ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ

към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

ДИПЛОМНА РАБОТА

по професия код 481020 „Системен програмист”
специалност код 4810201 „Системно програмиране”

Тема: **Шаблон за дипломна работа**

Дипломант:
Име, Презиме, Фамилия

Дипломен ръководител:
титли, Име, Фамилия

СОФИЯ
2024

ТУ-София

Факултет по Електронна Техника и Технологии

Катедра Силова Електроника

Практикум по Приложение на Графични Програмни Среди (ППГПС)

Задание за

КУРСОВА РАБОТА

Вариант 44 – Изправители

Задал: гл. ас. д-р инж. Теодора Тодорова

Студент: Владимир Гаристов

Фак.№: 101218008

Дата на задаване: **21.10.2021 г.** преподавател: …….

Краен срок за предаване: **22.12.2021 г.** студент: …….

Дефиниция на проблема:

Посредством Matlab/Simulink да се проектира еднофазен еднополупериоден неуправляем токоизправител с активен характер на товара, работещ при следните условия:

- Входно напрежение: 48 V_{rms} +/-10%, 50 Hz;
- Товарен ток: 20 A_{rms}.

Задачи за изпълнение:

1. Да се симулира схемата посредством Simulink.
2. Да се запишат уравненията за работата на изправителя, както и условието за край на провеждане на диода (той да се приеме за идеален).
3. Посредством скрипт или функция на Matlab да се изчислят отделните коефициенти на преобразуването на Фурие за изходното напрежение.
4. Да се построят графики на изходното напрежение при използване на 3, 5 и 10 члена на получената сума на Фурие.

Увод

В увода трябва да се изложи кратко въведение в областта *(максимум 1-2 страници)*.

В края на увода трябва да бъдат поставени целите и задачите на дипломната работа.

1 Методи и технологии за реализация на WEB приложения

Първата глава представлява проучвателната част на дипломната работа. Прави се преглед на съществуващи подобни програмни системи и продукти и преглед на известните развойни средства и среди *(максимум 8-10 страници)*.

1.1 Основни принципи, технологии и развойни среди за реализиране WEB приложения

1.2 Съществуващи решения и реализации

2

Проектиране на структурата на WEB базиран електронен магазин

3 Програмна реализация на WEB базиран електронен магазин

Същинската част на ДР с най - голям обем. Да включва описание на начина на реализация на алгоритъма или сценария на мултимедийната презентация или Web страница екрани, прозорци, фрагменти от сорс кода със съответни коментарии *(максимум 10-15 страници)*.

4 Ръководство за потребителя

Представлява Ръководство на потребителя. Трябва да се опише програмния продукт от инсталацията и изискванията към компютърната конфигурация, да се опише въвеждане, редактиране на информацията, съхранение, архивиране, извеждане на справки, информация със съответни екрани, прозорци, обяснителен текст, фигури и т.н *(максимум 6-10 страници)*.

Заключение

Заключението включва обобщение на постиженията в дипломната работа и виждане за усъвършенстване и обогатяване и бъдещо развитие на разработката *(1 страница)*.

Библиография

[1] G. Orwell, “Politics and the English language,” в *The collected essays, journalism and letters of George Orwell*, Harcourt, Brace, Javanovich, 1968, с. 127—140 (цитирано на стр. 21, 22).

[2] S. Peyton Jones, “How to Write a Great Research Paper,” в *2017 Imperial College Computing Student Workshop, ICCSW 2017, September 26-27, 2017, London, UK*, 2017, 1:1—1:1 (цитирано на стр. 21, 22).

[3] S. Pinker, *The sense of style: The thinking person’s guide to writing in the 21st century*. Penguin Books, 2015 (цитирано на стр. 21, 22).

[4] W. Strunk и E. Whyte, *The Elements of style*. Penguin, 2007 (цитирано на стр. 21).

[5] UNSW, *How Do I Cite Online Sources?* `https://student.unsw.edu.au/how-do-i-cite-electronic-sources`, Last accessed: 2019-02-27, 2019 (цитирано на стр. 22).

[6] K. Vonnegut, *How to write with style*. International Paper Company, 1980 (цитирано на стр. 21, 22).

[7] J. M. Williams и J. Bizup, *Style: the basics of clarity and grace*. Pearson Longman, 2009 (цитирано на стр. 21, 22).

