditor jest interpreter języka o tej samej nazwie, w skrócie nazywany językiem CE. Język ten został specjalnie opracowany na potrzeby programu i pozwala na sterowanie wszystkimi elementami edytora. Pozwala również na pisanie prostych skryptów. W rozdziale tym przybliżono możliwości interpretera CE oraz omówiono sposób posługiwania się nim.

3.1. Przeznaczenie interpretera

W założeniu interpreter CE miał być modułem ukrytym i jedynie pośredniczącym pomiędzy interfejsem programu, tzn. przyciskami, paskami narzędzi, elementami menu, a określonymi funkcjami wykonywanymi w momencie aktywacji tych elementów. Miało to stwarzać użytkownikowi możliwość tworzenia własnych skryptów, plugin-ów oraz definiowania własnych elementów menu czy pasków narzędzi. Podczas projektowania interpretera, autor doszedł do wniosku, że dodając do niego moduł operacji arytmetycznych i tworząc okno zapewniające dostęp do tego interpretera, można stworzyć dość ciekawe środowisko naukowe. W środowisku tym można bowiem pisać proste skrypty, czy też wywoływać funkcje z zewnętrznych bibliotek DLL, a więc odnosić się do programów skompilowanych w dowolnym innym języku. Wyniki generowane przez te programy i skrypty interpreter może przechwytywać, by następnie dokonywać na nich kolejnych operacji. Można wreszcie dokonywać różnorodnych obliczeń. Interpreter bowiem obsługuje liczby zespolone, macierze, pozwala na definiowanie różnorodnych własnych funkcji, zmiennych itp.

3.2. Wprowadzanie instrukcji

Instrukcje języka CE można wprowadzać za pomocą okna interpretera widocznego standardowo w lewym dolnym rogu programu. Okno to składa się z dwóch części. W dolnej części wpisuje się instrukcje języka CE, natomiast w górnej części widoczne są wyniki generowane przez interpreter.

Każda poprawnie zbudowana instrukcja powinna być zakończona średnikiem. Oto przykład:

Listing 3.1. Pierwsza instrukcja

```
1 CoreEditor>> sin(30*pi/180);
2 0,5
3 +OK [0,000]  :: Result: #250
```

Powyższa instrukcja powoduje obliczenie wartości funkcji sinus dla danego argumentu. W argumencie posłużono się wyrażeniem $30 * pi/180$, gdyż funkcja sinus domyślnie posługuje się radianami. Kod *pi* oznacza stałą π .

Zapis **+OK** oznacza, że instrukcja została wykonana poprawnie. W nawiasie znajduje się liczba określająca czas (w sekundach) jaki zajęło interpreterowi wykonanie instrukcji.

Wszystkie wyniki zwracane przez interpreter są zapamiętywane i otrzymują numer. Numer ten zapisany jest po słowie **Result:**. W przytoczonym przykładzie wynikiem jest liczba 0,5 i otrzymała ona numer #250. W kolejnych instrukcjach można posługiwać się tym wynikiem, wystarczy poprzedzić go znakiem #. Prezentuje to kolejny przykład:

Listing 3.2. Posługiwanie się wynikami

```
1 CoreEditor>> #250*#250;
2 0,25
3 +OK [0,000]  :: Result: #251
```

3.3. Liczby zespolone

Program CoreEditor pozwala na dokonywanie prostych obliczeń na liczbach zespolonych. Obecnie nie wszystkie wbudowane funkcje operują na takich liczbach, dlatego jeśli dana funkcja nie obsługuje liczb zespolonych program dokona automatycznej konwersji do liczby rzeczywistej poprzez odrzucenie części urojonej danej liczby. Podczas wprowadzania liczb zespolonych należy posługiwać się symbolem i dla oznaczenia części urojonej. Kolejny przykład prezentuje użycie liczb zespolonych w języku CE:

Listing 3.3. Posługiwanie się liczbami zespolonymi

```
1 CoreEditor>> 5i+(2-3i)^2;
2 -5 - 7i
3 +OK [0,000]   :: Result: #252
```

3.4. Macierze

CoreEditor pozwala na dokonywanie prostych operacji na macierzach. Macierze wprowadza się w nawiasach kwadratowych, kolejne wiersze umieszczając w jeszcze jednej parze takich nawiasów. Kolejny przykład pokazuje sposób wprowadzenia macierzy oraz dokonania na nich prostych operacji:

Listing 3.4. Posługiwanie się macierzami

```
1 CoreEditor>> [[1,2][3,4][5,6]];
2 [ 1 2 ]
3 [ 3 4 ]
4 [ 5 6 ]
5 +OK [0,000]   :: Result: #253
6 CoreEditor>> [[4,5][5,6][2,3]];
7 [ 4 5 ]
8 [ 5 6 ]
9 [ 2 3 ]
10 +OK [0,000]  :: Result: #254
11 CoreEditor>> #253+#254;
12 [ 5 7 ]
13 [ 8 10 ]
14 [ 7 9 ]
15 +OK [0,000]  :: Result: #255
16 CoreEditor>> det([[1,2][-2,3]]);
17 7
18 +OK [0,000]  :: Result: #256
```

CoreEditor wyposażony został w trzy funkcje pozwalające na generowanie macierzy, są to:

- **m1(n)** - tworzy macierz jednostkową wymiaru n ,
- **m0(m,n)** - tworzy macierz zerową wymiaru $m \times n$,
- **mr(m,n)** - generuje losową macierz wymiaru $m \times n$.

