

# Laboratorium JP

## Projekt 3

### Klasa, konstruktory i destruktory, przeciążanie funkcji i operatorów

1. Zaprojektować zestaw trzech klas będących obiektywnym odwzorowaniem ..... , z których jedna będzie klasą główną, reprezentującą cały obiekt i będzie zawierać dwa obiekty pozostałych klas (podobiekty; w tym jeden **automatyczny** i jeden **dynamiczny**). Podobiekt dynamiczny oznacza, że polem obiektu głównego jest wskaźnik na podobiekt (do celów tego projektu w konstruktorze domyślnym **ma być inicjowany na zero**). Odwzorowanie powinno być możliwie **realistyczne** - dla skomplikowanych obiektów odpowiednio uproszczone. Zaprojektowane klasy będą wykorzystywane w następnym projekcie – ewentualnie będą modyfikowane.
2. Model rozwiązania należy przedstawić prowadzącemu na zajęciach kolejnych (tydzień po otrzymaniu treści projektu). Model powinien określać:
  - nazwy klasy głównej oraz klas reprezentujących podobiekty: dynamiczny oraz automatyczny,
  - przykładowe pola oraz metody dla klas.
3. We wszystkich konstruktorach i destruktorach wstawić kod drukujący na ekran informację o wywołaniu (warunkowo, pod zmienną kompilacji **TESTPR**). Wydruki te będą pomocne w czasie śledzenia sekwencji wywołania konstruktorów i destruktorów.
4. Klasa główna ma zawierać mechanizm określania liczby stworzonych obiektów tego typu (**statyczne pole** klasy), oraz **statyczną metodę** zwracającą to statyczne pole klasy (chodzi o możliwość uzyskania informacji o braku instancji klasy).
5. Każda klasa powinna prawidłowo zachowywać się w przypadku **kopiowania** – realizacja konstruktora kopiującego.
6. Zaprojektować i zaimplementować dla klas kilka **sensownych, różnorodnych** operatorów, w tym: jeden jednoargumentowy, jeden dwuargumentowy oraz operator przypisania. Zastosować wybrane operatory jako metody klas oraz jako funkcje zaprzyjaźnione z klasami.
7. Jeżeli jest wybór pomiędzy stosowaniem mechanizmów, funkcji, instrukcji typowych dla języka C i C++ proszę stosować odpowiednie konstrukcje właściwe dla C++ np. char\* -> string, FILE\* -> ifstream, ablica w stylu C -> std::vector itp.
8. Napisać program główny testujący klasę główną i jej podklasy (oddzielny moduł/plik). Dla testów **stworzyć** obiekty **automatyczne, dynamiczne i statyczne (lokalne, globalne)** w funkcji **main** i innej **funkcji testowej** (w trakcie testu powinno powstać co najmniej kilkanaście obiektów). Celem powyższych testów jest między innymi obserwowanie **czasu życia obiektów** oraz zachowanie się funkcji i operatorów przeciążonych. Fragment programu powinien testować kopiowanie obiektów. Program testowy powinien być nieinterakcyjny (statyczny) z odpowiednią liczbą wydruków objaśniających, co zaszło w czasie działania programu.

<b>Zadanie</b>	<b>Punkty</b>
Poprawny model (8 grudzień 2015)	1 pkt
TESTPR	0,5 pkt
Klasa główna (pola, metody)	1 pkt
Podobiekt dynamiczny (pola, metody)	1,5 pkt
Podobiekt automatyczny (pola, metody)	0,5 pkt
Pole statyczne + metoda statyczna	1 pkt
Konstruktor kopiujący	0,5 pkt
Operator przypisania	1 pkt
Operatory inne	1 pkt
Wycieki pamięci / błędy	1 pkt
Jakość kodu (zmienne, wcięcia, przejrzystość)	1 pkt
<b>Razem</b>	10 pkt

### Lista obiektów reprezentujących klasę główną:

#### Grupa 8-10

1. Samochód
2. Komputer stacjonarny
3. Laptop
4. Drukarka
5. Samolot
6. Lotnisko
7. Sklep
8. Drukarka
9. Telewizor
10. Telefon

#### Grupa 16-18

11. Szpital
12. Pociąg
13. Lampa
14. Odkurzacze
15. Odtwarzacz DVD
16. Ekspres do kawy
17. Komisariat
18. Pracownik
19. Biurko
20. Warzywniak