

Laboratorium 2

1. Proszę zdefiniować klasę `Telewizor` z publicznymi polami `jasność`, `kontrast` i `głośność` typu `int`. Proszę zdefiniować publiczną metodę `wypisz()`, wypisującą informacje o telewizorze, np. „Telewizor o jasności 45, kontraście 80 i głośności 20.”. Proszę zapisać do pól wartości odpowiednio: -10, 2000 oraz -350.
2. Proszę zmodyfikować poprzednią klasę i program wprowadzając hermetyzację składowych (pola definiujemy jako prywatne, definiujemy publiczne „getter” oraz „setter” na polach: np. `int getJasność()`, `void setJasność(int j)`). „Setter” powinny zapewnić, że jasność i kontrast będą w przedziale $\langle 0, 100 \rangle$, natomiast głośność w przedziale $\langle 0, 30 \rangle$. Metoda `wypisz()` jak wyżej.
3. Proszę zdefiniować klasę `PunktZParzystejĆwiartki`, reprezentującą punkt na płaszczyźnie znajdujący się w 2 lub 4 ćwiartce układu współrzędnych, z prywatnymi polami `int x` oraz `int y`. Stosujemy hermetyzację składowych (standardowe „getter” oraz „setter” na polach) do zapewnienia, że modyfikacje pól nie spowodują wyjścia poza pożądane wartości. Zauważmy, że przy takiej implementacji nie da się w żaden sposób punktu przerzucić z 2 ćwiartki do 4 ani na odwrót. Aby temu zaradzić definiujemy więc metodę `void setXY(int, int)`, która będzie modyfikować równocześnie obie współrzędne punktu po sprawdzeniu, że są one poprawne. Definiujemy metodę `wypisz()`, która wypisze tekst typu: „Punkt z drugiej ćwiartki, o współrzędnych (-2,5)” lub „Punkt z czwartej ćwiartki, o współrzędnych (2,-1)”.
4. Klasa `trójkąt` (`Tri`), złożona z 3 punktów, z prywatnymi obiektami typu `Punkt`. Wszystkie metody wymienione następnie, mają być publiczne. Metoda `void setABC(int AX, int AY, int BX, int BY, int CX, int CY)` powinna ustawić wartości dla punktów wewnątrz klasy. Metoda `Punkt getA()` zwraca wierzchołek A, `Punkt getB()` zwraca wierzchołek B, tak samo dla C. Metoda `double permimeter()` ma zwrócić obwód trójkąta (**Wskazówka:** wykorzystać tutaj jest dobrze liczenie odległości z poprzednich zajęć).
5. (!) Metodę z 4., `void setABC(int, int, int)`, rozszerzyć- zadbać o to, żeby podane przez użytkownika dane faktycznie nadawały się do utworzenia trójkąta (**Wskazówka:** Trójkąt ABC można utworzyć z boków AB, BC, AC, tylko gdy spełniony dla nich jest pewien warunek w postaci nierówności.)