

Zad.1(definiowanie klas)

Twój kolega ma spory zbiór płyt muzycznych. Chce je uporządkować. Poprosił Ciebie, abyś napisał mu program, który pozwoli mu zarządzać tym zbiorem, czyli taką bazę płyt. O każdej płycie chce posiadać następujące informacje:

- tytuł płyty
- czas trwania płyty
- spis utworów znajdujących się na każdej płycie
- spis wykonawców, którzy wykonują utwory na danej płycie
- numer płyty (identyfikator płyty nadany przez kolegę)

O każdym utworze kolega chce wiedzieć:

- tytuł utworu
- czas trwania utworu
- spis wykonawców, którzy wykonują dany utwór na danej płycie
- kompozytora danego utworu
- numer utworu na płycie

Program powinien umożliwić:

- dodanie płyty do bazy danych
- wyświetlenie wszystkich płyt znajdujących się w bazie danych (same tytuły płyt)
- wyświetlenie szczegółowej informacji na temat płyty (tytuł płyty, typ płyty, czas trwania płyty, spis tytułów utworów znajdujących się na płycie z podanym numerem utworu na płycie)
- wyświetlenie wykonawców, którzy wykonują utwory na danej płycie
- wyświetlenie szczegółowej informacji na temat wybranego utworu z danej płyty (tytuł utworu, czas trwania utworu, spis wykonawców którzy wykonują dany utwór, kompozytora danego utworu)

Twoim pierwszym zadaniem jest opracowanie diagramu klas. Przed przystąpieniem do implementacji

musisz uzyskać akceptację kolegi (prowadzącego), czy projekt jest zgodny z jego oczekiwaniami.

Po uwzględnieniu uwag kolegi (prowadzącego) zaimplementuj program.

Zad.2 (pola i metody statyczne)

Twoim zadaniem jest utworzenie programu, który wczytywałby plik tekstowy i analizował go pod względem ilości występowania określonych znaków, fraz itp. Firma w której pracujesz zajmuje się obróbką tekstów, więc dodatkowym wymaganiem jest, abyś utworzył bibliotekę metod analizujących łańcuch znaków. Biblioteka ta powinna zawierać następujące metody:

- Metodę obliczającą ilość wystąpień określonego znaku.
- Metodę obliczającą ilość wystąpień określonego ciągu znaków
- Metodę obliczającą ilość wystąpień określonej litery. Metoda powinna mieć dwie wersje, jedną uwzględniającą wielkość litery, drugą obliczającą ilość wystąpień danej litery bez względu na jej wielkość.
- Metodę obliczającą ilość wystąpień określonej frazy. Metoda powinna mieć dwie wersje, jedną uwzględniającą wielkość liter, drugą obliczającą ilość wystąpień danej frazy bez względu na wielkość liter ją tworzących.
- Metodę obliczającą ilość wystąpień liter.
- Metodę obliczającą ilość wystąpień samogłosek.
- Metodę obliczającą ilość wystąpień spółgłosek.
- Metodę obliczającą ilość wystąpień cyfr.
- Metodę obliczającą ilość wystąpień znaków alfanumerycznych (cyfr i liter).
- Metodę obliczającą ilość wystąpień znaków nie-alfanumerycznych.
- Metodę obliczającą ilość wystąpień znaków białych (spacja, tabulacja, przejście do nowej linii).
- Metodę obliczającą ilość wystąpień liter małych.
- Metodę obliczającą ilość wystąpień liter wielkich.

Zad. 3 (Właściwości)

Twoim zadaniem jest stworzenie biblioteki, który ułatwi posługiwanie się różnymi jednostkami dla tych samych pojęć. Biblioteka ta będzie później używana różnych projektach. Powinieneś

uwzględnić następujące pojęcia i ich jednostki:

- ilość
 - kopa - 60
 - tuzin - 12
 - mendel - 15
- długość
 - metr
 - milimetr - 0,001 metra
 - cal - 25,4 milimetra
 - yard - 0,9144 metra
 - mila morska - 1 852 metrów
 - mila angielska - 1609,344 metra
- masa
 - kilogram
 - gram - 0,001 kilograma
 - tona - 1000 kilogramów
 - kwintal - 100 kilogramów
 - funt brytyjski - 0,453592
 - uncja - 28,35 grama
- temperatura
 - Kelvin
 - stopnie Celsjusza - stopnie w skali Kelvina – 273,15
 - stopnie Fahrenheita – $32 + 9/5 \cdot \text{stopnie w skali Celsjusza}$
- moc
 - wat
 - koń mechaniczny - 735,498
- objętość
 - metr sześcienny
 - litr - 0,001 metra sześciennego
 - kwaterka - 0,25 litra, inaczej ćwiartka
 - galon angielski - 4,5456 litra

Przy pomocy utworzonego mechanizmu utwórz program przeliczający jednostki.

