Правила и конвенции за именуване

В този документ са описани правилата и конвенциите за именуване на класове, методи и променливи (*а и не само*), които трябва да се спазват, докато реализираме Java-базиран софтуер.

Защо имената имат значение?

Писането на код е творчество. То е като писането на литературни произведения:

- в криминалните романи трябва да има мистерия, драма и напрежение;
- в публицистиката са възможни гръмки и преувеличени фрази, целящи опетняване на конкуренцията или всяване на ненужна паника;

Всеки един литературен жанр си има специфика и особености. Ако сте читатели на някой от изброените, ще знаете точно какво да очаквате. В литературата можем да си служим с думи, а в програмите си служим с конструкции, променливи и изрази.

Единственият начин да бъдем експресивни е да спазваме следните правила:

- пишем код за хора, а не за компютри – всичко трябва да е четимо и ясно;
- всяко едно име, трябва да описва своята история, предназначение и отношение спрямо заобикалящата го среда;
- трябва да спазваме конвенциите на езика, това са правила, които са дефинирани, за да направят работата и поддръжката на софтуера по-лесна и удобна.

Често ще споменаваме един от следните три подхода за именуване:

- **PascalCase** – при този подход, при съставяне на име, всяка една от думичките (*смисловите единици*) в него, трябва да започва с ***главна буква***.

```
Word

MultyWord

ComplexMultyWordName
```

- **camelCase** – при този подход, при съставяне на име, първата думичка (*смисловите единици*) трябва да ***започва с малка буква***, а всички останали са с водеща главна буква.

```
word

multyWord

complexMultyWordName
```

- **snake_case** – при този подход, при съставяне на име, всяка една от думичките (*смисловите единици*) в него, трябва да се изписва с малки букви, разделени с подчертаващо тире (*underscore*).

```
word

multy_word

complex_multy_word_name
```

Как да си именуваме проектите?

Имената на проектите, които създавате в IntelliJ IDEA (*или друга среда за разработка*), отразяват идеята, която трябва да разработите. Името на проекта и името на класовете **не** трябва да съвпадат. Разглеждайте имената на проектите като механизъм за комуникация, които да съдържа в себе си същността на софтуера, който разработвате.

ПРАВИЛО

> За именуване на проектите силно препоръчваме да се придържате към използването на **PascalCase** конвенция, но се допуска и известна креативна свобода.

Как да си именуваме класовете?

- При именуване на класове, те задължително трябва да са съществителни имена, назоваващи обекти от реалния свят или абстрактни концепции;
- При именуване на класове задължително използвайте **PascalCase** конвенция.

Как да си именуваме променливите?

Променливите са основният контейнер за съхранение и пренос на информация в една програма. Важно е да усвоите изкуството на правилното именуване.

- При именуване на променливи **задължително** използвайте **camelCase** конвенция.

Основните характеристики на доброто име на променлива са:

- отразява функционалната идея на информацията, която съхранява;
- дава информация за типа, към който принадлежи.

При именуване на променливи суфиксите са много полезни:

- ако променливата съдържа цели числа за бройка, може да ползвате суфикс “**Count**”;

- ако променливата съдържа сума (*цена*) можете да ползвате суфикса “**Amount**”;
- ако променливата съдържа **низ**, който може да бъде назован със събирателен термин, то използвайте го като суфикс. Например, заглавие “**Title**” или съобщение “**Message**”;
- ако променливата съдържа масив, можете да използвате като суфикс генералното понятие “**Collection**”, за да покажете че говорим за набор от данни, а не единична такава.

Никога **не** пишете имена на променливи:

- на шльокавица или език различен от английски;
- ползвайки съкращения, които **не** са общоизвестни в индустрията или езика;
- с помощта на генерални/общи понятия, които нямат никакъв смисъл като *test/variable*;
- съдържащи една буква.

Константи и магически символи?

При именуване на константи, идентификаторите се кръщават с главни букви и се разделят с подчертаващо тире. Например: *PURCHASE_AMOUNT_LIMIT*

Константите са механизъм за справяне с така наречените **магически символи**. Магически символ наричаме всяка употреба на число или низ, в рамките на кода, чието значение не е очевидно.

Например да разгледаме следния израз.

```
boolean isCustomerVip = purchessAmount >= 5000;
```

Числовият литерал “**5000**” е магическото число в този контекст. Добре е да се избягва употребата на подобни литерали, защото тяхното значение не е очевидно. Доста по-добър подход е да дефинираме константа, съдържаща специфичната стойност. По този начин ще използваме наименованието на константата вместо нейната стойност.

```
final int PURCHASE_AMOUNT_LIMIT = 5000;
boolean isCustomerVip = purchessAmount >= PURCHASE_AMOUNT_LIMIT;
```