

Popis jazyka

Základ jazyka je podobný regulárním výrazům:

nerezervovaný znak (nebo rezervovaný znak, který je předcházen zpětným lomítkem)	=	vypsán na výstup
znaky v hranatých závorkách	=	vypsán náhodný znak z výběru možno používat i intervaly - např. a-z
\d, \s, \w, \D, \S, \W	=	kategorie znaků (možno použít i uvnitř []): \d jsou číslice, \s bílé znaky, \w písmena, číslice a _ velkým písmenem se značí opak (např. \D -> cokoli kromě číslic)
(kód)	=	skupina, může začínat výrazem ve tvaru :název; - tedy (:název; kód) - takto lze skupinu pojmenovat
	=	znak ' ' odděluje možnosti, z možností va dané skupině se náhodně vybere
.	=	libovolný znak
\01	=	zpětná reference na 1. skupinu
\:název;	=	zpětná reference na skupinu s daným názvem
\$01	=	zpětná reference na 1. skupinu, namísto generování použije naposledy vygenerovanou hodnotu
\$:název;	=	zpětná reference na skupinu s daným názvem, namísto generování použije naposledy vygenerovanou hodnotu

Pozn.: rekurze pomocí zpětných referencí není povolena.

Stejně jako v regulárních výrazech lze používat kvantifikátory:

?	=	žádný nebo 1 výskyt
*	=	náhodný počet opakování, horní limit je knihovně zadán
*?	=	jako *, častěji generuje méně opakování
+	=	jako *, ale vždy vygeneruje alespoň 1 výskyt
+?	=	jako +, častěji generuje méně opakování
{n}	=	n výskytů
{n,}	=	alespoň n výskytů
{n,m}	=	alespoň n, nejvýše m výskytů

V kódu lze dále využít následující předdefinované skupiny:

\::maleName;	=	anglické mužské křestní jméno
\::femaleName;	=	anglické ženské křestní jméno
\::surname;	=	anglické příjmení
\::byte;	=	byte (číslo od 0 do 255)
\::int;	=	integer (4-bytové číslo se znaménkem)
\::uint;	=	bezznaménkový integer (4-bytové číslo bez znaménka)
\::usPhone;	=	telefonní číslo ve formátu používaném v USA
\::mailSuffix;	=	část platné mailové adresy začínající @
\::passwd;	=	heslo dlouhé 8 až 15 znaků, ve kterém je alespoň jedno malé písmeno, alespoň jedno velké písmeno, alespoň jedna číslice a alespoň jeden jiný znak

Další syntaxe:

(!U kód)	=	modifikuje vše co je generováno kódem na základě znaku za vykřičníkem, možné znaky jsou:
	U	= převod na velká písmena
	l	= převod na malá písmena
	C	= velká písmena na začátku slov
	_	= nahradí mezery a pomlčky za podtržítka
	''	= nahradí pomlčky a podtržítka za mezery
	-	= nahradí mezery a podtržítka za pomlčky
	n	= odstraní mezery
	t	= odebere bílé znaky ze začátku a konce
	0	= odebere nuly ze začátku
	!	= výstup kódu je skryt - možno použít jako komentář nebo pro definice skupin které mají být použity později
(?: kód)	=	na skupinu nelze zpětně odkazovat, v číslování skupin je ignorována
(" oddělovač ; opakování kód)	=	generuje pomocí kódu tolikrát, kolik je dáno opakováním, vygenerované výrazy oddělí oddělovačem, např. (" \ ;{5} \d{2})
(' oddělovač ; opakování kód)	=	jako s dvojitými uvozovkami, ale generuje pouze jeden výraz a ten opakuje

API knihovny

API se skládá ze třídy **RevgeX** ve jmeném prostoru **ReverseRegex**. K vytvoření generátoru slouží její konstruktor:

```
RevgeX(string code,  
       bool ignoreLineEndings = false,  
       bool ignoreWhitespace = false,  
       bool allowPredefinedGroups = true)
```

code	=	kód generátoru
ignoreLineEndings	=	pokud je nastaven na true , jsou v kódu ignorovány konce řádků
ignoreWhitespace	=	pokud je nastaven na true , jsou v kódu ignorovány bílé znaky
allowPredefinedGroups	=	nastavením na false lze zakázat předdefinované skupiny

Pozn.: je-li parametr *ignoreWhitespace* nastaven na **true**, pak musí být na **true** nastaven i *ignoreLineEndings*, jinak je vyvolána výjimka typu **ArgumentException**.

Je-li je zapnuto ignorování bílých znaků, lze bílé znaky v kódu vynutit escape sekvencí, konkrétně '\ ' pro mezeru, '\t' pro tabulátor a '\n' pro konec řádky.

Pokud je ve specifikovaném kódu syntaktická chyba, je vyvolána výjimka typu **FormatException** s příslušnou chybovou hláškou.

Ke generování dat slouží následující metoda:

```
string Generate(int repetitionLimit [, Random random])
```

repetitionLimit	=	maximální počet opakování při použití kvantifikátorů * a +, musí být ≥ 1
random	=	náhodný generátor

Vygeneruje náhodný string.

Je-li *repetitionLimit* < 1 , je vyvolána výjimka typu **ArgumentOutOfRangeException**.

Pokud není náhodný generátor specifikován, nebo je nastaven na null, bude použit výchozí generátor.

Pozn.: Knihovna generuje pouze ASCII znaky.