Oto przykład:

Listing 3.5. Automatyczne generowanie macierzy

```
1 CoreEditor>> m1(3);
2 [ 1  0  0 ]
3 [ 0  1  0 ]
4 [ 0  0  1 ]
5 +OK [0,000]  :: Result: #257
6 CoreEditor>> m0(2,3);
7 [ 0  0  0 ]
8 [ 0  0  0 ]
9 +OK [0,000]  :: Result: #258
10 CoreEditor>> mr(2,2);
11 [ -18  -16 ]
12 [  13   15 ]
13 +OK [0,019]  :: Result: #259
14 CoreEditor>> #259*#259;
15 [  116   48 ]
16 [ -39   17 ]
17 +OK [0,017]  :: Result: #260
```

3.5. Operatory arytmetyczne

Podczas wprowadzania różnych instrukcji CoreEditor pozwala na posługiwanie się wszystkimi podstawowymi operatorami arytmetycznymi. Zebrano je w tabeli 3.1. Oprócz operatora ! (silni), wszystkie pozostałe operatory są dwuargumentowe, a zatem umieszcza się je pomiędzy dwoma wyrażeniami. Kolejność wykonywania działań określonych przez te operatory jest zgodna z zasadami w matematyce.

Tabela 3.1: Interpreter CE - Operatory arytmetyczne

<i>Operator</i>	<i>Znaczenie</i>
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
^	potęgowanie
!	silnia

Korzystając z nawiasów okrągłych można grupować wyrażenia zmieniając w ten sposób kolejność wykonywania działań. Nawiasy można w sobie dowolnie zagnieżdżać. Kolejny przykład prezentuje sposoby wykorzystania niektórych z operatorów.

Listing 3.6. Operatory arytmetyczne

```

1 CoreEditor>> 2^4;
2 16
3 +OK [0,000] :: Result: #261
4 CoreEditor>> 5!;
5 120
6 +OK [0,000] :: Result: #262
7 CoreEditor>> #261-(2+3i);
8 14 - 3i
9 +OK [0,000] :: Result: #263
10 CoreEditor>> [[1,2][3,4]]*[[2,1][4,3]];
11 [ 10 7 ]
12 [ 22 15 ]
13 +OK [0,000] :: Result: #264

```

3.6. Operatory logiczne i porównania

CoreEditor potrafi wykonywać również operacje logiczne i warunkowe. Przy operacjach tego typu, jeśli wynikiem danego działania jest *prawda* wyświetlona zostanie liczba 1. Jeśli natomiast wynikiem jest *fałsz* wyświetlona zostanie liczba 0. W tabeli 3.2 zebrano wszystkie obsługiwane przez program operatory tego typu.

Tabela 3.2: Interpreter CE - Operatory logiczne

<i>Operator</i>	<i>Znaczenie</i>
==	sprawdza, czy wyrażenia są równe
!=	sprawdza, czy wyrażenia są różne
<	sprawdza, czy pierwsze wyrażenie jest mniejsze od drugiego
<=	sprawdza, czy pierwsze wyrażenie jest mniejsze bądź równe drugiemu
>	sprawdza, czy pierwsze wyrażenie jest większe od drugiego
>=	sprawdza, czy pierwsze wyrażenie jest większe bądź równe drugiemu
<=>	równoważność
=>	implikacja
&& lub AND	koniunkcja
lub OR	alternatywa
NOT	negacja

Listing 3.7. Operatory logiczne

```

1 CoreEditor>> (not (1 && 0))<=>(not 1 || not 0);
2 1
3 +OK [0,000]  :: Result: #265

```

3.7. Zmienne

Program CoreEditor pozwala na tworzenie i posługiwanie się dowolną liczbą zmiennych. Aby przypisać wartość zmiennej należy posłużyć się operatorem bezwzględnego przypisania tj.: ::= . Po lewej stronie od tego operatora należy umieścić identyfikator zmiennej, natomiast po prawej wartość przypisywaną. Oto przykład:

Listing 3.8. Posługiwanie się zmiennymi

```
1 CoreEditor>> $x:=30;
2 +OK [0.000]
3 CoreEditor>> sin($x*pi/180);
4 0.5
5 +OK [0.000]  :: Result: #265
```

Zmienna może być dowolnym ciągiem znaków, składającym się z liter języka angielskiego, cyfr i znaku podkreślenia. Identyfikator zmiennej nie może rozpoczynać się cyfrą. Uwaga! W programie CoreEditor ważna jest wielkość liter. Identyfikatory x i X oznaczają dwie różne zmienne!

3.8. Wbudowane funkcje matematyczne

Program CoreEditor wyposażony został w kilkadziesiąt różnych funkcji matematycznych. Zebrane one zostały w tabeli 3.3.

Tabela 3.3: Interpreter CE - Wbudowane funkcje matematyczne

<i>Nazwy funkcji</i>					
abs	sin	cos	tan	cotan	secant
cosecant	arcsin	arccos	arctan	arccotan	arcsec
arccsc	sinh	cosh	tanh	cotanh	secanth
cosecath	arcsinh	arccosh	arctanh	arccoth	arcsech
arccsch	radtodeg	degtorad	gradtodeg	degtograd	radtogra
gradtorad	ln	loge	log10	log	power
log2	combination	variation	sqr	sqrt	angle
sign	conjugate	inverse	trans	m0	m1
mr	norm	det	pi	e	

Większość z wbudowanych funkcji wymaga podania jednego argumentu. Jednakże niektóre z nich potrzebują dwóch lub więcej argumentów. Jest tak, na przykład, w przypadku funkcji **log**. Funkcja ta pozwala na obli-

czenie logarytmu o dowolnej podstawie. Wymaga ona podania dwóch argumentów - pierwszy z nich to liczba logarytmowana, natomiast drugi - to podstawa logarytmu. Poniższy przykład prezentuje sposób użycia tej funkcji by obliczyć wartość $\log_4 256$.

Listing 3.9. Posługiwanie się funkcjami wieloargumentowymi

```
1 CoreEditor>> log(256,4);
2 4
3 +OK [0.000]   :: Result: #266
```

3.9. Definiowanie funkcji

W programie CoreEditor można definiować dowolną liczbę własnych funkcji. Definicje funkcji są bardzo podobne do definicji zmiennych, wystarczy posłużyć się operatorem przypisania tj. `:=`. Po lewej stronie od tego operatora umieszczamy nazwę funkcji, natomiast po prawej jej definicję.