[8] О. Пищиков, *Речнико на Баце*. Пустиняк, 2005.

[9] К. В. Романович. “К155 (555) Состав серий ТТЛ ИС и их функциональные аналоги в сериях SN74…/SN54…(средней интеграции) - справочник по микросхемам ТТЛ серий.” (2000), url: `https://www.qrz.ru/reference/kozak/ttl/ttlh06.shtml` (дата на посещ. 21 юни 2020).

Съдържание

Списък на фигурите

Списък на таблиците

Разпечатка на сорс кода на програмата. Ако е твърде голям, да се приложи на CD ROM

ЗАДЪЛЖИТЕЛНО!

Всички работни файлове (фигури, алгоритми, графики, таблици, чертежи), както и текста на дипломната работа трябва да се приложат в отделна папка. Сорс кодът и работоспособен изпълним файл на програмната система трябва да се приложат в отделна папка на CD ROM, който е надписан с имената и випуска на дипломанта и поставени в джоб на задната корица на подвързаната ДР.

Примери за използване на шаблона

A

A.1 Примерен дълъг текст

В съвременния свят на технологиите и дигиталната трансформация, разработването на софтуерни системи се превръща в изключително важна и комплексна задача. Програмните продукти трябва да отговарят на все по-високи изисквания за производителност, сигурност и надеждност, като същевременно предоставят интуитивен и удобен потребителски интерфейс.

Разработката на големи софтуерни системи изисква задълбочено познаване на множество технологии, програмни езици и методологии за разработка. Съвременните разработчици трябва да са запознати с различни архитектурни подходи, шаблони за дизайн и добри практики при писането на код. Освен това, те трябва да могат да работят ефективно в екип, да използват системи за контрол на версиите и да следват установените процеси за разработка на софтуер.

Един от ключовите аспекти при създаването на качествен софтуер е правилното планиране и проектиране на системата. Това включва внимателен анализ на изискванията, създаване на детайлна архитектура и избор на подходящи технологии. Важно е също така да се предвидят възможните проблеми и предизвикателства, които могат да възникнат по време на разработката.

Тестването на софтуера е друг критичен елемент от процеса на разработка. То трябва да бъде извършвано на различни нива - от unit тестове до интеграционни и системни тестове. Автоматизацията на тестването е особено важна при големи проекти, тъй като позволява бързо откриване на проблеми и осигурява по-висока надеждност на крайния продукт.

Сигурността на софтуерните системи е тема, която придобива все по-голямо значение. Разработчиците трябва да са наясно с потенциалните заплахи и да прилагат подходящи мерки за защита. Това включва защита от SQL инжекции, XSS атаки, правилно управление на потребителските данни и криптиране на чувствителна информация.

Оптимизацията на производителността е друг важен аспект при разработката на софтуер. Това включва ефективно използване на системните ресурси, оптимизация на заявките към базата данни, кеширане на често използвани данни и минимизиране на мрежовия трафик. При уеб приложенията особено важно е да се осигури добра производителност при голям брой едновременни потребители.

Документацията на софтуера е често пренебрегван, но изключително важен елемент. Добрата документация улеснява поддръжката и развитието на системата, помага при въвеждането на нови членове в екипа и служи като референция при

възникване на проблеми. Тя трябва да включва както техническа документация за разработчици, така и ръководства за потребители.

Непрекъснатата интеграция и доставка (CI/CD) са станали неразделна част от съвременната софтуерна разработка. Те позволяват автоматизиране на процесите по изграждане, тестване и внедряване на софтуера, което води до по-бързо разработване и по-надеждни релийзи.

Поддръжката на софтуера след неговото внедряване е също толкова важна, колкото и самата разработка. Това включва отстраняване на открити бъгове, добавяне на нови функционалности и оптимизации базирани на обратна връзка от потребителите. Важно е да се поддържа баланс между добавянето на нови функции и запазването на стабилността на системата.

Управлението на техническия дълг е друго предизвикателство, с което се сблъскват разработчиците. Понякога, поради времеви ограничения или други фактори, се налага да се правят компромиси с качеството на кода. Важно е тези компромиси да бъдат документирани и планирани за бъдещо оптимизиране.

