

Ćwiczenie 5

Pakiety L^AT_EX i własne makra

5.1. Rozpoczęcie pracy

1. Pobrać archiwum z plikami startowymi zamieszczonymi na stronie z materiałami, rozpakować i otworzyć plik `wprowadzenie.tex` w edytorze np. Visual Studio Code albo VIM. Dokument `fragmenty.tex` zawiera tekst wszystkich listingów niniejszej instrukcji (aby nie kopiować znaków drukarskich z pliku PDF).

2. Zbuduj dokument `wprowadzenie.tex`, aby sprawdzić czy na tym etapie nie występują problemy z kompilacją. W ciągu zajęć ten dokument będzie rozbudowywany o kolejne polecenia czy fragmenty treści.

5.2. Dostosowanie dokumentu do krajowych standardów

3. Dodaj obsługę pakietu `caption`, aby zmodyfikować styl wyświetlania podpisów rysunków do stosowanego w Polsce. Przyjmuje on następujący wygląd: *Rys. 1.1. Logo projektu L^AT_EX*. Należy zwrócić uwagę na występowanie kropki, po numerze porządkowym rysunku i brak kropki na końcu podpisu, ponieważ jest tytułem rysunku. Aby to uzyskać dodaj do nagłówka dokumentu (przed poleceniem `\author`) poniższy fragment

```
1 \usepackage{caption}
2
3 \DeclareCaptionLabelFormat{cfscr}{#1 #2}
4 \DeclareCaptionFont{ftscr}{\fontsize{10}{12pt}\selectfont}
5 \DeclareCaptionLabelSeparator{labscr}{~}
6 \captionsetup[figure]{name={Rys. },labelformat=cfscr,labelfont=ftscr,
7     labelseparator=labscr,textfont=ftscr,position=bottom,skip=3pt}
8
9 \makeatletter
10 \renewcommand\thechapter{\@arabic \c@chapter .}
11 \renewcommand\thefigure{\ifnum \c@chapter >\z@ \thechapter \fi \@arabic \c@figure .}
12 \makeatother
```

4. Zmodyfikuj wyświetlanie stylu podpisów tabeli na następujący *Tab. 1.1. Przykładowa tabela*. Można to zrobić przez dopisanie kolejnej deklaracji podobnej do tej, która zmienia podpisy tabel.

```

1 \captionsetup[table]{name={Tab. },labelformat=cfscr,labelfont=ftscr,labelseparator
   =labscr,textfont=ftscr,position=top,skip=3pt}
2
3 \makeatletter
4 \renewcommand\thetable{\ifnum \c@chapter >\z@ \thechapter \fi \@arabic \c@table .}
5 \makeatother

```

5.3. Definiowane makr

5. Dodaj kropki po numerach punktów 1.1 i podrozdziałów 1.1.1. Tytuły sekcji powinny wyglądać następująco: 1.1. *To jest tytuł sekcji*, a tytuły podsekcji 1.1.1. *To jest tytuł podsekcji*. Aby zrealizować ten punkt, należy uzupełnić nagłówek dokumentu o poniższe deklaracje.

```

1 \makeatletter
2 \renewcommand\thesection{\thechapter \@arabic \c@section .}
3 \renewcommand\thesubsection{\thesection \@arabic \c@subsection .}
4 \makeatother

```

6. Przed poleceniem `\makeatother` zmodyfikuj styl wyświetlania numerów równań przez wpisanie polecenia:

```

1 \renewcommand\theequation{\thechapter \@arabic \c@equation}

```

7. Za pomocą `\newcommand` utwórz nowe polecenie L^AT_EX, gdzie `\kod` będzie zmieniało czcionkę na tę o stałej szerokości znaku tylko dla tekstu przekazanego jako parametr. Następnie dodaj poniższy fragment jako kolejny akapit podrozdziału Listingi:

```

1 W języku C funkcja \kod{main} oznacza miejsce od którego
2 rozpocznie się wykonywanie programu.
3 Wewnątrz niej znajduje się deklaracja zmiennej \kod{i} typu
4 \kod{int}, która będzie sterowała wykonaniem pętli.

```

Obszar wynikowy po skompilowaniu dokumentu `wprowadzenie.tex` powinien wyglądać następująco:

W języku C funkcja `main` oznacza miejsce od którego rozpocznie się wykonywanie programu. Wewnątrz niej znajduje się deklaracja zmiennej `i` typu `int`, która będzie sterowała wykonaniem pętli.