Listing 3.10. Definiowanie własnych funkcji

```
1 CoreEditor>> k:=2;
2 +OK [0.000]
3 CoreEditor>> k;
4 2
5 +OK [0.000]   :: Result: #270
6 CoreEditor>> f(x):=2*x;
7 +OK [0.000]
8 CoreEditor>> g(x):=sin(x);
9 +OK [0.000]
10 CoreEditor>> g(f(15*pi/180));
11 0.5
12 +OK [0.000]   :: Result: #271
13 CoreEditor>> t(x,y,z):=x^(y+z);
14 +OK [0.000]
15 CoreEditor>> t(2,2,3);
16 32
17 +OK [0.000]   :: Result: #272
```

W przykładzie pokazano sposób deklaracji funkcji 0-argumentowej oraz wielo-argumentowej. Pokazano także sposób zagnieżdżenia funkcji jednej w drugiej.

CoreEditor pozwala także na deklarowanie funkcji wektorowych (parametrycznych).

Listing 3.11. Definiowanie funkcji wektorowych

```
1 CoreEditor>> j(x):=(x,2*x);
2 +OK [0.000]
3 CoreEditor>> j(5);
4 [ 5 10 ]
5 +OK [0.000]  :: Result: #273
```

Wartości funkcji tego typu zwracane są w postaci macierzy wierszowej.

Istnieje wreszcie możliwość definiowania funkcji *klamrowych*. Oto przykład funkcji określającej wartość bezwzględną:

Listing 3.12. Definiowanie funkcji klamrowych

```
1 CoreEditor>> w(x):={x,x>=0},{-x,x<0};
2 +OK [0.000]
3 CoreEditor>> w(7);
4 7
5 +OK [0.000]  :: Result: #274
6 CoreEditor>> w(-9);
7 9
8 +OK [0.000]  :: Result: #275
```

3.10. Pochodne, całki, długości łuków i miejsca zerowe funkcji

Program CoreEditor może wyznaczać miejsca zerowe zdefiniowanej funkcji. Aby tego dokonać należy posłużyć się funkcją `calc`. Funkcji tej przekazujemy trzy argumenty. Pierwszym jest nazwa funkcji, natomiast kolejne dwa określają przedział poszukiwań miejsca zerowego.

Listing 3.13. Szukanie miejsc zerowych funkcji

```
1 CoreEditor>> f(x):=sin(x);
2 +OK [0.000]
3 CoreEditor>> calc('f',-1,1);
4 0
5 +OK [0.216]  :: Result: #247
6 CoreEditor>> calc('f',1,2);
7 ERR CODE:127 {Nie znaleziono miejsca zerowego} [0.126]
```

W bardzo prosty sposób można również obliczać długości łuków oraz wyznaczać całki oznaczone. Należy posłużyć się do tego celu funkcjami `length` oraz `integral`.

Listing 3.14. Obliczanie całki oznaczonej i długości łuku

```
1 CoreEditor>> f(x):=sin(x);
2 +OK [0.000]
3 CoreEditor>> integral('f',0,pi);
4 2
5 +OK [0.199]  :: Result: #246
6 CoreEditor>> length('f',0,pi);
7 3.8113
8 +OK [0.199]  :: Result: #248
```

CoreEditor wreszcie pozwala na obliczenie wartości pochodnej funkcji w punkcie. Do pochodnej danej funkcji odwołujemy się poprzedzając nazwę funkcji literą **d**.

Listing 3.15. Obliczanie pochodnej funkcji

```
1 CoreEditor>> f(x):=sin(x);
2 +OK [0.000]
3 CoreEditor>> f(0);
4 0
5 +OK [0.000]  :: Result: #245
6 CoreEditor>> df(0);
7 1
8 +OK [0.017]  :: Result: #246
```

3.11. Wbudowane funkcje systemowe

CoreEditor posiada kilkaset wbudowanych funkcji. Umożliwiają one sterowanie całym programem, tzn. tworzeniem dokumentów, operowaniem elementami z menu, paskami narzędzi, odwoływaniem się do konfiguracji programu, odwoływaniem się do tekstu znajdującego się w edytorze itp. Niektóre z funkcji pozwalają na wykonywanie różnych operacji związanych z tekstami, np. odszukiwanie wzorców, usuwanie duplikatów itp. Dzięki takiemu podejściu użytkownik programu może tworzyć własne elementy menu, własne paski narzędzi oraz oprogramowywać przyciski znajdujące się w tych elementach właśnie posługując się tymi funkcjami.

Z uwagi na liczbę wszystkich wbudowanych funkcji, zostały one podzielone na tzw. klasy lub przestrzenie nazw. Dostęp do składowych danej klasy (przestrzeni) uzyskuje się za pomocą operatora kropki.

Edytor Interpretera CE wyposażony również został w system *auto-dokańczenia*. Polega on na tym, że wprowadzając jakąś instrukcję języka

CE możemy wcisnąć klawisz **TAB**, a CoreEditor dokończy nazwę wprowadzanej funkcji. Jeśli wiele funkcji rozpoczyna się od ciągu znaków jaki wprowadzimy, wówczas wyświetlone zostanie okno z listą możliwych funkcji.

Wymienienie i omówienie każdej z wbudowanych w program CoreEditor funkcji zajęłoby z pewnością kilkadziesiąt stron. Dlatego w tym miejscu przytoczonych zostanie kilka najciekawszych, choć niekoniecznie najpraktyczniejszych funkcji.