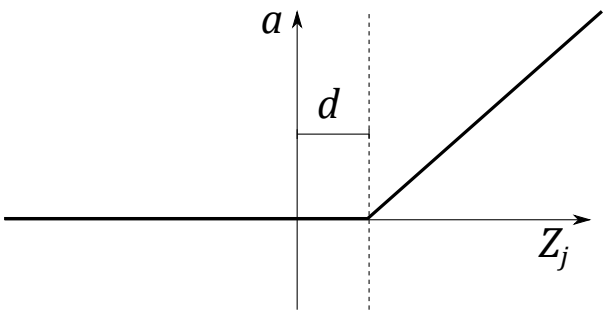
В заключение, разработката на софтуер е комплексна дейност, която изисква широк спектър от знания и умения. Успешните проекти са резултат от добро планиране, правилен избор на технологии, ефективна работа в екип и стриктно следване на установените добри практики в софтуерното инженерство.

A.2 Общи насоки за техническо писане

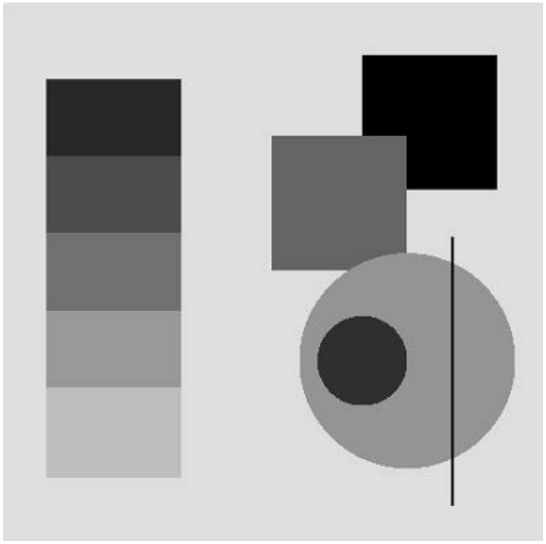
Тези точки се отнасят за цялата дипломна работа, не само за тази глава.

A.2.1 Фигури

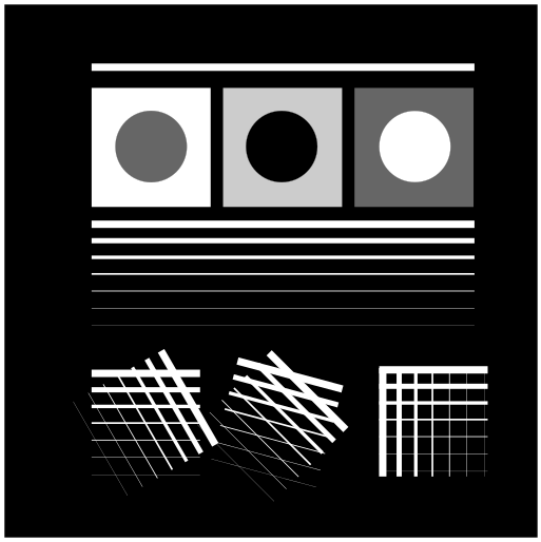
Винаги се позовавайте на включените фигури, като Фигура фиг. A.1, в основния текст. Включвайте пълни, обяснителни надписи и се уверете, че фигурите изглеждат добре на страницата. Можете да включите множество фигури в един float, както във Фигура фиг. A.2, използвайки `subcaption`, който е активиран в шаблона.



Фигура А.1.: В надписите на фигурите обяснете какво вижда читателят: „Схема на изправящата линейна единица, където a е изходната амплитуда, d е конфигурируема мъртва зона, а Z_j е входният сигнал”, както и защо читателят гледа това: „Забележително е, че няма никаква активация *изобщо* под 0, което обяснява първоначалните ни резултати.” **Използвайте векторни формати на изображения (.pdf) където е възможно**. Оразмерявайте фигурите подходящо и не ги правете прекалено големи или твърде малки за четене.



(а) Синтетично изображение, черно на бяло.



(б) Синтетично изображение, бяло на черно.

Фигура А.2.: Синтетични тестови изображения за алгоритми за откриване на ръбове. (а) показва различни нива на сиво, които изискват адаптивен алгоритъм. (б) показва по-предизвикателни тестове за откриване на ръбове, които имат пресичащи се линии. Сливането им в пълни сегменти обикновено изисква алгоритми като трансформацията на Хаф. Това е пример за използване на подфигури, с `subrefs` в надписа.

