

Kolokwium z Programowania Obiektowego – zestaw A1

Wszystkie poniższe czynności wykonaj w ramach jednego projektu tak, aby spełniony był załączony diagram UML. Możesz dodawać w klasach pola, metody (o ile nie zaznaczono inaczej), choć staraj się by ich liczba była jak najmniejsza i nie powielala funkcji, które są już w poleceniach.

0. Stwórz projekt – aplikacja konsolowa lub WPF (przemyśl wybór, bo zmiana może być czasochłonna).
1. Rozpocznij od typu wyliczeniowego Kolor. Ustaw wartości w kolejności zgodnej z diagramem.
2. Następnie stwórz klasę Dom. W tej klasie nie twórz w sposób jawny żadnego konstruktora. Dodatkowe informacje:
 - a. metoda WstawOkna powinna ustawić pole okno zgodnie z parametrem;
 - b. metoda ZamontujDrzwi powinna ustawić pole drzwi zgodnie z parametrem;
 - c. metoda PomalujElewacje powinna ustawić pole kolorElewacji zgodnie z parametrem;
 - d. przesłoń metodę ToString() tak, aby zwracała string z opisem domu np. Okna: plastikowe, Drzwi: antywłamaniowe, Kolor elewacji: Czerwony. Jeśli pole drzwi lub okno jest puste, to opis nie powinien uwzględniać tych pól.
3. Następnie stwórz klasę ProjektDomu. Dodatkowe informacje:
 - a. nie inicjuj pola dom przy jego deklaracji;
 - b. w wnętrzu metody NowyDom() do pola dom przypisz nowe odwołanie/referencję do obiektu typu Dom za pomocą konstruktora domyślnego;
 - c. metody WstawOkna i ZamontujDrzwi mają być wirtualne;
 - d. metoda PomalujElewacje ma być abstrakcyjna;
 - e. we wnętrzu metody WstawOkna dla pola dom wywołaj metodę WstawOkna zgodnie z parametrem;
 - f. we wnętrzu metody ZamontujDrzwi dla pola dom wywołaj metodę ZamontujDrzwi zgodnie z parametrem;
 - g. przesłoń metodę ToString() tak, aby zwracała opis pola dom.
4. Następnie stwórz klasę Biurowiec. Dodatkowe informacje:
 - a. przesłoń metodę PomalujElewacje tak, aby dla pola dom została wywołana metoda PomalujElewacje zgodnie z parametrem;

- b. przesłoń metodę `ZamontujDrzwi` tak, aby dla pola `dom` została wywołana metoda `ZamontujDrzwi` z parametrem `"szkłane"`;
 - c. przesłoń metodę `ToString()` tak, aby zwracała string powstały ze stringa z klasy wyżej z dodanym na początku łańcuchem `"Biurowiec:"`;
5. Dalej stwórz klasę `DomJednorodzinny`. Dodatkowe informacje:
- a. przesłoń metodę `PomalujElewacje` tak, aby dla pola `dom` została wywołana metoda `PomalujElewacje` z parametrem `Czerwony`;
 - b. przesłoń metodę `ToString()` tak, aby zwracała string powstały ze stringa z klasy wyżej z dodanym na początku łańcuchem `"Dom jednorodzinny:"`;
6. Następnie stwórz klasę `SzefBudowy`. Dodatkowe informacje:
- a. nie inicjuj pola `projektDomu` przy jego deklaracji;
 - b. w tej klasie nie twórz jawnie żadnego konstruktora;
 - c. metoda `WybierzProjekt` ma swoim wnętrzu ustawić pole `projektDomu` zgodnie z parametrem;
 - d. we wnętrzu metody `Buduj` dla pola `projektDomu` wywołaj w kolejnych instrukcjach metody: `NowyDom`, `WstawOkna`, `ZamontujDrzwi`, `PomalujElewacje` (parametry są sugestystycznie nazwane na diagramie);
 - e. przesłoń metodę `ToString()` tak, aby zwracała string z opisem pola `projektDomu()`.
7. Dokończ program robiąc aplikację konsolową/wpf (zgodnie z początkowym wyborem) na operacje wejścia/wyjścia tak, aby przetestować metody z klasy `SzefBudowy`.
- a. w aplikacji może być stworzony biurowiec lub dom jednorodzinny;
 - b. parametry np. drzwi, okna, kolor (gdy potrzeba) należy pobrać od użytkownika.
 - c. pola tekstowe nie powinny być puste lub mniejsze niż 2 znaki;
 - d. komunikat z potwierdzeniem udanej budowy odpowiedniego domu powinien być wyświetlony na konsoli/MessageBoxie lub dowolnej kontrolce.

Gotowy kod (niespakowany) prześlij jako oddzielne repozytorium do serwisu Github. Link wyślij mailem na adres piotr.jastrzebski@uwm.edu.pl lub piojas@matman.uwm.edu.pl w temacie wpisując KolokwiumPO.

Sprawdzenie będzie rozpoczęte tylko gdy kod kompiluje się. W przypadku braku kompilacji 0 pkt za całe kolokwium.

Punktacja:

- Polecenie 1 – 3 pkt
- Polecenie 2 – 7 pkt
- Polecenie 3 – 10 pkt
- Polecenie 4 – 5 pkt
- Polecenie 5 – 5 pkt
- Polecenie 6 – 10 pkt
- Polecenie 7 – 10 pkt za logikę aplikację, 10 pkt za poprawne działanie kontrole/właściwe działanie wejścia i wyjścia konsolowego, 10 pkt za właściwe wykorzystanie metod z punktów 1-6.

Przykładowe wyglądy aplikacji:

Rodzaj okien

Rodzaj drzwi

Kolor elewacji

Info o projekcie

```
Co chcesz zrobić?  
A - buduj biurowiec  
B - buduj dom jednorodzinny  
X - wyjście z aplikacji
```