- **Clipboard.*** - grupa funkcji operujących na systemowym schowku;
- **Clipboard.AsText** - zapewnia dostęp do systemowego schowka;
- **CloseAll** - zamyka wszystkie dokumenty;
- **CoreEditor.*** - grupa funkcji wyświetlających różne informacje o programie CoreEditor;
- **CoreEditor.Author** - wyświetla informacje o autorze programu;
- **Create.Document** - tworzy nowy dokument;
- **Demo** - wyświetla wbudowany wygaszacz ekranu;
- **Execute** - pozwala na uruchomienie zewnętrznego programu;
- **File.*** - grupa funkcji operujących na plikach;
- **FunctionFromLibrary** - pozwala na wywołanie funkcji znajdującej się w zewnętrznej bibliotece DLL;
- **Functions** - wyświetla listę wszystkich wbudowanych funkcji;
- **Game.Snake** - funkcja uruchamia ukrytą grę wbudowaną w program. Ponieważ gra ta jest ukryta, system *auto-dokańczenia* nie będzie podpowiadał tej nazwy.
- **HTTP.Get** - umożliwia pobranie pliku z sieci Internet;

- **MD5** - oblicza kod MD5 dla zadanego ciągu znaków;
- **MessageBox** - wyświetla okienko z zadaną informacją;
- **Pascal** - pozwala na wykonanie kodu zapisanego w języku Pascal;
- **SQLite.*** - grupa funkcji pozwalających na zarządzanie bazą SQLite;
- **String.*** - grupa funkcji operujących na napisach;
- **System.Uptime** - zwraca czas przez jaki system pracuje bez przerwy;
- **Timer.*** - grupa funkcji służących do tworzenia *timerów* (zadań wykonywanych cyklicznie);
- **Variables** - wyświetla listę wszystkich zdefiniowanych zmiennych;
- **Window.*** - grupa funkcji operujących na oknach;

3.12. Bazy danych a interpreter CE

Interpreter CE może również służyć jako klient SQL do różnych baz danych, głównie MySQL-a i PostgreSQL-a. Sposób połączenia oraz wydania kilku zapytań prezentuje kolejny przykład:

Listing 3.16. Interpreter i połączenie z bazą danych

```

1 CoreEditor>> SQL.Open('mysql','192.168.1.3',3306,'jacek','jacek','');
2 Połączyłem z bazą danych.
3 +OK [0.019]  :: Result: #491
4 CoreEditor>> SQL.Start;
5
6 | Database |
7 |-----|
8 | information_schema |
9 | coreeditor_new |
10 | karol |
11 | mysql |
12 | test |
13 |-----|
14 Count 5, Time: 0,00 s
15 +OK [0.017]
16 @> use karol
17 +OK [0.036]
18 @> show tables
19
20 | Tables.in_karol |
21 |-----|
22 | lang |
23 | menu |
24 | newsy |

```

```

25 | pages          |
26 | users          |
27 | ustawienia    |
28 |-----|
29 Count 6, Time: 0,06 s
30 +OK [0.073]
31 @> select * from lang
32 |-----|
33 | id | klucz          | slovo_pl          | slovo_en |
34 |-----|
35 | 8  | _title         | Moja strona domowa |           |
36 | 9  | _drukuj        | Drukuj             |           |
37 |-----|
38 Count 2, Time: 0,36 s
39 +OK [0.434]
40 @> exit

```

W przykładzie skorzystano z polecenia `SQL.Open()` w celu nawiązania połączenia z zewnętrzną bazą danych. Była to baza MySQL znajdująca się na komputerze o adresie `192.168.1.3`, pracująca na porcie `3306`. Nastąpiło logowanie na użytkownika o nazwie i hasle `jacek`.

Po nawiązaniu połączenia można wydawać zapytania do wybranej bazy posługując się poleceniem `SQL.Query()`. By jednak nie wpisywać za każdym razem tego polecenia, można wywołać funkcję `SQL.Start`. Funkcja ta otwiera sesję SQL. Od tego momentu wszystkie wpisywane komendy wędrują nie do interpretera CE, lecz bezpośrednio są wysyłane do połączonej bazy. Wydając polecenie `exit` - zakończono tę sesję.

Parametry do funkcji `SQL.Open()` to odpowiednio:

- protokół,
- adres serwera,
- port,
- nazwa użytkownika,
- hasło,
- domyślna baza danych.

Można również kliknąć prawy klawisz w wierszu wprowadzania komend Interpretera CE i z menu, które się wówczas pojawi wybrać opcję **Połącz z bazą danych SQL**. Pokaże się wówczas okno jak na rysunku

3.1. Wypełniając owe okno i klikając przycisk **Połącz z bazą** sprawimy, że kreator automatycznie skonstruuje odpowiednie polecenie `SQL.Open()` i wyda je interpreterowi CE.



Rysunek 3.1. CoreEditor - Okno połączenia z bazą danych

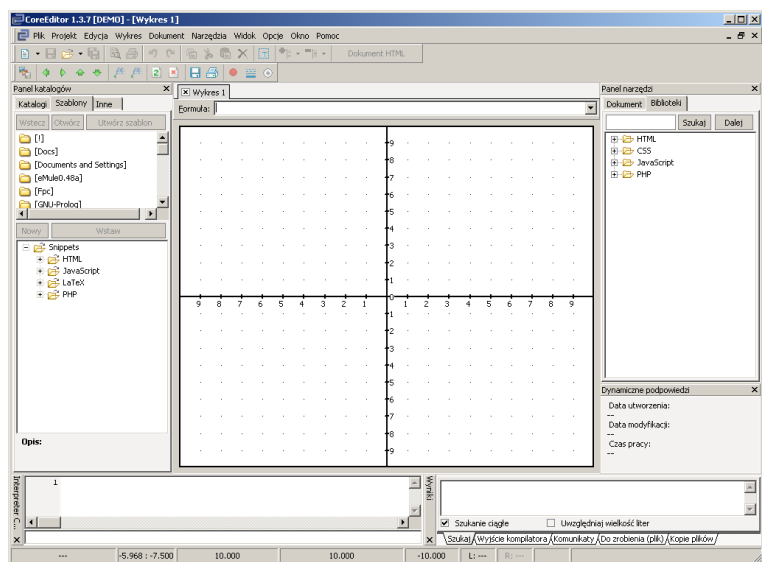
3.13. Dodatki

Jak zostało zaprezentowane wcześniej, interpreter CoreEditor-a potrafi wykonywać całkiem sporo różnych operacji matematycznych. Dlatego, aby wykorzystać go w pełni, do programu dołączone zostały dwie wtyczki (ang. *plugins*) rozszerzające jego możliwości o rysowanie wykresów oraz rysowanie fraktali.

