

Laboratorium 7

Klasy Abstrakcyjne i Interfejsy

1. Utwórz klasę abstrakcyjną o nazwie *Shape* z właściwością *name* typu *String*. W klasie *Shape* utwórz abstrakcyjne metody:
 - *getPerimeter()*, która oblicza obwód figury i zwraca wynik typu *double*.
 - *getArea()*, która zwraca wynik typu *double*. Metoda ta oblicza pole powierzchni figury geometrycznej.
 - *details()*, która będzie wyświetlała szczegóły figury.
2. Utwórz klasę dziedziczącą z klasy abstrakcyjnej o nazwie *Circle()*, która oprócz pola *name* będzie posiadała pole *radius*. Wypełnij metody *getPerimeter()* oraz *getArea()* w taki sposób, aby obliczały obwód i pole okręgu. Metoda *details()* ma wyświetlać szczegóły obiektu.
3. Utwórz kolejne klasy dziedziczące z klasy abstrakcyjnej dla **kwadratu**, **prostokąta** i **trójkąta**. Dodaj odpowiednie pola potrzebne do obliczenia obwodów i pól powierzchni wyżej wymienionych figur. Metoda *details()* ma wyświetlać szczegóły każdego z obiektów.

Zadbaj aby napisany program był odporny na wprowadzone niepoprawne dane. Utwórz przykładowe obiekty i zaprezentuj działanie metod.

4. Utwórz interfejs *ShapeIF*, który ma zawierać właściwość *name* oraz metody z punktu pierwszego sprawozdania. Utwórz następnie 4 klasy z poprzednich zadań, które będą dziedziczyły po interfejsie *ShapeIF*. Zadbaj aby napisany program był odporny na wprowadzone niepoprawne dane. Utwórz przykładowe obiekty i zaprezentuj działanie metod.

5. Zaprojektuj klasę enum *Week*, z nazwami dni tygodnia. Enum powinien zawierać metodę *isWeekend()*, która sprawdza, czy dany dzień tygodnia jest weekendem (sobota, niedziela) oraz metodę *isWorkingDay()*, która sprawdza, czy dany dzień jest dniem pracującym. Metody powinny zwrócić wartość typu logicznego. W klasie testującej utwórz metodę przyjmującą jako parametr dzień tygodnia i wyświetlającą stosowny komunikat. Przetestuj działanie programu w metodzie *main()*.