5.4. Pakiety

8. Stosując pakiet `enumitem` zmień globalnie w dokumencie domyślne wypunktowanie z dotychczasowej kropki na pauzę (---) – uzyskiwaną przez potrójny znak minus. Następnie również globalnie skasuj domyślny odstęp pomiędzy wypunktowaniami i wyliczeniami.

9. W punkcie zatytułowanym *Środowiska* dodaj nowe wypunktowanie z kilkoma pozycjami w którym lokalnie zostanie zmieniony znak wypunktowania na gwiazdkę, a wcięcie etykiety wyrażone długością `leftmargin` zostanie ustawione na wartość `2.25\parindent`. **Uwaga!** Tej

możliwości nie stosuje się do zwiększenia wcięć dla zagnieżdżonych wypunktowań gdyż te konfiguruje się parametrem liczbowym w poleceniu `\setlist[itemize,2]`.

10. W dokumencie użyj pakietu `amsmath`, a następnie w sekcji zatytułowanej *Wzory* wyświetl formułę (5.1). **Wskazówka!** W tym celu użyj środowiska `matrix` i operatorów wyrównania `&` i `\` (jak w środowisku tabular).

$$\begin{cases} y &= & 2x &- & 2 \\ x &= & -3y &+ & 2 \end{cases} \quad (5.1)$$

11. Dodany w zadaniu 10 pakiet `amsmath` pozwala na zmianę odwołań do wzorów ze sztywnego (`\ref{eq:wzor}`) wykorzystującego nawiasy okrągłe na dostępne polecenie `\eqref{eq:wzor}`, które będzie można później dowolnie dostosowywać.

12. Pakiet `xcolor` umożliwia definiowanie własnych kolorów używanych w dokumencie. W dokumencie `wprowadzenie.tex` dodaj użycie pakietu w nagłówku i korzystając z polecenia `\DefineNamedColor` zdefiniuj kolor o nazwie `wydzialowy` składający się z komponentów RGB: (75, 160, 165) w skali $0 \div 255$ czyli (0.2941, 0.6274, 0.647) w skali $0 \div 1$.

13. Użyj polecenia `\textcolor` do pokolorowania wymyślnego tekstu dopisanego pod ostatnim równaniem kolorem o nazwie `wydzialowy`. Dopisz kolejne zdanie kolorem o nazwie `red`.

14. Do dokumentu dołącz użycie pakietu `listingsutf8` z parametrem `final`. Następnie w sekcji *Listingi* za pomocą polecenia `\lstinputlisting{plik}` dołącz listing pliku `zrodlo.c`.

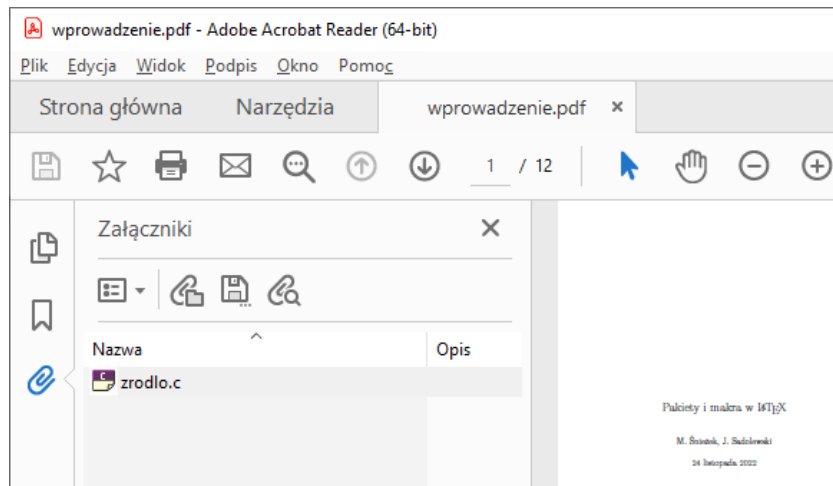
15. Do polecenia `\lstinputlisting` dodaj dwa opcjonalne parametry: `language={ [ANSI]C }`, `input encoding=utf8/cp1250`. Pierwszy z nich dotyczy wybranego zestawu składni dla języka programowania, a drugi wskazuje niezbędną konwersję znaków (zwłaszcza poprawnej interpretacji łańcuchów z polskimi znakami). Następnie w pliku `zrodlo.c` zmień tekst "Hello world" na "Witaj świecie" (gdzie polskie znaki diakrytyczne zakodowane są w UTF-8).

