

Programowanie Zawansowane 2

Lab 1 – console App

1. Napisz program sprawdzający czy podany przez użytkownika rok jest przestępny.
2. Oblicz sumę wszystkich cyfr występujących w podanej przez użytkownika dodatniej liczbie.
3. Felek napisał na kartce n liczb. Zastanawia się teraz, czy są one permutacją liczb od 1 do n , czyli czy każda z liczb 1, 2, 3 ... $n-1$, n , występuje dokładnie jeden raz w tym ciągu.

Wejście

Pierwszy wiersz standardowego wejścia zawiera jedną liczbę całkowitą n ($n < 1000$), oznaczającą ilość liczb jakie wypisał Felek. Kolejny wiersz zawiera ciąg n liczb całkowitych odseparowanych spacją.

Wyjście

Pierwszy i jedyny wiersz wyjścia powinien zawierać słowo 'TAK', jeśli ciąg Felka jest permutacją liczb od 1 do n , lub słowo 'NIE', jeśli ciąg Felka nie jest permutacją liczb od 1 do n .

Przykład

Dla danych wejściowych:

5

1 4 3 2 5

poprawną odpowiedzią jest:

TAK

4. Felek postanowił się odchudzić. Jest po n dniach diety i intensywnego treningu, jednak waga nie spadała mu równomiernie, a czasem nawet (ku zdziwieniu Felka) zwiększała się. Felek codziennie zapisywał swoją wagę i teraz chce się pochwalić kolegom, więc wybierze taki fragment swojego dzienniczka, w którym schudł najbardziej. Znajdź ten fragment i policz, ile w nim schudł (czyli oblicz maksymalny spadek wagi Felka).

Wejście: W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n . W drugim wierszu wejścia znajduje się n liczb całkowitych oznaczających wagę Felka w kolejnym dniu diety.

Wyjście: W pierwszym i jedynym wierszu wyjścia powinna być jedna liczba całkowita, oznaczająca maksymalny spadek wagi Felka.

Przykład: Dla danych wejściowych:

5

6 9 5 4 2

poprawną odpowiedzią jest: 7

5. Napisać program podający statystykę wystąpienia poszczególnych liter w tekście podanym z klawiatury. Rozmiar litery nie ma znaczenia. Podajemy statystykę tylko tych znaków, które wystąpiły w podanym tekście. Zadanie rozwiązań stosując algorytm o złożoności liniowej.

6. Napisać program zamieniający podane fragmenty tekstu na wzorzec podany z klawiatury. Program najpierw prosi o podanie dowolnego tekstu z klawiatury. Następnie prosi o podanie który fragment tekstu chcesz zamienić podać tekst szukany. Następnie podajemy tekst, który chcemy w to miejsce wstawić.

7. Napisać program, który pozwoli zapisywać informacje o studentach. Ilość studentów znana jest dopiero w momencie uruchomienia programu. Każdy student opisany jest za pomocą nazwy kierunku, roku studiów oraz wyników egzaminów. Ilość egzaminów dla każdego ze studentów może być różna. Zaprojektować odpowiednią strukturę opisującą studenta a następnie zaimplementować program zgodnie z powyższym opisem. Proszę starać się używać do implementacji kolekcji omawianych na wykładzie.

8. Utwórz podstawy systemu zarządzania pracownikami w pewnej firmie. Zaimplementuj klasę reprezentującą pracownika.

Każdy pracownik opisany jest właściwościami:

- imię i nazwisko
- kontrakt

Klasa reprezentująca pracownika udostępnia operacje:

- konstruktor inicjujący pracownika o podanym w argumentach imieniu i nazwisku z domyślnym kontraktem stażysty
- metodę pozwalającą zmienić kontrakt przypisany do pracownika
- metoda zwracająca wysokość pensji pracownika uzależnionej od podpisanego kontraktu
- dociążoną metodę `ToString()` zwracającą łańcuch znakowy zawierający imię, nazwisko i wysokość pensji pracownika

Każdy kontrakt reprezentowany jest przez obiekt udostępniający publicznie następujące operacje:

- metodę o nazwie `Pensja()`, która zwraca wysokość pensji wypłacanej przy danym kontrakcie.
- konstruktor pozwalający zainicjować wszystkie pola składowe obiektu
- konstruktor domyślny

Obecnie w firmie podpisywane są 2 rodzaje kontraktów: **staż** i **etat**.

Kontrakty określone są przez następujące własności:

- **Staż**: stawka miesięczna (domyślnie 1000 zł)
- **Etat**: stawka miesięczna (domyślnie 5000 zł), ilość nadgodzin (domyślnie 0)

Wysokość pensji wyznaczana jest dla każdego z tych kontraktów inaczej, odpowiednio:

- Pensja **stażysty** równa jest stawce miesięcznej
- Pensja **pracownika etatowego** = stawka miesięczna + ilość nadgodzin * (stawka miesięczna/60)

Zakładamy, że w przyszłości pojawią się inne typy kontraktów różniące się sposobem naliczania wysokości pensji. Za obliczanie pensji odpowiedzialny jest obiekt klasy `kontrakt`, który udostępnia metodę publiczną zwracającą wysokość wynagrodzenia za pomocą metody `Pensja()`. Dlatego, dodanie nowego typu kontraktu sprowadza się do zaimplementowania tylko jednej klasy, bez potrzeby modyfikowania istniejących już klas.

9. Mamy 4 klasy dziedziczące po klasie `Ssak`: `Krowa`, `Kotek` oraz `Owca`. Każde z tych zwierząt posiada atrybut `waga` oraz metody `SprawdzWage()` i `PijWode()`. Wszystkie zwierzęta oprócz `Kotka` mają również zaimplementowaną metodę `JemSiano()`.

`Krowa` pije 3l wody i je 80 jednostek siana. `Owca` pije 2l wody i je 40 jednostek siana. `Kotek` pije 0.5L wody oraz ma zaimplementowaną metodę `JemWszystkoOproczSiana()`.

Zaprojektuj klasy a następnie je zaimplementuj. Przy tworzeniu każdego ze zwierząt należy ustawić jego wagę. Możesz zdefiniować także inne metody jeśli mogą być dla Ciebie użyteczne.

Nasza farma liczy 5 miejsc hodowlanych. Zapełnij ją całkowicie dowolnymi zwierzętami a następnie podaj mi następujące informacje:

- Ile wody zostanie wypite podczas jednego pojenia,
- Ile zostanie zjedzonego siana podczas jednego karmienia;
- Ile waży najlżejsze zwierzę