



Имена Променљивих

(њихове скривене моћи)

Бранко Поповић



Именованье?

- Зашто је важно?
- Именујемо променљиве
- Именујемо функције
- Именујемо пројекат
- Именујемо, именујемо, именујемо



Лоше именовање

```
x = x - xx;
```

```
xxx = aretha + SalesTax( aretha );
```

```
x = x + LateFee( x1, x ) + xxx;
```

```
x = x + Interest( x1, x );
```



Добро именовање

```
balance = balance - lastPayment;  
monthlyTotal = NewPurchases + SalesTax( newPurchases );  
balance = balance + LateFee( customerID, balance ) + monthlyTotal;  
balance = balance + Interest( customerID, balance );
```



Основно правило!

- Пун и прецизан опис
- Фраза као назив
- Пример:
 - Број седишта на стадиону:

var brojSedistaNaStadionu = ...



Пример

Purpose of Variable	Good Names, Good Descriptors	Bad Names, Poor Descriptors
Running total of checks written to date	<i>runningTotal, checkTotal, nChecks</i>	<i>written, ct, checks, CHKTTL, x, x1, x2</i>
Velocity of a bullet train	<i>velocity, trainVelocity, velocityInMph</i>	<i>vclt, v, tv, x, x1, x2, train</i>
Current date	<i>currentDate, todaysDate</i>	<i>cd, current, c, x, x1, x2, date</i>
Lines per page	<i>linesPerPage</i>	<i>lpp, lines, l, x, x1, x2</i>



Дужина назива

- Too long: *numberOfPeopleOnTheUsOlympicTeam*
numberOfSeatsInTheStadium
maximumNumberOfPointsInModernOlympics
- Too short: *n, np, ntm*
n, ns, nsisd
m, mp, max, points
- Just right: *numTeamMembers, teamMemberCount*



Опсег и променљиве

- Краћа имена прихватљива
- `for (int i = 0; i <)`
- Краће име – мали опсег трајања



Супротне променљиве

- begin/end
- first/last
- locked/unlocked
- min/max
- next/previous
- old/new
- opened/closed
- visible/invisible
- source/target
- source/destination (less common)
- up/down



Имена променљивих на основу улоге



Индекси у петљи

```
for ( teamIndex = 0; teamIndex < teamCount; teamIndex++ ) {  
    for ( eventIndex = 0; eventIndex < eventCount[ teamIndex ]; eventIndex++ ) {  
        score[ teamIndex ][ eventIndex ] = 0;  
    }  
}
```

Избегавање забуне шта је *i* а шта је *j*



Статусне променљиве

- *flag* није добро име
- Који је смисао статусне променљиве?
- Лоше име може доста да збунуи



Статусне променљиве

```
if ( statusFlag & 0x0F ) ...  
if ( printFlag == 16 ) ...  
if ( computeFlag == 0 ) ...
```

```
flag = 0x1;  
statusFlag = 0x80;  
printFlag = 16;  
computeFlag = 0;
```

ЛОШЕ!

```
if ( dataReady ) ...  
if ( characterType & PRINTABLE_CHAR ) ...  
if ( reportType == ReportType_Annual ) ...  
if ( recalNeeded == True ) ...
```

```
dataReady = True;  
characterType = CONTROL_CHARACTER;  
reportType = ReportType_Annual;  
recalNeeded = False;
```

ДОБРО!



Привремене променљиве

- Показују (не)разумевање кода
- Третирају се опуштеније него остале
- Више подложне грешкама



Привремене променљиве

```
// Compute roots of a quadratic equation.  
// This assumes that (b^2-4*a*c) is positive.  
temp = sqrt( b^2 - 4*a*c );  
root[0] = ( -b + temp ) / ( 2 * a );  
root[1] = ( -b - temp ) / ( 2 * a );
```

ЛОШЕ!

ДОБРО!

```
// Compute roots of a quadratic equation.  
// This assumes that (b^2-4*a*c) is positive.  
discriminant = sqrt( b^2 - 4*a*c );  
root[0] = ( -b + discriminant ) / ( 2 * a );  
root[1] = ( -b - discriminant ) / ( 2 * a );
```



Булеанске променљиве

- *done*
- *error*
- *found*
- *success*
- добра име имплицирају: **TRUE** / **FALSE**



Конвенције именовања



Конвенције?