A.2.2 Уравнения

Уравненията трябва да бъдат набрани правилно и прецизно. Уверете се, че размерът на скобите е правилен и пунктуацията на уравненията е правилна (запетаята е важна и отива *вътре* в блока на уравнението). Обяснете ясно всички използвани символи, ако не са дефинирани по-рано.

Например, бихме могли да дефинираме:

$$\hat{f}(\xi) = \frac{1}{2}\left[\int_{-\infty}^{\infty} f(x) e^{2\pi i x \xi}\right], \tag{A.1}$$

където $\hat{f}(\xi)$ е Фурие трансформацията на времевия сигнал $f(x)$.

A.2.3 Алгоритми

Алгоритмите могат да бъдат набрани с помощта на `algorithm2e`, както в Алгоритъм алгоритъм 1.

Data: $f_X(x)$, a probability density function returing the density at x.
σ a standard deviation specifying the spread of the proposal distribution.
x_0, an initial starting condition.
Result: $s = [x_1, x_2, \ldots, x_n]$, n samples approximately drawn from a distribution with PDF $f_X(x)$.
begin
    $s \longleftarrow []$
    $p \longleftarrow f_X(x)$
    $i \longleftarrow 0$
    while $i < n$ **do**
        $x' \longleftarrow \mathcal{N}(x, \sigma^2)$
        $p' \longleftarrow f_X(x')$
        $a \longleftarrow \frac{p'}{p}$
        $r \longleftarrow U(0, 1)$
        if $r < a$ **then**
            $x \longleftarrow x'$
            $p \longleftarrow f_X(x)$
            $i \longleftarrow i + 1$
            append x to s
        end
    end
end

Algorithm 1: Алгоритъмът Metropolis-Hastings MCMC за извличане на проби от произволни вероятностни разпределения, специализиран за нормални предложени разпределения $q(x'|x) = \mathcal{N}(x, \sigma^2)$. Симетрията на нормалното разпределение означава, че правилото за приемане приема опростената форма.

A.2.4 Таблици

Ако трябва да включите таблици, като Таблица табл. A.1, използвайте инструмент като
https://www.tablesgenerator.com/ за генериране на таблицата, тъй като е изключително досадно в противен случай.

Таблица A.1.: Стандартната таблица с оператори в Python, заедно с техните функционални еквиваленти от пакета `operator`. Забележете, че надписите на таблиците отиват над таблицата, а не под нея. Не добавяйте допълнителни линии/черти към таблиците.

Операция	**Синтаксис**	**Функция**
Addition	`a + b`	`add(a, b)`
Concatenation	`seq1 + seq2`	`concat(seq1, seq2)`
Containment Test	`obj in seq`	`contains(seq, obj)`
Division	`a / b`	`div(a, b)`
Division	`a / b`	`truediv(a, b)`
Division	`a // b`	`floordiv(a, b)`
Bitwise And	`a & b`	`and_(a, b)`
Bitwise Exclusive Or	`a ^b`	`xor(a, b)`
Bitwise Inversion	`~a`	`invert(a)`
Bitwise Or	`a | b`	`or_(a, b)`
Exponentiation	`a ** b`	`pow(a, b)`
Identity	`a is b`	`is_(a, b)`
Identity	`a is not b`	`is_not(a, b)`
Indexed Assignment	`obj[k] = v`	`setitem(obj, k, v)`
Indexed Deletion	`del obj[k]`	`delitem(obj, k)`
Indexing	`obj[k]`	`getitem(obj, k)`
Left Shift	`a <<b`	`lshift(a, b)`
Modulo	`a % b`	`mod(a, b)`
Multiplication	`a * b`	`mul(a, b)`
Negation (Arithmetic)	`- a`	`neg(a)`
Negation (Logical)	`not a`	`not_(a)`
Positive	`+ a`	`pos(a)`
Right Shift	`a >>b`	`rshift(a, b)`
Sequence Repetition	`seq * i`	`repeat(seq, i)`
Slice Assignment	`seq[i:j] = values`	`setitem(seq, slice(i, j), values)`
Slice Deletion	`del seq[i:j]`	`delitem(seq, slice(i, j))`
Slicing	`seq[i:j]`	`getitem(seq, slice(i, j))`
String Formatting	`s % obj`	`mod(s, obj)`
Subtraction	`a - b`	`sub(a, b)`
Truth Test	`obj`	`truth(obj)`
Ordering	`a <b`	`lt(a, b)`
Ordering	`a <= b`	`le(a, b)`

```python
def create_callahan_table(rule="b3s23"):
   """Generate the lookup table for the cells."""
   s_table = np.zeros((16, 16, 16, 16), dtype=np.uint8)
   birth, survive = parse_rule(rule)

   # generate all 16 bit strings
   for iv in range(65536):
      bv = [(iv >> z) & 1 for z in range(16)]
      a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p = bv

      # compute next state of the inner 2x2
      nw = apply_rule(f, a, b, c, e, g, i, j, k)
      ne = apply_rule(g, b, c, d, f, h, j, k, l)
      sw = apply_rule(j, e, f, g, i, k, m, n, o)
      se = apply_rule(k, f, g, h, j, l, n, o, p)

      # compute the index of this 4x4
      nw_code = a | (b << 1) | (e << 2) | (f << 3)
      ne_code = c | (d << 1) | (g << 2) | (h << 3)
      sw_code = i | (j << 1) | (m << 2) | (n << 3)
      se_code = k | (l << 1) | (o << 2) | (p << 3)

      # compute the state for the 2x2
      next_code = nw | (ne << 1) | (sw << 2) | (se << 3)

      # get the 4x4 index, and write into the table
      s_table[nw_code, ne_code, sw_code, se_code] = next_code

   return s_table
```