3.13.1. Rysowanie wykresów

Aby uruchomić moduł rysowania wykresów należy z menu **Narzędzia** wybrać **Rysowanie wykresów**. Można również w oknie Interpretera CE wpisać komendę `plot`.

Ukaże się nowe okno jak na rysunku 3.2. W środkowej części obszaru roboczego widoczny jest układ współrzędnych oraz siatka. W tym miejscu rysowane będą wszystkie wykresy. Na samej górze tego okna znajduje się pole **Formuła**, w które należy wprowadzać formuły funkcji, które chcemy narysować.



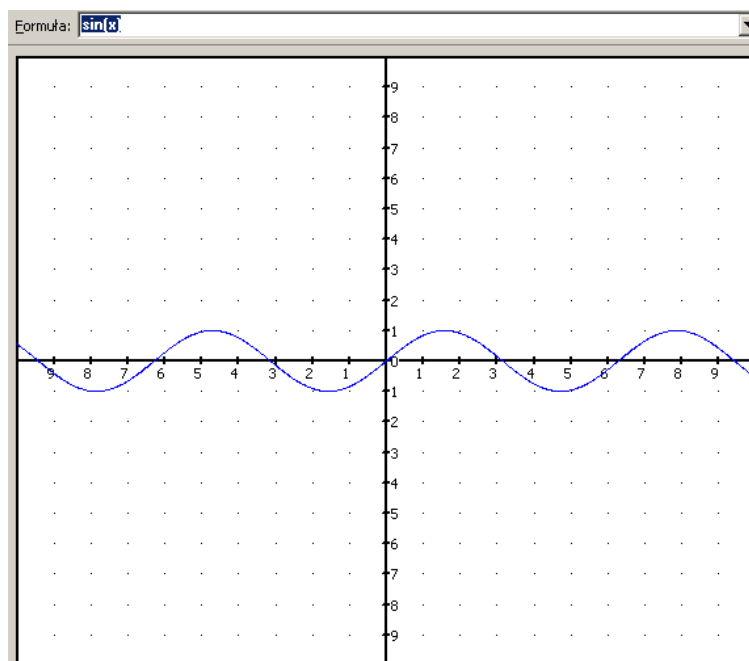
Rysunek 3.2. CoreEditor - Rysowanie wykresów

Wraz z pojawieniem się okna z układem współrzędnych pojawia się również nowy pasek narzędzi oraz nowe menu **Wykres**. Za pośrednictwem elementów znajdujących się w tych miejscach możemy wykonywać takie czynności jak:

- przesuwać wykres,
- pomniejszać i powiększać wykres,
- zapisywać wykres w pliku graficznym,
- drukować wykres,
- zmieniać kolor i styl rysowania linii,
- osadzać dodatkowe obiekty takie jak: odcinki, okręgi czy punkty,
- określać widoczność siatki, układu współrzędnych itp.

Aby narysować wykres funkcji bez jej definiowania *explicite*, należy podać jej definicję bezpośrednio w polu **Formuła**. Prezentuje to rysunek 3.3.

Można również najpierw zdefiniować funkcję w oknie Interpretera CE, a następnie w polu **Formuła** podać jedynie jej nazwę. Można wreszcie



Rysunek 3.3. CoreEditor - Rysowanie wykresów 2

narysować pochodną dowolnej funkcji, poprzedzając jej nazwę literą **d**. Poniżej zaprezentowano listing z definicją funkcji, a na rysunku 3.4 pokazano jak owa funkcja została narysowana.

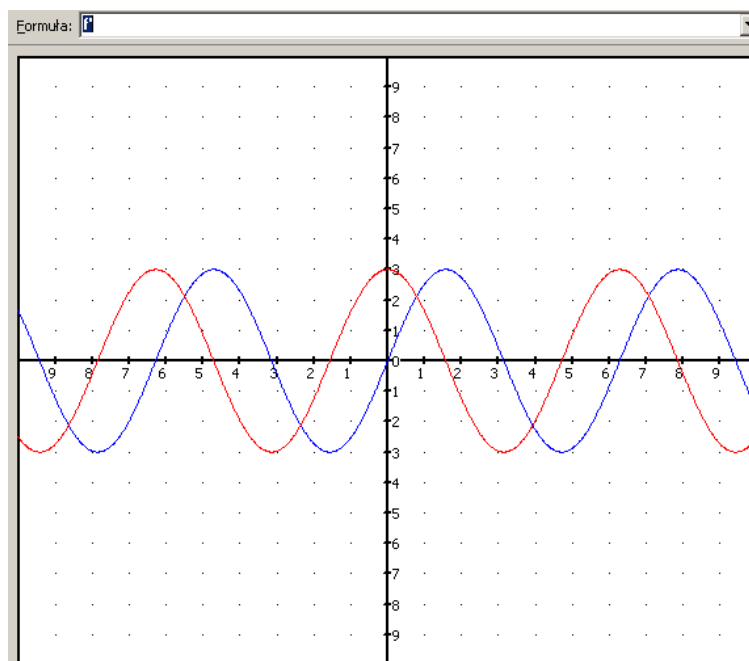
Listing 3.17. Definiowanie funkcji na potrzeby wykresu

```
1 CoreEditor>> f(x):=3*sin(x);
2 +OK [0.000]
```

Mając narysowany wykres dowolnej funkcji, klikając prawym klawiszem myszy na owym wykresie sprawimy, że pojawi się punkt. Przesuwając następnie kursorem, punkt ten będzie wędrował po ostatnio narysowanym wykresie, a w pasku statusu będą się wyświetlały jego dokładne współrzędne.

Na narysowanym wykresie można również wyznaczać obszary, które mają zostać powiększone. Wystarczy kliknąć lewy przycisk myszy, a następnie przesunąć kursorem, rysując w ten sposób prostokątny obszar. Po puszczeniu przycisku CoreEditor przeskaluje narysowane wykresy do nowego obszaru.