16. Poleceniem `\lstset` w nagłówku dokumentu zmień domyślny format wyświetlania listingów na:

- język składni (`language`) na wartość `{ [ANSI]C }`,
- wyświetlanie spacji (`showspaces`) i tabulatorów (`showtabs`) na wartość `false`,
- ramkę wokół listingu (`frame`) na *maskę bitową* `rightline|single|shadowboxi|leftline|topline|bottomline`,
- wyświetlanie numerów wierszy (`numbers`) po lewej stronie (`left`),
- krok wyświetlania numerów (`stepnumber`) na wartość 1,
- użycie rozszerzonego zakresu znaków (`extendedchars`) na `true`,
- styl podstawowy (`basicstyle`) na `\ttfamily\small`,
- styl słów kluczowych (`keywordstyle`) na `\fontfamily{lmtt}\fontseries{b}\selectfont\small\color{blue}`,
- styl łańcuchów (`stringstyle`) na `\color{olive}`,
- styl komentarzy (`commentstyle`) na `\color{green}\emph`,
- wyświetlanie spacji w łańcuchach (`showstringspaces`) na `false`.

17. Dodaj użycie pakietu `hyperref` umożliwiającego zmianę tytułu, tematu, autora i słów kluczowych w generowanym pliku PDF. Użyj do tego polecenia `\hypersetup`.

18. Dodaj użycie pakietu `embedfile` aby umożliwić tworzenie plików PDF z załącznikami. Następnie w dokumencie poleceniem `\embedfile{plik}` załącz po wydrukowanym listingu kodu języka C jego oryginalną postać wyświetlaną jako odrębny załącznik. **Uwaga!** Nie wszystkie narzędzia do renderowania PDFów wyświetlają załączniki i umieją je wyodrębnić. Stąd wynikowy dokument warto otworzyć w Adobe Reader. Wówczas pojawi się możliwość zapisania załącznika jak na rysunku 5.1.



Rysunek 5.1. Lista załączników w Adobe Reader

5.5. Dodawanie bibliografii

Uwaga! Niestety zainstalowane rozszerzenie do Visual Studio Code nie zawsze radzi sobie z kodowaniem dokumentu w którym stosuje się Bib_TE_Xa. Stąd w przypadku problemów z wyświetlaniem polskich znaków w trakcie prac nad dokumentem będącym przedmiotem ćwiczenia warto spróbować ręcznie skompilować plik korzystając z linii poleceń interpretera Bash, Zsh albo CMD (pod Windows) wpisując:

⇒ `pdflatex wprowadzenie`

⇐ *Długi ciąg tekstu informujący o procesie kompilacji z końcową adnotacją o statusie kompilacji*

⇒ `bibtex wprowadzenie`

⇐ *Kilka linii tekstu o przetwarzaniu dokumentu*

⇒ `pdflatex wprowadzenie`

⇐ *Ponownie raport kompilacji*

⇒ `pdflatex wprowadzenie`

⇐ *Finalny raport kompilacji uruchomionej w celu wstawienia docelowych numerów stron i referencji*

19. L^AT_EX umożliwia sprawne dołączanie bibliografii do dokumentu korzystając z tekstowej bazy danych zapisanej w pliku o rozszerzeniu bib i narzędzia Bib_TE_X. W edytowanym dokumencie znajdź rozdział *Bibliografia* i w miejscu podkreślonego fragmentu *cytowanie* wpisz polecenie `\cite{Doroszewski}` będzie to oznaczało odwołanie do pozycji literatury o kluczu Doroszewski.

20. W tym samym dokumencie przed poleceniem `\listoffigures` wstaw wiersze z poleceniami `\bibliography{zbior}` – zaimportowanie bazy literatury i `\bibliographystyle{plain}` – styl cytowań.

21. Skompiluj dokument i zaobserwuj, że w rozdziale 4 zamiast cytowania pojawia się odwołanie reprezentowane przez [1], a dalej jest nienumerowany dodatek o nazwie *Bibliografia*, gdzie pozycja [1] została uzupełniona o dane bibliograficzne książki.

