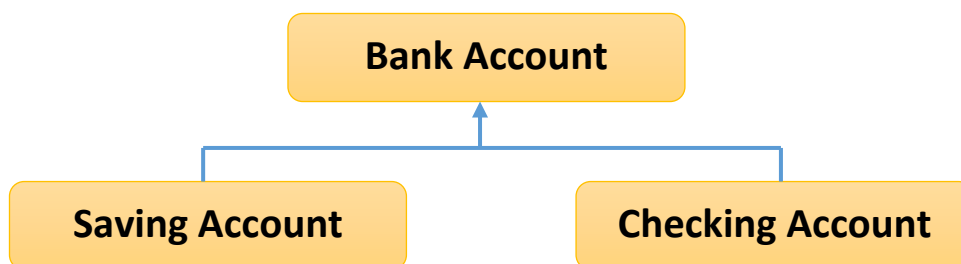


Задача 5: Интелигентен банкомат (Предефиниране)

Банка предлага нов вид банкомат (Интelligentен банкомат), чрез който може да се създават два вида банкови сметки:

- **Спестовна сметка, чрез която клиентът печели лихва.** Лихвите се съставят ежемесечно и се изчисляват въз основа на **минималната месечна сума** по сметката.
- **Разплащателна сметка, чрез която клиентът не печели нищо.** Няма лихва, но дава възможност за три безплатни тегления месечно от банкомата на банката и таксува по 1 лев всяка следваща транзакция.

Да се дефинира следната йерархия от класове:



Базовият клас *BankAccount* има следните задачи:

- дава възможност на клиента да депозира или да тегли пари от сметката си;
- отпечатва текущия баланс по сметката.

Този клас (*BankAccount.cs*) ви е даден.

Производният клас *SavingAccount* изпълнява задачи, които са специфични за действията със спестовна банкова сметка:

- депозирание (вносяне) на пари по сметката;
- теглене на пари;
- изчисляване на текущ баланс по сметката;
- изчисляване на месечна лихва.

Производният клас *CheckingAccount* има отговорности, които са специфични за действията с разплащателна банкова сметка:

- депозирание (вносяне) на пари по сметката;
- теглене на пари (характерно в този случай е три безплатни тегления и удържане на 1 лев за всяко следващо теглене);
- изчисляване на текущ баланс по сметката;
- нулиране на месечните тегления (всеки нов месец позволява нови три безплатни транзакции).

Поле на класа *BankAccount*:

Тип	Име	Описание
реално число/protected	balance	баланс по сметка

Конструктор `BankAccount()` – задава стойност 0 на полето *balance*.

Конструктор `BankAccount(double b)` – копира стойността на параметъра си *b* в полето *balance*.

Метод `virtual void Deposit(double amount)` – обновява баланса по сметката като добавя стойността на параметъра *amount* (сумата, която клиентът е депозирал) към текущия баланс. Отпечатва текущия баланс.

Метод `virtual void Withdraw(double amount)` – разрешава тегленето само ако наличността в сметката е достатъчна и обновява баланса (изважда стойността на *amount*). При недостатъчна наличност се извежда подходящо съобщение.

Метод `void PrintBalance()` – отпечатва текущия баланс по сметката (с два знака след десетичния разделител).

Полета на класа *SavingAccount*:

Тип	Име	Описание
реално число/private	interestRate	лихвен процент
реално число/private	minBalance	минимален баланс

Конструктор `SavingAccount()` – съхранява стойност 0 в полетата на класа и извежда съобщението “You have created a Saving Account!”.

Конструктор `SavingAccount(double rate, double mnBalance, double balance)` – копира стойностите на параметрите си в полетата на класа. Преценете за кой/кои параметри да използвате конструкцията *base*. Извежда съобщението “You have created a Saving Account!”.

Метод `override void Withdraw(double amount)` – организира теглене на посочената сума от сметката и евентуално актуализира минималната сума. Отпечатва текущия баланс.

Метод `void MonthEnd()` – изчислява месечната лихва, която се депозира в сметката. Отпечатва текущия баланс.

