

コーディングの命名規則一覧

By: [murashun](#) Last updated: 2019.10.30

目次

- 1. 真偽値を表す命名規則
- 2. 文字列操作を表す命名規則
- 3. データ操作 (追加) を表す命名規則
- 4. データ操作 (変更) を表す命名規則
- 5. データ操作 (削除) を表す命名規則
- 6. データ操作 (書き込み) を表す命名規則
- 7. データ操作 (読み込み) を表す命名規則
- 8. データ操作 (検証) を表す命名規則
- 9. 許可・禁止を表す命名規則
- 10. ハンガリアン記法
- 11. 命名方式
- 12. 命名時のポイント

section. 1

真偽値を表す命名規則

真偽値を表す命名規則一覧

場所	単語	意味	例
prefix	is	期待する状態になっているか	isEnabled
prefix	can	期待する処理ができるか	canRemove
prefix	should	命令を実行するべきか	shouldMigrate
prefix	need	命令を実行する必要があるか	needFileCopy
prefix	has	期待するデータやプロパティを持っているか	hasConnection
—	exists	期待するデータやプロパティが存在するか	exists(dir)
—	contains	期待するデータやプロパティが含まれているか	contains(item)

section. 2

文字列操作を表す命名規則

文字列操作を表す命名規則一覧

場所	単語	意味	例

prefix	find	情報を検索する (発見可能前提)	findString
prefix	search	情報を探索する (発見不可前提)	searchString
—	seek	連続した情報を順番に探査する	file.seek()
—	extract	情報のある条件で抽出する	hash.extract()
—	filter	情報のある条件で除外する	filter()
—	replace	既存のデータを置き換える	String.replace()
—	join	既存のデータを結合する	String.join()
—	parse	既存のデータを解析する	String.Parse()

section. 3

データ操作 (追加) を表す命名規則

データ操作 (追加) を表す命名規則一覧

場所	単語	意味	例
prefix	set	データを設定する	setProperty
prefix	add	データやオブジェクトを追加する	addList
prefix	put	データやオブジェクトを追加する	hash.put(key,value)
prefix	insert	データやオブジェクトを挿入する	insertQueue
prefix	append	データやオブジェクトを末尾に追加する	appendQueue
prefix	push	データやオブジェクトを先頭に追加する	pushQueue
prefix	prepend	データやオブジェクトを先頭に追加する	prependQueue
prefix	register	データやオブジェクトを登録する	registerStorage
prefix	create	新しいデータやファイルを作る	createAccount
prefix	new	新しいデータやファイルを作る	newAccount
prefix	make	既存データを加工してデータやファイルを作る	makeFile
prefix	build	既存データからデータやファイルを組み立てる	buildFile
prefix	from	既存データを流用してデータやファイルを作る	fromConfigFile
prefix	generate	何かのルールに従ってデータやファイルを作る	generateFile

section. 4

データ操作 (変更) を表す命名規則

データ操作 (変更) を表す命名規則一覧

場所	単語	意味	例
prefix	update	既存のデータを最新化する	updateAccount
prefix	upgrade	既存のデータをより優れたものに交換する	upgradeAccount
prefix	apply	既存のデータを適用する	applyAccount
prefix	refresh	既存のデータを更新する	refreshAccount
prefix	changed	既存のデータを変更する	changedAccount
prefix	modified	既存のデータを修正する	modifiedAccount
prefix	revised	既存のデータを改版する	revisedAccount
prefix	enable	既存のデータを使用可能にする	enableAccount
prefix	disable	既存のデータを使用不可にする	disableAccount
prefix	fix	既存のデータの問題を解決する	fixAccount
prefix	repair	既存のデータを修理する	repairAccount
prefix	restore	既存のデータを復元する	restoreAccount
prefix	recover	既存のデータを復旧する	recoverAccount
prefix	edit	既存のデータを編集する	editAccount
prefix	adjust	既存のデータを調整する	adjustString
prefix	adapt	既存のデータを適合させる	adaptString
prefix	convert	既存のデータを変換する	convertString
prefix	to	既存のデータを変換する	toString

section. 5

データ操作 (削除) を表す命名規則

データ操作 (削除) を表す命名規則一覧

場所	単語	意味	例
prefix	delete	既存のデータを削除する (復元不可)	deleteAccount
prefix	remove	既存のデータを除去する (復元可能)	removeAccount
prefix	trash	既存のデータを廃棄する (復元可能)	trashAccount
prefix	erase	既存のデータを消去する (書き直し可能)	eraseAccount
prefix	clear	既存のデータを初期化する	clearAccount
prefix	flush	既存のデータを初期化する	flushAccount

prefix	reset	既存のデータを初期化する	resetAccount
prefix	dispose	既存のデータを開放する (再利用可能)	disposeAccount
prefix	destroy	既存のデータを破棄する (再利用不可)	destroyAccount
prefix	unregister	登録済みのデータを解除する	unregisterStorage
prefix	unset	定義済みのデータを未定義にする	unsetAccount
prefix	pop	先頭のデータを取り出して取り除く	popQueue
—	initialize	既存のデータを初期化する	initialize()

section. 6

データ操作（書き込み）を表す命名規則

データ操作（書き込み）を表す命名規則一覧

場所	単語	意味	例
prefix	save	既存のデータを保存する	saveAccount
prefix	output	既存のデータを出力する	outputAccount
prefix	export	既存のデータを書き出す	exportAccount
prefix	write	既存のデータを書き込む	writeAccount
prefix	store	既存のデータを貯蔵する	storeAccount
prefix	send	既存のデータを送信する	sendAccount
prefix	commit	既存のデータを確定する	commitAccount