22. W edytorze Visual Studio Code otwórz plik `zbior.bib` z bazą danych dostępnych elementów bibliografii. Każdy wpis rozpoczyna się znakiem `@`, po którym następuje rodzaj dokumentu np. `BOOK` – książka, `ARTICLE` – artykuł, `InBook` – rozdział w książce, `InProceedings` – materiały konferencyjne, `Manual` – instrukcja. Po rodzaju dokumentu występują objęte nawiasami klamrowymi pary atrybutów i wartości. Pierwszym atrybutem jest klucz – czyli unikatowa nazwa określająca daną pozycję. Ten klucz stosuje się w poleceniu `\cite`, aby uzyskać dany wpis w bibliografii. Następne atrybuty mają też wartości adekwatne do rodzaju dokumentu. Najczęściej są to `Author` – autorzy dokumentu, `Title` – tytuł dokumentu, `Year` – rok. Do tego dokumentu dodaj nowy wpis ze swoją ulubioną książką, albo dowolnej innej publikacji.

23. Użyj cytowania w treści dokumentu pozycji dopisanej w punkcie poprzednim. Również uzupełnij w dowolnie wybranych miejscach cytowania do już istniejących pozycji w bazie i skompiluj dokument `wprowadzenie`.

24. Zmień format wyświetlania cytowań z `plain` na `alpha` i zaobserwuj różnice w uzyskanym dokumencie końcowym.

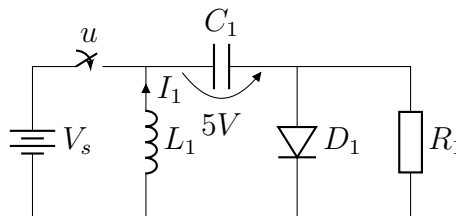
5.6. Zadania dla ambitnych

25. Do dokumentu dodaj pakiet `circuitikz` z opcjami `smartlabels`, `european`. Następnie pod rysunkiem z poprzedniego zadania dodaj nowy akapit o następującej treści: *A teraz czas na obwody elektroniczne*. Po którym należy zamieścić poniższy kod w celu powstania rysunku 5.2:

```

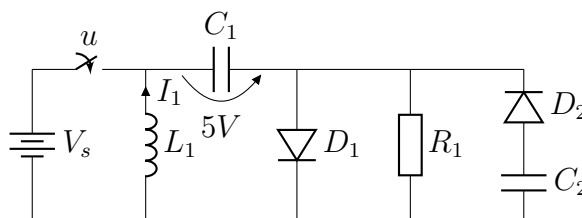
1 \begin{figure}[!htp]
2   \begin{center}
3     \ctikzset{bipoles/length=1cm}
4     \ctikzset{inductors/scale=1, inductor=american}
5     \begin{circuitikz}[scale=1]
6       \draw
7         (0,1) to [cspst=$u$] (1.5,1)
8         (0,-1) node [] {} -- (5,-1)
9         (0,-1) {to [battery, l_=$V_s$,invert] (0,1)}
10        (1.5,1) to [L, l=$L_1$, i<^=$I_1$] (1.5,-1)
11        (1.5,1) to [C, l=$C_1$, v>={{\$5V$}}] (3.5,1)
12        (3.5,1) {to [diode, l=$D_1$] (3.5,-1)}
13        (5,-1) {to [R, l_=$R_1$] (5,1)}
14        (3.5,1) -- (5,1)
15      ;
16    \end{circuitikz}
17  \end{center}
18 \end{figure}

```



Rysunek 5.2. Obwód elektroniczny z zadania 25

Do rysunku 5.2 dodaj diodę D_2 skierowaną przeciwnie do diody D_1 połączoną szeregowo z nowym kondensatorem C_2 za rezystorem R_1 aby otrzymać taki wynik jak na rysunku 5.3. **Wskazówka!** Popatrz na wirtualne koordynaty przekazane w nawiasach okrągłych odnoszące się do kartezjańskiego układu współrzędnych.



Rysunek 5.3. Obwód elektroniczny z zadania 25

26. Utwórz graf przedstawiony przez prowadzącego na zajęciach.

5.7. Zakończenie ćwiczenia

27. Zarchiwizować we własnym zakresie otrzymane pliki, a następnie usunąć je z dysku komputera będącego stanowiskiem laboratoryjnym w celu dostarczenia czystego środowiska pracy dla kolejnej grupy.