Posługując się opisanymi już wcześniej funkcjami `integral` oraz `length`



Rysunek 3.4. CoreEditor - Rysowanie wykresu pochodnej

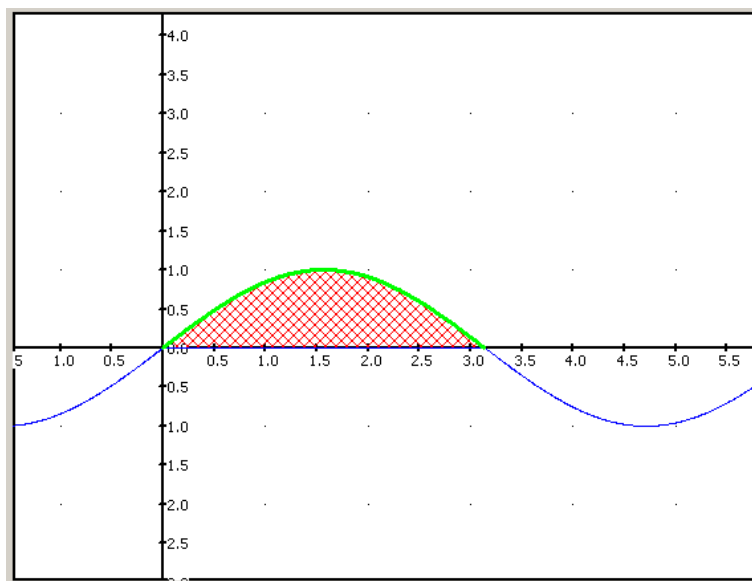
w polu **Formuła** sprawimy, że CoreEditor oprócz obliczenia zadanych parametrów postara się również wyrysować odpowiednie obszary lub łuki na wykresie.

3.13.2. Fraktale

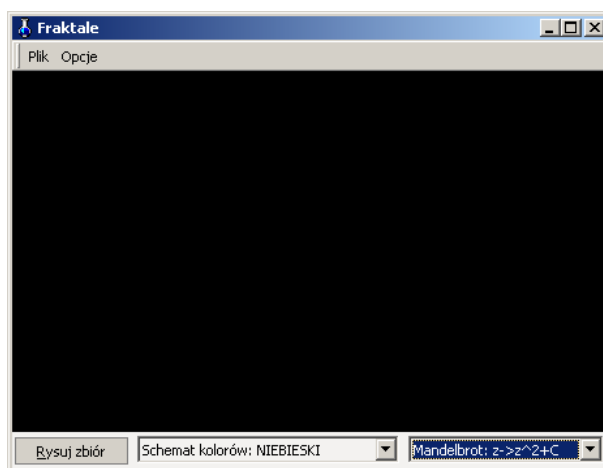
Program CoreEditor został wyposażony również w małą bibliotekę o nazwie CEFnFractals.dll, która wykorzystuje możliwości interpretera CE na potrzeby rysowania różnego rodzaju fraktali. Aby wywołać okno rysowania fraktali, należy wydać komendę `fractals` w oknie interpretera CE. Ukaże się wówczas okno jak na rysunku 3.6.

Okno to składa się z następujących elementów:

- **Menu** - widoczne na samej górze. Zawiera elementy pozwalające m.in. na zapisanie uzyskanego fraktala w pliku graficznym;
- **Część robocza** - na której rysowanie są wszystkie fraktale;
- **Panel narzędzi** - widoczny na samym dole, którego zawartość może się zmieniać w zależności od typu rysowanych obiektów.

