Zadanie 1.
Zdefiniuj klasę `TOsoba` posiadającą następujące pola:

- `imie`
- `nazwisko`
- `wiek`
- `adres` (obiekt niezależnej klasy o polach: `miasto`, `ulica`, `numer` z konstruktorem ustawiającym pola na podstawie parametrów, destruktorem oraz metodą `Wyswietl`).

Klasa zawiera następujące metody:

- konstruktor ustawiający wszystkie pola na podstawie swoich parametrów,
- `Wyswietl` – wyświetlającą informacje o osobie,
- `PodajImie` – zwraca imię,
- `PodajNazwisko` – zwraca nazwisko,
- destruktor.

Zdefiniuj klasę `TStudent` dziedziczącą po klasie `TOsoba`. Klasa `TStudent` posiada dodatkowo pola:

- `kierunek`
- `rok`

oraz metody:

- konstruktor ustawiający wszystkie pola na podstawie swoich parametrów,
- `Wyswietl` – wyświetlającą informacje o studencie,
- `PodajKierunek` – zwraca kierunek,
- destruktor.

Dla każdej klasy zdefiniuj konstruktory i destruktory tak, aby wyświetlały komunikaty informujące o swoim działaniu.
Napisz funkcję `Testuj`, która:
a) definiuje zmienną lokalna `st` typu `TStudent`, wyświetla dane studenta,
b) wykorzystując wskaźnik na typ bazowy tworzy zmienną dynamiczną typu `TStudent`, wyświetla dane studenta,
c) definiuje zmienną referencyjną typu `TOsoba`, inicjalizuje ją zmienną `st` wyświetla wartości pól,
d) wyświetla nazwiska i kierunki studentów z podpunktu a) i b),
e) usuwa zmienną dynamiczną z podpunktu b).

Zadanie 2.
Napisz klasę `TKomorka` odzwierciedlającą działanie telefonu komórkowego. Klasa ta powinna posiadać następujące pola prywatne:

- `koszt_minuty` – koszt minuty rozmowy (pole stałe typu rzeczywistego)
- `koszt_smsa` – koszt wysłania SMS-a (pole stałe typu rzeczywistego)
- `czas_rozmow` – łączny czas wykonanych rozmów w sekundach (pole typu całkowitego)
- `ilosc_smsow` – łączna ilość wysłanych SMS-ów (pole typu całkowitego)

oraz metody:

- Konstruktor - inicjujący pola `koszt_minuty` i `koszt_smsa` wartościami swoich parametrów oraz zerujący pola `czas_rozmow` i `ilosc_smsow`;
- `Rozmawiaj` – przyjmująca jako parametr czas wykonanej rozmowy w sekundach i zwiększająca o ten czas wartość pola `czas_rozmow`;
- `WyslijWiadomosc` – przyjmująca jako parametr ilość SMS-ów składających się na wysłaną wiadomość tekstową i zwiększająca o tą ilość wartość pola `ilosc_smsow`. Zakładamy, że wiadomość nie może składać się z więcej niż 6 SMS-ów. Jeśli wiadomość będzie dłuższa powinien zostać wyświetlony odpowiedni komunikat ostrzegawczy.
- `ZerujLiczniki` – zerująca czas wszystkich przeprowadzonych rozmów i ilość wszystkich wysłanych SMS-ów;
- `KosztCalkowity` – zwracająca całkowity koszt przeprowadzonych rozmów i wysłanych SMS-ów z uwzględnieniem faktu, iż co dziesiąty SMS jest gratis;
- `WystawRachunek` – korzystająca z metody `KosztCalkowity` i wyświetlająca na ekranie

informację o tym ile należy zapłacić za używanie telefonu, a następnie zerująca liczniki rozmów i SMS-ów.