- Конвенције убијају креативност и ефикасност ?
- Ефективни стандарди моћан алат



Конвенције?

- Допуштају да се неке ствари прихватају здраво за готово
- Лакше разумевање туђег кода
- Спречавају исти назив за различите ствари
- Наговештавају везу између одређених делова кода



Конвенције када користити?

- Више програмера на пројекту
- Препуштање софтвера другим људима на одржавање
- Када код пролази ревизију других људи
- Сувише велики пројекат да би се све памтило
- Термини специфични за пројекат



Типови конвенција

- Неформални
 - Често се користе – не зависе од језика
- Формални
 - Засновани на конкретном језику



Table 11-3. Sample Naming Convention for C++, and Java

Entity	Description
<i>ClassName</i>	Class names are in mixed upper and lower case with an initial capital letter.
<i>TypeName</i>	Type definitions including enumerated types and typedefs use mixed upper and lower case with an initial capital letter
<i>EnumeratedTypes</i>	In addition to the rule above, enumerated types are always stated in the plural form.
<i>localVariable</i>	Local variables are in mixed uppercase and lowercase with an initial lower case letter. The name should be independent of the underlying data type and should refer to whatever the variable represents.
<i>RoutineName()</i>	Routines are in mixed uppercase and lowercase. (Good routine names are discussed in Section 5.2.)
<i>m_ClassVariable</i>	Member variables that are available to multiple routines within a class, but only within a class, are prefixed with an <i>m_</i> .
<i>g_GlobalVariable</i>	Global variables are prefixed with a <i>g_</i> .
<i>CONSTANT</i>	Named constants are in <i>ALL_CAPS</i> .
<i>MACRO</i>	Macros are in <i>ALL_CAPS</i> .
<i>Base_EnumeratedType</i>	Enumerated types are prefixed with a mnemonic for their base type stated in the singular—for example, <i>Color_Red</i> , <i>Color_Blue</i> .



Префикси



Префикси

- Кориснички дефинисани типови (UDT) - скраћенице
- Односе се на тип објекта или променљиве коју именујемо
- Користе се кратки кодови
- Не односе се на стандардне типове програмског језика



Префикси

Table 11-6. Sample of UDTs for a Word Processor

UDT Abbreviation	Meaning
<i>ch</i>	Character (a character not in the C++ sense, but in the sense of the data type a word-processing program would use to represent a character in a document)
<i>doc</i>	Document
<i>pa</i>	Paragraph
<i>scr</i>	Screen region
<i>sel</i>	Selection
<i>wn</i>	Window



Префикси

- Семантички префикси
- Стандардни кроз пројекте
- Ближе одређују сврху



Префикси

Semantic Prefix	Meaning
<i>c</i>	Count (as in the number of records, characters, and so on)
<i>first</i>	The first element that needs to be dealt with in an array. <i>first</i> is similar to <i>min</i> but relative to the current operation rather than to the array itself.
<i>g</i>	Global variable
<i>i</i>	Index into an array
<i>last</i>	The last element that needs to be dealt with in an array. <i>last</i> is the counterpart of <i>first</i> .
<i>lim</i>	The upper limit of elements that need to be dealt with in an array. <i>lim</i> is not a valid index. Like <i>last</i> , <i>lim</i> is used as a counterpart of <i>first</i> . Unlike <i>last</i> , <i>lim</i> represents a noninclusive upper bound on the array; <i>last</i> represents a final, legal element. Generally, <i>lim</i> equals <i>last</i> + 1.
<i>m</i>	Class-level variable
<i>max</i>	The absolute last element in an array or other kind of list. <i>max</i> refers to the array itself rather than to operations on the array.
<i>min</i>	The absolute first element in an array or other kind of list.
<i>p</i>	Pointer



Префикси - предности

- Дају прецизност именима
- Чине имена компактнијим
- Мана: запостављање смисленог имена уз префикс
- Префикс све решава?



Имена за избегавање

- Збуњујући називи
- Имена са сличним значењем (input – inputValue)
- Слично име а различито значење
- Имена која слично звуче (wrap – rap)
- Нумеричка имена (file1, record2, ...)
- Погрешно спеловање
- Више стварних језика



Литература

Clean Code

Robert C. Martin



Popovich.Branko@gmail.com