section. 7

データ操作（読み込み）を表す命名規則

データ操作（読み込み）を表す命名規則一覧

場所	単語	意味	例
prefix	get	既存のデータを取得する	getAccount
prefix	load	既存のデータを呼び出す	loadAccount
prefix	input	既存のデータを入力する	inputAccount
prefix	import	既存のデータを読み出す	importAccount
prefix	read	既存のデータを読み込む	readAccount
prefix	restore	既存のデータを復元する	restoreAccount

prefix	fetch	既存のデータを取得する	fetchAccount
--------	-------	-------------	--------------

section. 8

データ操作（検証）を表す命名規則

データ操作（検証）を表す命名規則一覧

場所	単語	意味	例
prefix	check	対象のデータがある条件に適合するか確認する	checkAccount
prefix	test	対象のデータがあるルールを満たすか確認する	testAccount
prefix	validate	対象のデータが正しいか検証する	validateAccount
prefix	compare	対象のデータを比較する	compareAccount
prefix	verify	対象のデータを照合する	verifyAccount

section. 9

許可・禁止を表す命名規則

許可・禁止を表す命名規則一覧

場所	単語	意味	例
prefix	allow	対象に利用権限を与える	allowAccount
prefix	disallow	対象に利用権限を与えない	disallowAccount
prefix	accept	対象を承認する	acceptAccount
prefix	deny	対象を否認する	denyAccount
prefix	refuse	申請や要求を辞退する	refuseAccount
prefix	reject	申請や要求を拒否する	rejectAccount
prefix	grant	対象にある範囲の権限を与える	grantAccount
prefix	revoke	対象から権限を剥奪する	revokeAccount

section. 10

ハンガリアン記法

ハンガリアン記法には、シモニーの考案した変数の意味や使用目的から接頭辞を決定する **アプリケーションハンガリアン** と、変数名にデータ型を表す接頭語や接尾語をつける記法だと誤解された **システムハンガリアン** の

2種類があります。単に「ハンガリアン記法」と書かれている場合、一般的にはシステムハンガリアンの意味で用いられます。

下記は、C 言語などの明示的な型付けを行う言語で用いられるシステムハンガリアン記法の一覧です。システムハンガリアン記法は、静的な型付けを行う言語とは相性が良いですが、動的な型付けを行う言語では使用されません。

システムハンガリアン記法一覧

場所	単語	意味	例
prefix	b or f	論理型 (binary)	bDirtyFlag
prefix	ch	文字型 (character)	chSeparator
prefix	by	byte 型 (符号なし 1 バイト整数)	byGrayLevel
prefix	i or n	整数型 (integer)	nPower
prefix	l	長整数 (long)	lDate
prefix	ui	符号なし整数 (unsigned integer)	uiCount
prefix	w	ワード型 (word)	wLanguageCode
prefix	dw	ダブルワード型 (dword)	dwSize
prefix	f or fp	単精度浮動小数点型 (float)	fPrice
prefix	d or dp	倍精度浮動小数点型 (double)	dPi
prefix	p or lp	ポインタ型 (pointer)	pDirectSound
prefix	s	文字列型 (string)	sPlayerName
prefix	sz	ゼロ終端文字列型 (string zero)	szFileName
prefix	fn	関数ポインタ型 (function pointer)	fnCallback
prefix	h	ハンドル型 (handle)	hThread
prefix	g_	グローバル変数	g_iErrorCode
prefix	c_	定数 (const)	c_nBufferSize
prefix	s_	静的変数 (static)	s_pLookupTable
prefix	m_	メンバー変数	m_nLength

section. 11

命名方式

キャメルケース (ローワーキャメルケース)

複合語の先頭を、小文字で書き始める。

例) `getInputReader`

パスカルケース (アッパーキャメルケース)

複合語の先頭を、大文字で書き始める。

例) `GetInputReader`

スネークケース

アンダースコア (`_`) を区切記号として単語をつなげる。

例) `quoted_printable_encode`

チェインケース

ハイフン (`-`) を区切記号として単語をつなげる。

例) `Get-Process`

section. 12

命名時のポイント

明確な単語を使う

`get` や `size` などの単語は抽象的であるため、「どこから」・「何の」といった明確な情報を命名に含ませる。

汎用的な名前を避ける

- 命名するときは、それらが何の目的を持っているかを考慮する
- 一時的な保管を目的とした変数などは例外として扱う

長い名前を避ける

名前は長くても 20 文字程度に収まるように設計する。長い名前は可読性が落ち、入力が面倒で打ち間違いの原因になる。また、慣例として略語が一般的なものは、その例に従う。(例：`average` は `avg` にするなど)

理解できる名前にする

略語にした場合、他の要素と混同する名前になる場合は、必要以上に省略せず理解できる名前にする。

具体的な名前を使う

その変数や関数があるの情報を持つか、またはどんな処理を行うのかを具体的な名前として含ませる。

対義語を定義する

命名規則に一貫性を持たせるために対義語を正しく定義する。

- `Start` ⇔ `Stop`
- `Top` ⇔ `Bottom`
- `High` ⇔ `Low`
- `Big` ⇔ `Small`
- `Attach` ⇔ `Detach`
- `Input` ⇔ `Output`
- `Show` ⇔ `Hide`

接尾辞・接頭辞を使って情報を追加する

値を保管する変数には単位を入れる。(px や ms など)

