

Przykładowe pytania na kolokwium z Programowania obiektowego C++

Ważne pojęcia:

- klasa i struktura, obiekt, zmienna
- abstrakcja i enkapsulacja
- interfejs i implementacja
- modyfikatory dostępu: składowe prywatne, publiczne i chronione
- funkcje: składowe, stałe składowe, pomocnicze, zaprzyjaźnione
- składniki statyczne
- wskaźnik, referencja
- stos i sarta
- wyciek pamięci
- konstruktor: domyślny, kopiujący, konwertujący
- operator przypisania, destruktor
- kopia płytka i głęboka
- dziedziczenie
- klasa bazowa i klasa pochodna
- polimorfizm statyczny i dynamiczny
- funkcje wirtualne i czysto wirtualne
- klasa abstrakcyjna
- klasa funkcyjna

Przykładowe pytania dot. pojęć:

1. Wymień 3 podstawowe techniki programowania zorientowanego obiektowo. Zdefiniuj jedną z nich
2. Uzasadnij dlaczego stosuje się składowe prywatne w klasach
3. Wyjaśnij różnicę pomiędzy płytką a głęboką kopią obiektu na przykładzie obiektów r1 i r2 klasy R. Możesz posłużyć się rysunkiem.

```
class R {  
    int rozmiar;  
    char *napis;  
    //...  
};
```

```
R r1;  
R r2 = r1;
```

Inne zadania:

4. Zdefiniuj klasę **P** zawierającą:
 - prywatne pole statyczne typu **int**
 - chronione pole typu wskaźnik na **char**
 - **vector** liczb typu **double**
 - publiczny konstruktor domyślny
 - prywatny konstruktor kopiujący
 - prywatny operator przypisania
 - wirtualną funkcję, która nic nie zwraca i nie przyjmuje parametrów wywołania deklarację przyjaźni z funkcją, która zwraca zmienną typu wskaźnik na **char** i pobiera stałą referencję do **vector**-a liczb **double**

5. Pewien programista zadeklarował następujące operatory dla struktury W:

```
struct W {
    int wartosc;
    W(int nowa_wartosc): wartosc(nowa_wartosc) { };
    operator int() { return wartosc; };
    W operator+=(const W& prawy);
};
W W::operator+=(const W& prawy) { wartosc+=prawy.wartosc; return *this; };
const W operator+(W& lewy, W& prawy) { return lewy+=prawy; }
```

Następnie skompilował program zawierający poniższy kod. Jakie wartości dostał na wyjściu?

```
W a=1,b=2,c=3;
(a += b) += c;
cout << a << " " << b << " " << c << endl; .....
a + b;
cout << a << " " << b << " " << c << endl; .....
```

Popraw deklaracje operatorów, by były zgodne z zasadami sztuki pisania operatorów.

Jeśli przewidujesz, że po Twojej poprawce kod nie skompiluje się, zaznacz gdzie wystąpi błąd.

```
..... W::operator+=(.....) { wartosc+=prawy.wartosc; return *this; };
..... operator+(....., .....) { return lewy+=prawy; }
```

6. W poniższym kodzie wystąpił błąd kompilacji. Dopisz brakującą składową kl. Y, co rozwiąże problem:

```
class Y {
    std::vector<int> dane;
public:
    Y(int n): dane(std::vector<int>(n,0)) { };
    int& operator[](int który) { return dane[który]; };
    .....
};
int main() {
    Y y(5);
    const Y y2(4);
    y[0] = y2[0];
}
```

7. Zaznacz i opisz błędy w poniższym kodzie. Przyjmujemy, że komponenty biblioteczne są dostępne:

```
class Publikacja {
    string p_autor, p_tytul;
    int p_rok;
    Publikacja(const string &a, const string &t, int r) : p_autor(a), p_tytul(t), p_rok(r) { };
public:
    virtual void pokaz() const = 0;
    virtual ~Publikacja() {};
};
class Artykul : public Publikacja {
    string a_czasopismo;
public:
    Artykul(const string &a, const string &t, int r, const string& cz):
        Publikacja(a,t,r), a_czasopismo(cz) { };
    ~Artykul() {};
};
int main() {
    Publikacja pub("Wincenty Kadlubek", "Chronica...", 1208);
};
```

9. Na osiach życia obiektów zaznacz, w których zakresach są dostępne (pogrubienie)
 Ponadto zaznacz znakiem „-” utworzenie, a znakiem „X” usunięcie obiektu z pamięci.

	a	lis	wsk_lis	*wsk_lis
Lista a(10);				
Lista *funkcja1(int n)				
{				
Lista* wsk_lis = 0;				
while (n) {				
Lista lis(n);				
if (/* ... */) {				
wsk_lis = new Lista(lis);				
break;				
}				
n--;				
}				
return wsk_lis;				
}				
.....				
int main()				
{				
Lista* wsk_lis = funkcja1(21);				
//				
delete wsk_lis;				
}				
.....				

Uwagi

- Przewiduję od 6 do 8 zadań na kolokwium
- Obowiązuje zasada „zero tolerancji” dla ściągania (bez ostrzeżeń i bez możliwości odwołania)

Podział na grupy 19/11

- Osoby o nazwiskach zaczynających się na litery od A do Ł, godz. 13.15
- Osoby o nazwiskach zaczynających się na litery od M do Ż, godz. 14.15