

Wprowadzenie do Programowania - Kolokwium **GRUPA A**

Zadania - łącznie **25 pkt**

- 1 - 3: otwarte, po **3 pkt** każde
4: otwarte, **6 pkt**
5 - 9: zamknięte, po **2 pkt** każde

1. Napisz metodę *Even* która generuje **liczbę losową** z przedziału 0 – 9 i wyświetla na ekranie "Good" **jeśli** jest parzysta, albo "Bad" w przeciwnym razie. Metoda nie powinna przyjmować żadnych argumentów.

```
static void Even()  
{  
    // Ciało metody do napisania  
}
```

Do generowania losowych liczb w przedziale [0, m) użyj klasy *Random* oraz metody *Next(0, m)* z tej klasy.

-
2. Napisz metodę *Product* która przyjmuje jako argument liczbę całkowitą *n* i zwraca **iloczyn wszystkich liczb** od 1 do *n*, czyli: $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$. Jeśli $n \leq 1$, funkcja powinna zwrócić 1.

```
static int Product(int n)  
{  
    // Ciało metody do napisania  
    return ...  
}
```

-
3. Napisz metodę *Sum2D* która przyjmuje jako argument **tablicę 2D** liczb double i, używając **dwóch pętli for**, zwraca sumę wszystkich elementów tablicy. Np. w poniższym przypadku wynik to $3.0 + 4.0 + 7.1 + 2.5 + 6.0 + 1.0 = 23.6$.

3.0	4.0	7.1
2.5	6.0	1.0

```
static double Sum2D(double[,] tab2D)  
{  
    // Ciało metody do napisania  
    return ...  
}
```

Wysokość i szerokość tablicy można otrzymać za pomocą *tab2D.GetLength(x)*, gdzie *x* = 0 lub 1 to numer pożądanego wymiaru.

4. Programowanie obiektowe. Napisz:
- klasę** *Square* z jedną publiczną właściwością *double Side* posiadającą publiczne akcesory *get* i *set*,
 - konstruktor** do klasy który przyjmuje jeden argument typu *double* i używa go do ustawienia właściwości *Side*,
 - interfejs** *IFigure* z deklaracją jednej publicznej metody *Area* która nie przyjmuje żadnych argumentów, a zwraca typ *double*,
 - zaimplementuj** powyższy interfejs w klasie *Square*, pisząc kod wymagany do działania. Ciało metody zapełnij wedle woli, tak by nie była pusta.

Wskazówka: właściwości mają formę public TYP NAZWA { get; set; }

5. Które z poniższych służy do przechowywania wielu elementów tego samego typu, ze stałą i z góry ustaloną liczbą elementów (bez możliwości dodawania nowych i usuwania starych)?
- lista
 - tablica
 - interfejs
6. Do łączenia warunków logicznych (np.: jeśli $x > 0$ i **jednocześnie** $x < 5$) służy:
- &&
 - ||
 - !
7. Klasa A posiada publiczną właściwość X. Klasa B posiada publiczną właściwość Y oraz dziedziczy po klasie A. Które z poniższych jest prawdziwe?
- klasa A ma dostęp tylko do X, a klasa B tylko do Y
 - klasa A ma dostęp tylko do X, a klasa B zarówno do X jak i Y
 - każda z klas A i B ma dostęp zarówno do X jak i Y
8. Dla poniższego fragmentu kodu:
- ```
int x = 0;
for (int i = 0; i < 5; ++i)
 x += 10;
```
- wynik działania to:
- $x = 30$
  - $x = 40$
  - $x = 50$
9. Do przechwytywania i obsługi wyjątków (Exceptions) służy:
- while...
  - try... catch... finally...
  - if... else if... else...