Zad. 4

Utworzyć klasę wektor, której obiekty będą stanowić wektory(tablice) liczb rzeczywistych i dostarczyć w niej następujących metod i konstruktorów:

1. konstruktor tworzący wektor zerowy o rozmiarze **n** (**n** – argument konstruktora)
2. konstruktor tworzący wektor z podanej jako argument tablicy liczb rzeczywistych
3. metoda **get**, która zwróci *i*-ty element wektora
4. metoda **set**, która ustala *i*-ty element wektora na podaną wartość,
5. metoda **max**, która zwraca maksymalny element wektora,
6. metoda **min**, która zwraca minimalny element wektora,
7. metoda **add**, która zwraca wektor, będący sumą poelementową dwóch wektorów
8. metoda **size**, która zwraca liczbę elementów wektora
9. metoda **toString**, która zwraca łańcuch znakowy przedstawiający elementy wektora
10. statyczna metoda **show(double[])**, która wyprowadza w konsoli w jednym wierszu elementy przekazanej jako argument tablicy
11. metoda **sort**, która zwraca posortowany wektor w porządku rosnącym lub malejącym (to zależy od argumentu)

Zad.1

Dostałeś nowe zadanie zaprojektowania i utworzenie biblioteki figur geometrycznych. Każda z figur musi posiadać nazwę w celu jej identyfikacji oraz metodę zwracającą napis opisującą daną figurę. W przypadku figur, dla których jest to możliwe, powiniene również umożliwienie obliczenie długości obwodu oraz pola powierzchni. Utwórz również program testujący utworzone figury. Figury, które powinieneś zaimplementować na początku to:

- odcinek
- koło
- trójkąt
- kwadrat

Zwróć uwagę, że trójkąt oraz kwadrat są wielokątami. Każdy wielokąt posiada takie cechy jak:

- współrzędne wierzchołków wielokąta
- liczba boków
- długość każdego boku
- posiada obwód oraz pole

Uwaga:

Przy implementacji klasy reprezentującej kwadrat, możesz założyć, że boki kwadratu są równoległe do osi układu współrzędnych. Możesz uznać, że punkty leżące na jednej prostej tworzą również trójkąt. Nie musisz sprawdzać, czy punkty są współliniowe.

Do obliczenia pola trójkąta skorzystaj ze wzoru Herona:

$$P = \sqrt{p(p-a)(p-b)(p-c)}$$

gdzie:

a, b, c – długości boków trójkąta

p – połowa długości obwodu

Zad.2

Twoim zadaniem jest zaimplementowanie i przetestowanie hierarchii składającej się z następujących klas: *Osoba*, *Student*, *Wykładowca* oraz *Stypendysta*. Poszczególne klasy zawierają następujące cechy:

- *Osoba*: imię, nazwisko, rok urodzenia oraz płeć
- *Student*: zawiera wszystkie cechy zawarte w klasie *Osoba* oraz numer indeksu
- *Wykładowca*: zawiera wszystkie cechy zawarte w klasie *Osoba* oraz tytuł (tytuł naukowy lub stopień naukowy)
- *Stypendysta*: zawiera wszystkie cechy zawarte w klasie *Student* oraz kwotę stypendium

Każda klasa powinna również zawierać:

- zestaw odpowiednich konstruktorów
- zestaw właściwości umożliwiający zmianę i odczyt odpowiednich pól
- metody zwracającej w postaci napisu pełną informację zawartą w danej klasie

Zad.4

Firma w której pracujesz dostała zlecenie na napisanie programu wspierającego pracę biblioteki. Biblioteka oprócz książek w swoich zbiorach zawiera również czasopisma, płyty CD i DVD zarówno audio jak i programy komputerowe oraz kasety audio i VHS. O każdym z materiałów bibliotecznych potrzebujesz następujących informacji:

- książka: imię i nazwisko autora ewentualnie autorów, tytuł książki, rok wydania, nazwę wydawnictwa, numer ISBN
- czasopismo: tytuł czasopisma, numer, rok wydania, czy jest to kwartalnik, miesięcznik itp.
- płyta CD/DVD audio: tytuł płyty, spis wykonawców, spis utworów, typ nośnika (DVD czy CD)
- płyta CD/DVD programy komputerowe: nazwa programu, nazwa firmy która jest właścicielem programu
- kasecie audio: tytuł kasety, spis wykonawców, spis utworów
- kasecie VHS: tytuł filmu, nazwisko reżysera

Ponieważ program jest potrzebny już, zdecydowano, że pierwsza wersja programu będzie umożliwiać ewidencję tylko książek i czasopism. Funkcjonalność obejmująca płyty DVD i CD oraz kasety zostanie dodana w następnej wersji oprogramowania. Twoim pierwszym zadaniem jest opracowanie projektu programu, tak aby był łatwo rozszerzalny o nowe rodzaje materiałów bibliotecznych. Drugim zadaniem jest implementacja programu, który będzie uwzględniał książki i czasopisma.

Program powinien obejmować następującą funkcjonalność:

- dodawanie dowolnego materiału bibliotecznego do bazy
- zapis zaewidencjonowanych materiałów bibliotecznych do pliku (ewentualnie do bazy danych)
- odczytywanie bazy materiałów bibliotecznych z pliku (ewentualnie do bazy danych)
- wyświetlanie informacji o zaewidencjonowanych materiałach bibliotecznych