

**Pracownia Programowania  
Obiektowego**

**Lista nr 4 - 23.03.2018**

**Język programowania:** *C#*

**OS:** Windows 10 Enterprise 64-bit

**IDE i kompilator:** JetBrains Raider

**Target framework:** .NETFramework,  
version: 3.5

Opis implementacji zadań 2 oraz 4

Mateusz Zając, 298654

23.03.2018

## 1 Zadanie 2

Zadanie polega na utworzeniu kolekcji ***PrimeCollection***, która „w locie” wylicza kolejne liczby pierwsze. Każdy obrót pętli **foreach** powoduje wyświetlenie następnej liczby pierwszej, aż do osiągnięcia zakresu typu **int**. Implementacja używa interfejsu **IEnumerator**, aby móc wykorzystać pętlę **foreach**, która jest wymagana w poleceniu zadania. Rozpoczynając od liczby **2**, pętla po tym wypisze kolejno liczby: **3, 5, 7** itd.

Pole występujące w klasie ***Prime\_Num***:

- **int n** - zawiera informację o aktualnie badanej liczbie pierwszej;

## 2 Zadanie 4

Zadanie polega na utworzeniu generatora, który będzie tworzył słowa za pomocą operacji gramatyki bezkontekstowej. Innymi słowy, program przy pomocy zdefiniowanych operacji gramatyki  $\mathbb{G}$  generuje poprawne słowa tejże gramatyki.

Niech  $\mathbb{G} = \{\mathbb{N}, \mathbb{T}, \mathbb{P}, \mathbb{S}\}$  będzie rozpatrywaną gramatyką bezkontekstową. Wtedy kolejno:

- $\mathbb{N} = \{\mathbb{S}\}$  - Zbiór nieterminali
- $\mathbb{T} = \{a, b, c, d\}$  - Zbiór terminali
- $\mathbb{P} = \{\mathbb{S} \rightarrow a\mathbb{S}b, \mathbb{S} \rightarrow c\mathbb{S}d, \mathbb{S} \rightarrow \mathbb{S}\mathbb{S}, \mathbb{S} \rightarrow \varepsilon\}$  - Zbiór produkcji
- $\mathbb{S}$  - Symbol startowy (wyróżniony)
- $\varepsilon$  - Symbol słowa pustego, w programie oznaczany także jako: \*

Klasa ***Productions*** odpowiada za każdy pojedynczy element wygenerowanego słowa. Przykładowo, w słowie:  $a\mathbb{S}b$ , są obecne obiekty o wartościach:  $a$ ,  $\mathbb{S}$  oraz  $b$ .

(Schemat tworzenia słów dostępny w załączniku)

Pola klasy ***Productions***:

- **string value;** - wartość obiektu, jeden z terminali lub nieterminali
- **Productions prev;** - odnośnik do kolejnego elementu słowa
- **Productions next;** - odnośnik do poprzedniego elementu słowa

Klasa ***Lista*** odpowiada za produkcję słów w gramatyce. Tablica obiektu tej klasy zawiera w sobie referencje do każdego obecnego w słowie symbolu startowego  $\mathbb{S}$ . Dowiązane zaś do tych elementów ciągi znaków terminali mogą być dowolnie długie. W przykładzie:  $acac\mathbb{S}dbdb$  jedyną referencją jest środkowy symbol  $\mathbb{S}$ . Pozostałe elementy są dowiązane do elementu startowego.

Pola klasy ***Lista***:

- **Productions[ ] listOfProductions;** - tablica mająca w sobie referencje do wszystkich  $S$ ;
- **int lastAdded;** - indeks ostatnio dodanego elementu tablicy;
- **int indexer;** - indeks odpowiadający za przejście po kolejnych elementach tablicy;
- **int listSize;** - wielkość listy;