Програмен код А.1: The algorithm for packing the 3×3 outer-totalistic binary CA successor rule into a $16 \times 16 \times 16 \times 16$ 4 bit lookup table, running an equivalent, notionally 16-state 2×2 CA.

А.2.5 Код

Избягвайте да поставяте големи блокове код в отчета (повече от страница в един блок, например). Използвайте оцветяване на синтаксиса, ако е възможно, както в Листинг прогр. код А.1.

А.2.6 Доказателства

Уверете се, че представяте доказателствата си добре. Използвайте подходящи визуализации, техники за докладване и статистически анализ, според случая. Целта не е да изсипете всички данни, които имате, а да представите добре подкрепен с доказателства аргумент.

Ако използвате числови доказателства, посочете разумен брой значещи цифри; не казвайте „18.41141% от потребителите са били успешни”, ако сте имали само 20 потребители. Ако усреднявате *каквото и да е*, представете както мярка за централна тенденция (напр. средна стойност, медиана), *така и* мярка за разпръскване (напр. стандартно отклонение, мин/макс, интерквартилен диапазон).

Можете да използвате `siunitx` за дефиниране на единици, подреждане на числата спретнато и задаване на прецизност за целия LaTeX документ.

Например, тези числа ще се появят с два десетични знака: 3.14, 2.72, а това ще се появи с разумно разстояние 1 000 000.00.

Ако използвате статистически процедури, уверете се, че разбирате процеса, който използвате, и че проверявате дали необходимите предположения са валидни във вашия случай.

Ако визуализирате, следвайте основните правила, както е илюстрирано във Фигура фиг. A.3:

- Надпишете всичко правилно (оси, заглавие, единици).
- Надпишете подробно.
- Цитирайте в текста.
- **Включете подходящо показване на несигурността (напр. ленти на грешката, Box plot)**
- Минимизирайте безпорядъка.

