

Zajęcia 7: Dziedziczenie i funkcje wirtualne

1. Rozbudowa klasy Pracownik

Do klasy Pracownik dodaj dwie poniższe metody:

- wirtualną metodę `virtual void WypiszDane()`, która ma wypisywać na ekran wszystkie dane o pracowniku;
- wirtualną metodę `virtual Pracownik* KopiaObiektu()const`, której zadaniem jest zwrócenie nowo stworzonego obiektu (na wzór obiektu, na rzecz którego wywołana jest ta metoda).
- wirtualny destruktor

2. Opracowanie klasy Kierownik

Do projektu dodaj klasę o nazwie Kierownik pochodną klasy Pracownik.

Dodanie danych składowych do klasy Kierownik

W części prywatnej zadeklaruj następujące pola składowe:

```
Napis m_NazwaDzialu;
int m_nliczbaPracownikow;
```

W części publicznej zadeklaruj następujące metody:

- konstruktor domyślny inicjujący wszystkie składowe obiektu klasy;
- konstruktor kopiujący inicjujący pola składowe obiektu bazując na wzorcu;
- operator przypisania;
- operator porównania:
`bool operator==(const Kierownik & wzor) const;`
metoda ma zwracać `true` gdy dane składowe obiektów są jednakowe, `false` w przeciwnym wypadku.
- wirtualną metodę `virtual void WypiszDane()`, która ma wypisywać na ekran wszystkie dane o kierowniku;
- wirtualną metodę `virtual Pracownik* KopiaObiektu()const`, której zadaniem jest zwrócenie nowo stworzonego obiektu (na wzór danego);
- globalną, zaprzyjaźnioną z klasą Kierownik funkcję operatorową wstawiania do strumienia wyjściowego. Zadaniem tej funkcji jest wyprowadzenie informacji o danym sprzedawcy. Działanie funkcji przetestuj na strumieniach:
`ostream & operator<<(ostream & wy, const Kierownik & s);`
- globalną, zaprzyjaźnioną z klasą Kierownik funkcję operatorową pobierania ze strumienia wejściowego. Zadaniem tej funkcji jest wprowadzenie informacji do danych składowych obiektu klasy. Działanie funkcji przetestuj na strumieniach:
`istream & operator>>(istream & we, Kierownik & s);`

3. Funkcja główna

Zmodyfikuj odpowiednio interfejs użytkownika, w którym uwzględnione będzie m.in. dodawanie obiektów Kierownik do listy.