

Celem zadania jest prosta **obsługa bankowa**: wyliczania **stałej regularnej raty spłaty kredytu** oraz **prognozy** ile pozostało do spłacenia po wpłaceniu i-tej raty.

Celem technicznym zadania jest **przećwiczenie podstawowych konstrukcji** języka c++ stosowanych w **implementacji klas ze wspólnej hierarchii polimorficznej**.

Klasa **creditDet** jest **publiczną** pochodną klasy bazowej **credit**. Obiekty obu klas będą obsługiwane za pomocą wskaźników lub referencji klasy bazowej **credit**

Funkcja **main()** jest przygotowana w **całości** (*zadanie uruchamiaj kolejnymi etapami*). Porównaj swoje wyniki z wynikami zamieszczonymi w pliku **pdf** zadania.

ETAP 1

Zaimplementuj wymagane składowe dla klasy **credit** (tylko **niezbędne**, uzasadnij)

Tylko w tej klasie zaimplementuj **operator<<** (zdefiniuj metodę **write**, którą w nim wywołasz)

Jeśli już teraz widzisz co powinno być wirtualne w tej klasie, to możesz o to zadbać.

Obliczanie stałej raty kredytu (payment)

```
r=value*q^n*(q-1)/(q^n-1)
```

gdzie $n = \text{payments_in_years} * \text{years}$, $q = 1 + p$, ^-oznacza potęgowanie

gdzie **p** oprocentowanie, np. dla $\text{percent} = 3(\%)$ oraz $\text{years} = 12$ mamy $p = 0.0025$

ETAP 2

Zaimplementuj wymagane składowe dla klasy **creditDet** (tylko **niezbędne**, uzasadnij)

W tej klasie **nie** definiujemy **operator<<**

W implementacji klasy wykorzystaj metodę **init_projection**, która alokuje pamięć na dynamiczną tablicę **projection** oraz ją wypełnia (w przypadku braku podanych wartości zeruje).

W konstruktorze klasy wywołaj metodę **init_projection** oraz **rozlicz**, która wypełni tablicę **projection**. Dla elementu tablicy wpisujemy ile kredytu (rzeczywistego) pozostało do spłacenia po i-tej racie (czyli na koniec 0)

Rozliczanie kredytu (payoff) (i-ty element tablicy projection)

```
w=value
dla j=0..i (włącznie)
    odsetki=w*p
    w-=r-odsetki      // gdzie r - rata spłaty kredytu (payment)
projection[i]=w;
```

ETAP 3

Jest **dobra** i **zła** wiadomość ;)...

Jeśli przewidziałeś odpowiednie zachowanie klas, to ten etap masz gotowy.

Jeśli nie widzisz pełnych danych w swojej konsoli, to **popraw implementację** klas **credit** lub **creditDet** tak, by to uzyskać (ale bez modyfikacji f.main)

Czy obiekty prawidłowo są **niszczone**? uzasadnij

ETAP 4

Napisz **własną** funkcję **sort** (w implementacji **NIE** wolno wykorzystywać żadnych innych funkcji, poza przekazanym w parametrze kryterium sortowania), by sortować tablicę kredytów wg **różnych** kryteriów, tutaj dwa: **cmp_value** (wg wartości kredytu) oraz **cmp_value_info** (w pierwszej kolejności decyduje wartość kredytu, a gdy są takie same to decyduje info)

Kryterium będzie trzecim parametrem funkcji i reprezentuje adres funkcji porównującej o nagłówku postaci:

```
bool cmp(const credit& k1, const credit& k2)
```

ETAP 5

Dopisz do klas wirtualną metodę **clone** odpowiedzialną za tworzenie “głębokich” kopii obiektów klas (*jawne sprawdzanie typów obiektów jest zabronione*).

Przetestuj

ETAP 1 - test class credit

```
credits:
k1:
P 23/2017  credit:    1000.00 ( 9.00%)  years: 5   r/r: 12  payment: 20.76
k2:
P 23/2017  credit:    1000.00 ( 9.00%)  years: 5   r/r: 12  payment: 20.76
k3:
P 23/2017  credit:    1000.00 ( 9.00%)  years: 5   r/r: 12  payment: 20.76
k2=k4
P 10/2021  credit:     500.00 ( 3.00%)  years: 2   r/r: 20  payment: 12.89
```

ETAP 2 - test class creditDet

```
creditDet:
kF1:
K 45/2017  credit:   10000.00 ( 4.00%)  years: 2   r/r: 3   payment: 1745.30
           projection: [ 8388.03 6754.57 5099.33 3422.01 1722.34 0.00 ]
kF2:
K 45/2017  credit:   10000.00 ( 4.00%)  years: 2   r/r: 3   payment: 1745.30
           projection: [ 8388.03 6754.57 5099.33 3422.01 1722.34 0.00 ]
k3:
K 45/2017  credit:   10000.00 ( 4.00%)  years: 2   r/r: 3   payment: 1745.30
           projection: [ 8388.03 6754.57 5099.33 3422.01 1722.34 0.00 ]
*p:
K 50/2021  credit:   20000.00 ( 4.00%)  years: 2   r/r: 3   payment: 3490.61
           projection: [ 16776.06 13509.14 10198.65 6844.03 3444.68 0.00 ]
```

ETAP 3

```
credits:
K 23/2021  credit:   10000.00 ( 3.00%)  years: 2   r/r: 3   payment: 1725.48
           projection: [ 8374.52 6732.78 5074.62 3399.88 1708.40 0.00 ]
R 15/2021  credit:    1000.00 ( 4.00%)  years: 1   r/r: 4   payment: 256.28
           projection: [ 753.72 504.98 253.74 -0.00 ]
xxxxxxxxxx credit:   10000.00 ( 7.50%)  years: 2   r/r: 4   payment: 1357.75
K 37/2021  credit:    1000.00 ( 3.00%)  years: 1   r/r: 12  payment: 84.69
           projection: [ 917.81 835.41 752.80 669.99 586.97 503.75 420.31 336.67
252.82 168.75 84.48 0.00 ]
xxxxxxxxxx credit:   50000.00 ( 4.50%)  years: 4   r/r: 2   payment: 6899.23
```

projection: [44225.77 38321.62 32284.62 26111.80 19800.08 13346.35
 6747.41 0.00]
 R 50/2021 credit: 5000.00 (5.00%) years: 2 r/r: 4 payment: 660.67
 K 35/2021 credit: 1000.00 (3.00%) years: 3 r/r: 2 payment: 175.53
 projection: [839.47 676.54 511.16 343.31 172.93 -0.00]
 K 10/2021 credit: 1000.00 (4.00%) years: 2 r/r: 3 payment: 174.53
 K 20/2021 credit: 100.00 (9.00%) years: 1 r/r: 4 payment: 26.42
 projection: [75.83 51.11 25.84 -0.00]
 P 22/2021 credit: 1000.00 (7.00%) years: 2 r/r: 3 payment: 180.54

ETAP 4 - sortowanie

Obejrzyj samodzielnie, czy ok ?.

ETAP 5 - klonowanie

Obejrzyj samodzielnie, czy ok ?.