МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

***Циклова комісія програмування***

**РЕФЕРАТ**

з курсу «Тренінг-курс з основ програмування»

**з теми «Стилі кодування та написання якісного коду»**

Ткач Анни Богданівни та Чернишова Романа Святославовича

*ПІБ студентів*

студентів групи **1П-21**

Викладач Марченко С. В.

Черкаси-2022

**Зміст**

1.Вступ ---------------------------ст.3

2.Стандарт оформлення коду ---------------------ст.4-6

3.Стилі програмування ----------------------------ст.7-16

4.Джерела --------------------------------------------ст.17

**Вступ**

Відповідаючи на це питання одним рядком, можна сказати, що хороший стиль програмування має на увазі наявність структури у написаної програми. Відповідно, якщо програма містить багато рядків програмного коду в одній процедурі, відділена на смислові блоки, нагадує "спагетті", тобто має безліч переходів управління, містить мало коментарів і, як наслідок, стає заплутаною, громіздкою, важко модифікується і керується, то це і є "поганий стиль програмування".

Стандарт оформлення коду

**Стандарт оформлення коду**

 набір правил та угод, що використовуються при написанні сирцевого коду на деяких мовах програмування**.**

**Використання**

Стандарт оформлення коду зазвичай приймається та використовується групою розробників програмного забезпечення для єдиного стилю оформлення коду, над яким йде спільна праця. Метою прийняття та використання стандарту є спрощення сприйняття програмного коду людиною, мінімізація навантаження на пам'ять та зір при читанні програми.

Зразком для стандарту кодування може стати набір угод, прийнятих в якій-небудь поширеній печатній праці з мови програмування (наприклад, стандарт кодування мовою С, що отримав коротку назву K&R, виходить з класичного описання С його творцями — Керніганом та Рітчі), поширена бібліотека або API (так, на поширення угорської нотації вплинув той факт, що її використовували в MS-DOS та Windows API, а більшість стандартів кодування для Delphi використовують манеру кодування бібліотеки VCL). Рідше розробник мови програмування випускає детальні рекомендації по кодуванню на ній; випущені, наприклад, стандарти кодування на C# від Microsoft та на Java від Sun. Запропонована розробником та прийнята в загально відомих джерелах манера кодування може бути доповнена та уточнена у корпоративних стандартах.

**Склад**

Стандарт дуже залежить від використовуваної мови програмування. Наприклад, стандарт оформлення коду для [С](https://uk.wikipedia.org/wiki/C_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) буде сильно відрізнятися від стандарту для мови [BASIC](https://uk.wikipedia.org/wiki/BASIC). Зазвичай метою стандарту є досягнення такого стану, коли [програміст](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%96%D1%81%D1%82) достатньої кваліфікації міг би дати висновки про функцію, яку виконує конкретна ділянка коду, а в ідеалі — також визначити його коректність, вивчивши тільки цю ділянку коду, або мінімально вивчивши інші частини програми. Іншими словами, сенс коду повинен буди зрозумілим з самого коду без необхідності вивчати контекст. Тому стандарти кодування будуються так, щоб шляхом визначеного візуального оформлення елементів програми збільшити інформативність коду для людини.

Зазвичай, стандарт оформлення коду описує:

* засоби вибору значень та використовуваний регістр символів для імен [змінних](https://uk.wikipedia.org/wiki/%D0%97%D0%BC%D1%96%D0%BD%D0%BD%D0%B0) та інших ідентифікаторів.
  + запис [типу](https://uk.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) змінної в її ідентифікаторі ([угорська нотація](https://uk.wikipedia.org/wiki/%D0%A3%D0%B3%D0%BE%D1%80%D1%81%D1%8C%D0%BA%D0%B0_%D0%BD%D0%BE%D1%82%D0%B0%D1%86%D1%96%D1%8F" \o "Угорська нотація)).
  + регістр символів (нижній, верхній, «верблюжий», «верблюжий» з малої букви), використання знаку підкреслення для розділу слів.
* стиль відступів при оформленні логічних блоків — чи використовуються [табуляції](https://uk.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D1%83%D0%BB%D1%8F%D1%86%D1%96%D1%8F), ширина відступу.
* спосіб розстановки дужок, що обмежують логічні блоки.
* використання пробілів при оформленні логічних та [арифметичних](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D0%BA%D0%B0) виразів.
* стиль коментарів та використання документуючих коментарів.

Поза стандартом існують правила про:

* відсутність магічних чисел
* обмеження розміру коду по горизонталі на вертикалі.

## Стандарти кодування і синтаксис мов програмування

Основні принципи поширених стандартів кодування в останній час впливають на синтаксис нових мов програмування. В деяких з них, угоди, що раніше використовувались тільки в стандартах кодування, стають обов'язковими елементами синтаксису. Так, у деяких сучасних мовах ([Python](https://uk.wikipedia.org/wiki/Python), [Nemerle](https://uk.wikipedia.org/wiki/Nemerle" \o "Nemerle)) відступи впливають на логіку виконання (тобто блоки коду виділяються не ключовими словами, а розміром відступів). В інших стали частиною мови угоди про регістри літер та префіксів для типів, [констант](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D0%B0%D0%BD%D1%82%D0%B0), змінних та полів [класів](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)).

Якщо раніше недисциплінований програміст міг ігнорувати стандарти кодування з особистих міркувань, заради зручності та швидкості написання коду, то тепер, при роботі з новими мовами, дотримання стандартів деякою мірою контролюється [транслятором](https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%82%D0%BE%D1%80)