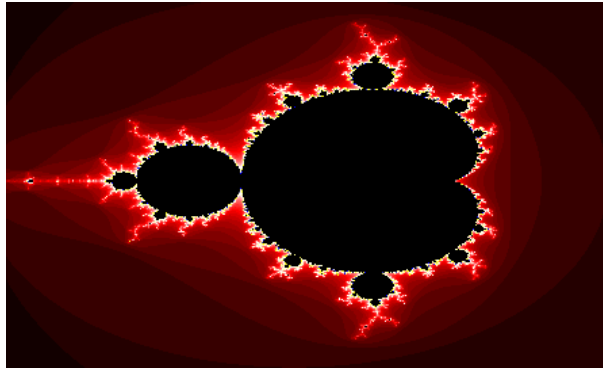


Rysunek 3.5. CoreEditor - Rysowanie całki oznaczonej i łuku

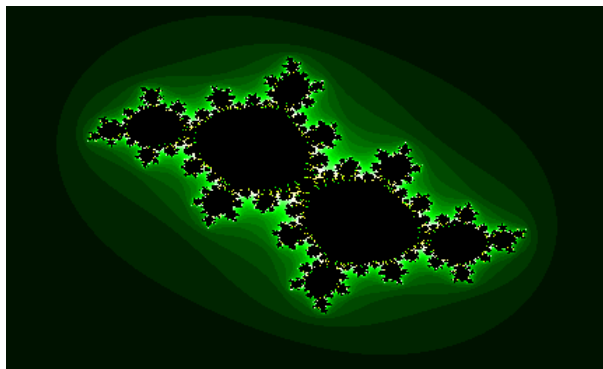


Rysunek 3.6. CoreEditor - Okno rysowania fraktali

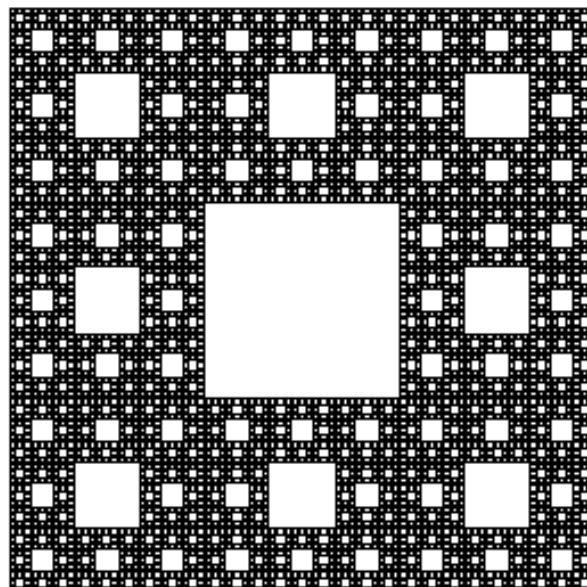
Moduł **Fraktale** pozwala na rysowanie fraktali klasycznych, np.: *Krzywą Kocha*, *Trójkąt* i *Dywan Sierpińskiego*, fraktali wyznaczanych za pomocą przekształceń afinicznych jak również typowych fraktali, jak np. różne wersje zbioru *Mandelbrota*. Moduł pozwala na dowolne kolorowanie uzyskiwanych fraktali, jak również na powiększanie dowolnych ich fragmentów. Na poniższych rysunkach zaprezentowano różne przykładowe obrazy uzyskane za pomocą modułu.



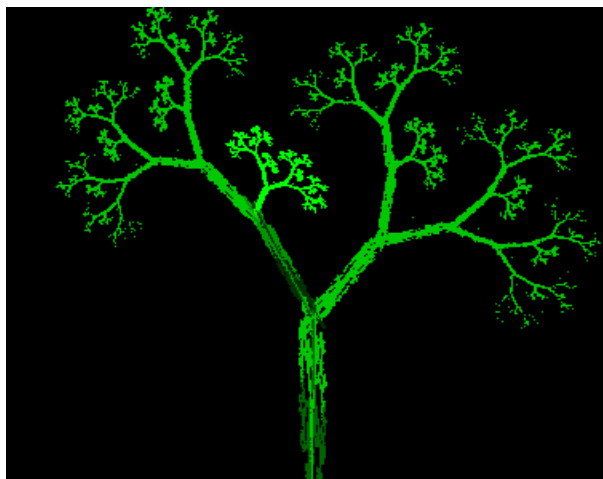
Rysunek 3.7. Fraktale - Mandelbrot: $z = z^2 + C$



Rysunek 3.8. Fraktale - Dysk Siegela



Rysunek 3.9. Fraktale - Dywan Sierpińskiego



Rysunek 3.10. Fraktale - Drzewo - Rysowane na podstawie siedmiu różnych przekształceń afinicznych

Zakończenie

CoreEditor to uniwersalne środowisko do tworzenia stron internetowych, składania dokumentów w \LaTeX -u, ale również do pisania programów w językach Pascal, Asembler, C#, Java czy Nemerle. Posiada liczne systemy ułatwień, które sprawiają, że tworzenie dokumentów jest dużo prostsze. Sięgają po niego zaawansowani programiści PHP, gdyż duża baza gotowych szablonów funkcji znacznie usprawnia im pracę. Sięgają po niego również początkujący, gdyż dzięki różnym mechanizmom ułatwień i systemowi pomocy, program ten dla nich nadaje się najlepiej.

Swoją uniwersalność program zawdzięcza sposobowi w jaki został zaprojektowany. Dzięki interpreterowi CE, umieszczeniu definicji menu i pasków narzędzi w zewnętrznych plikach DLL lub INI, użytkownik zyskał możliwość rozbudowywania programu o dodatkowe narzędzia, kreatory, moduły wspierające nowe języki programowania.

CoreEditor jest jeszcze młodym programem, jednak został zauważony już w świecie. Dzięki angielskiej wersji językowej programu oraz jego strony domowej, sięga po niego coraz więcej osób spoza naszego kraju. Natomiast dzięki licencji edukacyjnej, coraz więcej szkół i uczelni wykorzystuje właśnie ten program, by uczyć przy jego pomocy tworzenia stron WWW.

Prace nad programem nie zostały przerwane. Program stale się rozwija. Poprawiane są w nim znalezione i zgłoszone przez wbudowany system *Raportowania* błędy. Dzięki dodatkowej aplikacji **CEUpdate**, możliwe jest pobieranie z sieci Internet aktualizacji do programu bez konieczności pobierania i instalowania, coraz to większej pełnej wersji instalacyjnej.

Bibliografia

- [1] M. Cantu – *Delphi 6 Praktyka programowania, tom 1*, Mikom, Warszawa 2002.
- [2] M. Cantu – *Delphi 6 Praktyka programowania, tom 2*, Mikom, Warszawa 2002.
- [3] X. Pacheco, S. Teixeira – *Delphi 6 Vademecum profesjonalisty, tom 2*, Helion, Gliwice 2002.
- [4] A. Boduch – *Delphi 2005 Kompendium Programisty*, Helion, Gliwice 2005.
- [5] A. Ralston – *Wstęp do analizy numerycznej*, PWN, Warszawa 1983.
- [6] R.S. Guter, B.K. Owczyński – *Wstęp do metod numerycznych*, PWN, Warszawa 1965.
- [7] B. Baron – *Metody numeryczne w Turbo Pascalu*, Helion, Gliwice 1995.
- [8] B. Baron, A. Pasierbek, M. Maciążek – *Algorytmy numeryczne w Delphi Księga Eksperta*, Helion, Gliwice 2006.
- [9] T. Jurlewicz, Z. Skoczylas – *Algebra liniowa 1, Definicje, twierdzenia, wzory*, Oficyna Wydawnicza GiS, Wrocław 2003.
- [10] M. Gewert, Z. Skoczylas – *Analiza matematyczna, Definicje, twierdzenia, wzory*, Oficyna Wydawnicza GiS, Wrocław 2003.
- [11] W. Krysicki, L. Włodarski – *Analiza matematyczna w zadaniach, cz. 1*, PWN, Warszawa 1999.