Поле на класа *CheckingAccount*:

Тип	Име	Описание
цяло число/protected	withdrawals	брой на тегленията

Конструктор `CheckingAccount()` – съхранява стойност 0 в полето на класа и извежда съобщението “You have created a Checking Account!”.

Конструктор `CheckingAccount(int wdr, double balance)` – копира стойностите на параметрите си в полетата на класа. Преценете за кой/кои параметри да използвате конструкцията *base*. Извежда съобщението “You have created a Checking Account!”.

Метод `override void Withdraw(double amount)` – организира теглене на посочената сума от сметката и проверява броя на тегленията. Ако тегленията са повече от 3, към всяко следващо теглене се налага такса от 1 лев. Отпечатва текущия баланс.

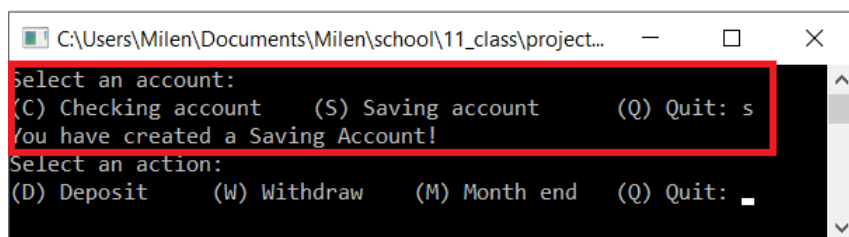
Метод `void NewMonth()` – нулира броя на тегленията (полето *withdrawals*) в „края на месеца“. Отпечатва текущия баланс.

Като се използват класовете *BankAccount*, *SavingAccount* и *CheckingAccount*, да се напише програмата, която:

- създава сметка от един от двата вида чрез потребителско меню:

(C) Checking account (S) Saving account (Q) Quit

Например:



```
C:\Users\Milen\Documents\Milen\school\11_class\project...
Select an account:
(C) Checking account    (S) Saving account    (Q) Quit: s
You have created a Saving Account!
Select an action:
(D) Deposit            (W) Withdraw          (M) Month end        (Q) Quit: _
```

- обработва различни типове банкови операции чрез потребителско меню:

(D) Deposit (W) Withdraw (M) Month end (Q) Quit

за спестовна сметка (Saving account) и

(D) Deposit (W) Withdraw (N) New month (Q) Quit

за разплащателна сметка (Checking account).

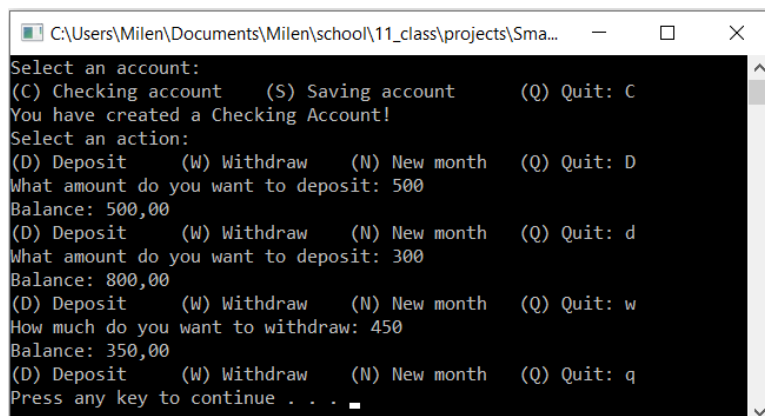
За създаването на спестовна сметка са използвани следните примерни данни:

SavingAccount sa = *new SavingAccount*(0.75, 10.0, 10.0);

0.75% лихвен процент (печалба);

10.00 лева минимален баланс при откриването на сметката;

