

Programowanie obiektowe – Egzamin – 14.09.2010

Zadanie 1: Przy założeniach początkowych:

```
long n=13, t[]={2,4,7,9,3}, *p=t+1;
double z[]={5,3,2,1}, *q=z+2;
char napis[] = "poprawka", *tp=napis+1;
```

podaj typ i wartość wyrażenia:

- | | |
|---------------------------------|--------------------|
| (a) $x? *p: *t$ | (e) $n / *p **t$ |
| (b) <code>sizeof(napis)</code> | (f) $*napis - *tp$ |
| (c) <code>sizeof(z) / *q</code> | (g) $*t *p \& n$ |
| (d) $z[p-t] + z[*p - *t]$ | (h) $n > *t < *p$ |

Zadanie 2: Przy założeniach poprzedniego zadania, które zmienne po wykonaniu poszczególnych instrukcji zmienią swoją wartość i na jaką?

- | | |
|------------------------|------------------------------|
| (a) $*t *= *p--;$ | (d) $*p += *t \& \& n--;$ |
| (b) $tp += *p++;$ | (e) $*(napis + *p) += *t;$ |
| (c) $n \&= *++p *t;$ | (f) $*(z + n \% 4) = -- *t;$ |

Zadanie 3: Przedstaw fragmenty klasy **Macierz** (w C++) z dynamiczną alokacją pamięci: pola do przechowywania danych, konstruktor z dwoma argumentami określającymi wymiary macierzy, konstruktor kopiujący oraz destruktor.

Zadanie 4: Napisz w C++ szablon funkcji:

- **Abs** wyznaczającej wartość bezwzględną elementów typów dysponujących operatorami - (jednoargumentowy minus) oraz <, a także konstruktorem z jednym argumentem całkowitym,
- **MaxAbs** wyznaczającej element o maksymalnej wartości bezwzględnej dla tablic elementów dowolnego typu zgodnego z założeniami dla **Abs**.

Zadanie 5: Opisz funkcjonalność C# związaną ze słowem kluczowym **params**.

Zadanie 6: Napisz w C# interfejs **IFigura** zawierający metody **Obwód** oraz **Pole** i klasę **Prostokąt** implementującą ten interfejs.

Ocena:

Za każde zadanie można uzyskać 1 punkt. Ocena zostanie wystawiona na podstawie łącznej liczby punktów, według następującej tabeli:

Punkty	Ocena
[0, 3)	2
[3, 4)	3
[4, 5)	4
[5, 6]	5

Osoby ubiegające się o zaliczenie powinny przekroczyć próg 2.4 punktu.

Priorytety operatorów języka C++

Operatory	Łączność
() [] -> :: .	lewostronna
! ~ + - ++ -- & * sizeof new delete (typ)	prawostronna
. * ->*	lewostronna
* / %	lewostronna
+ -	lewostronna
<< >>	lewostronna
< <= > >=	lewostronna
== !=	lewostronna
&	lewostronna
^	lewostronna
	lewostronna
&&	lewostronna
	lewostronna
?:	lewostronna
= *= /= %= += -= &= ^= = <<= >>=	prawostronna
,	lewostronna