- [12] A. Cewe, H. Nahorska, I. Pancer – *Tablice matematyczne*, Podkova, Gdańsk 2004.
- [13] K. Wierzchołowski – *Delphi Samouczek*, Oficyna HELP, Michałowice 2004.
- [14] K. Wierzchołowski – *HTML Samouczek, wyd. 2*, Oficyna HELP, Michałowice 2005.
- [15] D. Oliver – *HTML 4 w 24 godziny*, Infoland, Warszawa 2001.
- [16] B. Pfaffenberger, S.M. Schafer, C. White, B. Karow – *HTML, XHTML i CSS. Biblia*, Helion, Gliwice 2005.
- [17] K. Wierzchołowski – *CSS Kaskadowe Arkusze Stylów*, Oficyna HELP, Michałowice 2004.
- [18] E.A. Meyer – *CSS. Kaskadowe arkusze stylów*, Helion, Gliwice 2002.
- [19] A. Watt, J. Watt, J. Simon, J. O'Donnell – *JavaScript*, Helion, Gliwice 2003.
- [20] Z. Greant, G. Merrall, T. Wilson, B. Michlitsch – *PHP Leksykon funkcji*, Translator s.c., Warszawa 2002.
- [21] L. Welling, L. Thomson – *PHP i MySQL Tworzenie stron internetowych Vademecum profesjonalisty*, Helion, Gliwice 2003.
- [22] L. Atkinson – *MySQL Przewodnik zaawansowanego programisty*, Helion, Gliwice 2003.
- [23] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein – *Wprowadzenie do algorytmów*, PWN, Warszawa 2004.
- [24] N. Wirth – *Algorytmy + Struktury danych = programy*, WNT, Warszawa 1999.

- [25] R. Sedgewick – *Algorytmy w C++*, RM, Warszawa 1999.
- [26] K. Wierzchołowski – *Okiem Matematyka*, Komputer Świat Ekspert Plus, wydanie specjalne Nr 1/2005 s.78.
- [27] H.-O. Peitgen, H.Jurgens, D.Saupe – *Granice chaosu. Fraktale. tom 1*, PWN, Warszawa 2002.
- [28] H.-O. Peitgen, H.Jurgens, D.Saupe – *Granice chaosu. Fraktale. tom 2*, PWN, Warszawa 2002.
- [29] Wikipedia – *Wolna Encyklopedia*, <http://pl.wikipedia.org/>.
- [30] W3C – *World Wide Web Consortium*, <http://www.w3.org/>.
- [31] *Numeryczne obliczanie pochodnych funkcji jednej zmiennej*, http://www.zftik.agh.edu.pl/mof/lab/MOF_A_15.pdf.
- [32] *Specyfikacja języka ECMAScript*, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>.
- [33] *JavaScript 1.5 dokumentacja*, <http://devedge-temp.mozilla.org/library/manuals/2000/javascript/1.5/reference/>.
- [34] *Dokumentacja PHP*, <http://www.php.net/manual/pl/>.
- [35] *Binboy at Sphere*, <http://binboy.sphere.pl>.

Listings

1.1. Obliczanie średniej arytmetycznej	33
3.1. Pierwsza instrukcja	49
3.2. Posługiwanie się wynikami	49
3.3. Posługiwanie się liczbami zespolonymi	50
3.4. Posługiwanie się macierzami	50
3.5. Automatyczne generowanie macierzy	51
3.6. Operatory arytmetyczne	52
3.7. Operatory logiczne	53
3.8. Posługiwanie się zmiennymi	54
3.9. Posługiwanie się funkcjami wieloargumentowymi	55
3.10. Definiowanie własnych funkcji	55
3.11. Definiowanie funkcji wektorowych	55
3.12. Definiowanie funkcji klamrowych	56
3.13. Szukanie miejsc zerowych funkcji	56
3.14. Obliczanie całki oznaczonej i długości łuku	57
3.15. Obliczanie pochodnej funkcji	57
3.16. Interpreter i połączenie z bazą danych	59
3.17. Definiowanie funkcji na potrzeby wykresu	63

Spis tabel

1.1. CoreEditor - Wspierane typy plików	10
1.2. CoreEditor - Menu Plik	17
1.3. CoreEditor - Menu Projekt	21
1.4. CoreEditor - Menu Edycja	23
1.5. CoreEditor - Komendy w trybie VIM	28
2.1. CoreEditor - Wspierane kompilatory	36
3.1. Interpreter CE - Operatory arytmetyczne	52
3.2. Interpreter CE - Operatory logiczne	53
3.3. Interpreter CE - Wbudowane funkcje matematyczne	54

Spis rysunków

1.1. Instalator - Wybór języka	6
1.2. Instalator - Akceptacja warunków licencji	7
1.3. Instalator - Podsumowanie instalacji	8
1.4. CoreEditor - Interfejs programu	9
1.5. CoreEditor - Nowy dokument XHTML	15
1.6. CoreEditor - Szablony dokumentów	20
1.7. CoreEditor - Tworzenie projektu - projekt publikacji w \LaTeX	21
1.8. CoreEditor - Ustawienia programu	29
1.9. CoreEditor - Okno Oblicz wyrażenie	31
2.1. CoreEditor - Paleta kolorów	41
2.2. CoreEditor - Edytor map odsyłaczy	42
2.3. CoreEditor - Edytor Kaskadowych arkuszy stylów	43
2.4. CoreEditor - Narzędzie PHPCrypt	45
2.5. CoreEditor - Narzędzie wstawiania rysunków w języku \LaTeX	47
3.1. CoreEditor - Okno połączenia z bazą danych	61
3.2. CoreEditor - Rysowanie wykresów	62
3.3. CoreEditor - Rysowanie wykresów 2	63
3.4. CoreEditor - Rysowanie wykresu pochodnej	64
3.5. CoreEditor - Rysowanie całki oznaczonej i łuku	65
3.6. CoreEditor - Okno rysowania fraktali	65
3.7. Fraktale - Mandelbrot: $z = z^2 + C$	66
3.8. Fraktale - Dysk Siegela	66
3.9. Fraktale - Dywan Sierpińskiego	66
3.10. Fraktale - Drzewo - Rysowane na podstawie siedmiu róż- nych przekształceń afinicznych	67