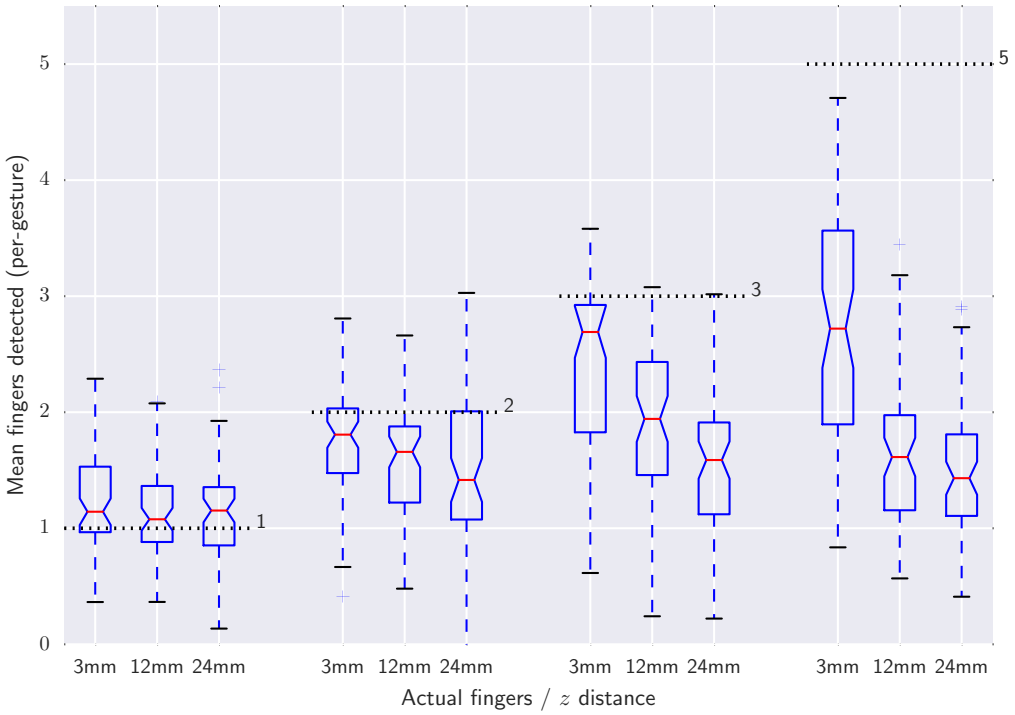
Вижте файла `guide_to_visualising.pdf` за допълнителна информация и насоки.

A.2.7 TODO-та

TODO: Напиши цялата документация на дипломната работа

A.2.8 Цитати

> *Абсолютно всички цитати на известни личности в Интернет са 100% верни.*
>
> — **Алберт Айнщайн**
> (ТУЕС-ар)



Фигура А.3.: Среден брой пръсти, открити от сензора за докосване на различни височини над повърхността, усреднено за всички жестове. Пунктираните линии показват истинския брой присъстващи пръсти. Box plots включват bootstrapped несигурност за медианата. Ясно е, че устройството е склонно към подценяване на броя на пръстите, особено при по-високи z разстояния.

А.2.9 Източници и ръководства за стил

Има много ръководства за добро писане на български и английски език. Не е задължително да ги прочетете, но те ще подобрят начина, по който пишете.

- *Как да напишем страхотна научна статия* [2] (**препоръчително**, въпреки че не пишете научна статия)
- *Как да пишем със стил* [6]. Кратко и лесно за четене. Достъпно онлайн.
- *Стил: Основите на яснотата и изяществото* [7] Много популярно съвременно ръководство за английски стил.
- *Политика и английският език* [1] Известно есе за ефективно и ясно писане на английски.
- *Елементите на стила* [4] Остаряло и американско, но класическо.
- *Усетът за стил* [3] Отлично, макар и доста задълбочено.

Стилове на цитиране

- Ако се позовавате на източник като съществително, цитирайте го така: „Orwell [1] разглежда ролята на езика в политическата мисъл.”
- Ако се позовавате имплицитно на източници, използвайте: „Има много добри книги за писане [1], [3], [7].”

Пълно ръководство за добри практики при цитиране от Питър Кокс-хед е достъпно тук: `http://www.cs.bham.ac.uk/~pxc/refs/index.html`. Ако не сте сигурни как да цитирате онлайн източници, вижте UNSW [5]. [1]

A.2.10 Предупреждение за плагиатство

ВНИМАНИЕ

Ако включвате материал от други източници без пълно и коректно цитиране, извършвате плагиатство. Наказанията за плагиатство са тежки. Цитирайте всеки включен текст и го посочвайте правилно. Цитирайте ясно всички изображения, фигури и др. в техните описания.

A.2.11 Цитиране на текст

Когато цитирате дълъг пасаж, използвайте средата `quote`:

> Ако драскате мислите си как да е, читателите със сигурност ще почувстват, че изобщо не ви е грижа за тях. Те ще ви определят като егоманиак или глупак - или още по-лошо, ще спрат да ви четат. Най-осъдителното разкритие, което можете да направите за себе си, е че не знаете кое е интересно и кое не е.
>
> [6]

Когато цитирате в текста, както следната забележка на Саймън Пейтън-Джоунс, използвайте кавички „Предаването на интуицията е първостепенно, а не второстепенно” [2].

[1] Понякога е по-подходящо да се посочи онлайн ресурс като `https://developer.android.com/studio` в бележка под линия, отколкото да се включва като формална референция.