10.00 лева текущ баланс по откритата сметка.



```
C:\Users\Milen\Documents\Milen\school\11_class\projects\Sma...
Select an account:
(C) Checking account    (S) Saving account    (Q) Quit: C
You have created a Checking Account!
Select an action:
(D) Deposit            (W) Withdraw          (N) New month        (Q) Quit: D
What amount do you want to deposit: 500
Balance: 500,00
(D) Deposit            (W) Withdraw          (N) New month        (Q) Quit: d
What amount do you want to deposit: 300
Balance: 800,00
(D) Deposit            (W) Withdraw          (N) New month        (Q) Quit: w
How much do you want to withdraw: 450
Balance: 350,00
(D) Deposit            (W) Withdraw          (N) New month        (Q) Quit: q
Press any key to continue . . . _
```

- 1) Създава се разплащателна сметка с избор на буква **C** от менюто.
- 2) Депозират се 500 лева с избор на буква **D** от менюто и въвеждане на 500 в следващия ред.
- 3) Отпечатва се текущият баланс, който е на стойност 500,00.
- 4) Депозират се още 300 лева с избор на буква **d** и въвеждане на 300 в следващия ред.

- 5) Отпечатва се текущият баланс, който е на стойност 800,00.
- 6) Изтеглят се 450 лева с избор на буква **W** от менюто и въвеждане на 450 в следващия ред.
- 7) Отпечатва се текущият баланс, който е на стойност 350,00.
- 8) Излиза се от програмата с избор на буква **Q** от менюто.

```

C:\Users\Milen\Documents\Milen\school\11_class\projects\Smar...
Select an account:
(C) Checking account    (S) Saving account    (Q) Quit: c
You have created a Checking Account!
Select an action:
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: d
What amount do you want to deposit: 190
Balance: 190,00
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: W
How much do you want to withdraw: 10
Balance: 180,00
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: w
How much do you want to withdraw: 20
Balance: 160,00
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: W
How much do you want to withdraw: 50
Balance: 110,00
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: W
How much do you want to withdraw: 10
Balance: 99,00
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: N
Balance: 99,00
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: W
How much do you want to withdraw: 10
Balance: 89,00
(D) Deposit    (W) Withdraw    (N) New month    (Q) Quit: q
Press any key to continue . . .

```

От горното изображение се вижда, че се създава разплащателна сметка, в която се депозират 190 лева. След четвъртото теглене се забелязва, че банкоматът удържа 1 лев за транзакцията. Изборът на командата New month (буква **N**) нулира тегленията (минава в следващия месец), след което при ново теглене не се удържа такса от 1 лев.

```

C:\Users\Milen\Documents\Milen\school\11_class\projects...
Select an account:
(C) Checking account    (S) Saving account    (Q) Quit: s
You have created a Saving Account!
Select an action:
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: d
What amount do you want to deposit: 100
Balance: 110,00
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: m
Balance: 110,08
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: q
Press any key to continue . . .

```

- 1) Създава се спестовна сметка с избор на буква **S** от менюто.
- 2) Депозират се 100 лева с избор на буква **D** от менюто и въвеждане на 100 в следващия ред.
- 3) Отпечатва се текущият баланс, който е на стойност 110,00 тъй като при откриване на сметката са вложени 10 лева (за да има сметка).
- 4) Избира се проверка на месечен баланс чрез буква **M** (Month end) от менюто.
- 5) Към текущия баланс се добавя печалба от 0.75% и така балансът става 110.08

6) Излиза се от програмата с избор на буква **Q** от менюто.

По-долу е даден още един пример.

```
C:\Users\Milen\Documents\Milen\school\11_class\projects\S...
Select an account:
(C) Checking account    (S) Saving account    (Q) Quit: S
You have created a Saving Account!
Select an action:
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: D
What amount do you want to deposit: 1000
Balance: 1010,00
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: W
How much do you want to withdraw: 10
Balance: 1000,00
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: W
How much do you want to withdraw: 100
Balance: 900,00
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: W
How much do you want to withdraw: 300
Balance: 600,00
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: W
How much do you want to withdraw: 110,50
Balance: 489,50
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: W
How much do you want to withdraw: 0,50
Balance: 489,00
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: D
What amount do you want to deposit: 1
Balance: 490,00
(D) Deposit    (W) Withdraw    (M) Month end    (Q) Quit: Q
Press any key to continue . . .
```