Napisz następnie klasę potomną `TKomorkaZAparatem`, która rozszerza funkcjonalność klasy `TKomorka` o możliwość robienia zdjęć i wysyłania ich jako MMS. Klasa ta powinna posiadać następujące pola:

- `koszt_mmsa` – koszt wysłania wiadomości multimedialnej (pole stałe typu rzeczywistego)
- `ilosc_mmsow` – ilość wysłanych MMS-ów (pole typu całkowitego)

oraz metody:

- Konstruktor - inicjujący pola klasy bazowej oraz pola własnej klasy odpowiednimi wartościami;
- `WyslijZdjecie` – odpowiadająca za wysłanie jednej wiadomości multimedialnej tj. większenie wartości pola `ilosc_mmsow` o jeden.

Metoda `ZerujLiczniki` powinna w tej klasie, oprócz zerowania liczników rozmów i SMS-ów, zerować licznik wysłanych wiadomości multimedialnych. Natomiast metoda `KosztCalkowity` powinna brać pod uwagę również koszt wysłanych MMS-ów, z uwzględnieniem takiej promocji, w której co piąty MMS kosztuje 1 grosz.

Napisz ciąg instrukcji pozwalających na wystawienie rachunku za przeprowadzenie dwóch trzydziestominutowych rozmów, wysłanie wiadomości tekstowej składającej się z 4 SMS-ów i wysłanie 3 wiadomości multimedialnych według cennika:

1 minuta rozmowy - 48 groszy 1 SMS - 20 grosze 1 MMS - 1 złote

Zadnie 3.

Zdefiniuj klasy opisujące pracownika zatrudnionego tylko na etat i pracownika mającego oprócz etatu godziny nadliczbowe. Dokładniej, klasa `TPracownik` pozwala na zapamiętanie następujących danych o pracowniku:

- `pesel` – ciąg cyfr, domyślnie pusty łańcuch tekstowy;
- `imie`, `nazwisko` – łańcuchy tekstowe;
- `pensja` – miesięczna pensja pracownika w groszach, liczba całkowita, domyślnie 0 gr.

oraz metody:

- `UstawPesel`, `UstawImie`, `UstawNazwisko`, `UstawPensja` – ustawiają wartości pól;
- `Brutto` – zwraca wartość pola `pensja`;
- `Potracenia` – wyznacza kwotę potrąceń w groszach, 30% z `Brutto`;
- `DoWyplaty` – wyświetla na ekranie imię, nazwisko, kwotę potrąceń, kwotę do wypłaty; kwoty należy wyświetlić w złotych i groszach zawsze z dwiema cyframi groszy;
- `Podwyzka` – pozwalająca na zwiększenie pensji o zadaną kwotę w groszach.

Klasa `TPracownikZN` jest klasą potomną klasy `TPracownik` posiadająca dodatkowo pola:

- `czas` – ilość godzin nadliczbowych – liczba rzeczywista;
- `stawka` – stawka za jedną godzinę nadliczbową w groszach

oraz metody:

- `UstawCzas` – ustawia wartości pola `czas`;
- `UstawStawka` – ustawia wartości pola `stawka`;
- `Brutto` – zwraca wartość odziedziczonej metody `Brutto` zwiększoną o `czas*stawka` (w groszach)
- `Potracenia` – oblicza potrącenia jako sumę potrącenia z typu bazowego plus 20% z kwoty `czas* stawka`;

Odziedziczona metoda `DoWyplaty` powinna wyświetlić kwotę potrąceń i kwotę do wypłaty pracownika mającego godziny nadliczbowe.

Korzystając ze wskaźników na typ bazowy zadeklaruj pracowników: Jan Kowalski pesel: 6012134913 z pensją 1200 zł oraz Marian Nowicki pesel 72110334254 z pensją 1095 zł, 45.5 godz. nadliczbowymi płatnymi po 12 zł. Następnie Kowalskiemu podnieś pensję o 133 zł 23 gr. Wyświetl dla obu pracowników kwotę potrąceń i kwotę do wypłaty.