**Laboratorul 3.Programare bazata pe clase**

**Incapsularea folosind clase in limbajele c++,Python si java**

Exemplu:

Class: Un tip de dată definit de programator.

Descrie caracteristicile unui lucru.

Grupează:

• date – atribute

• comportament – metode

Clasa este definită într-un fișier header (.h)

Sintaxă:

/\*\*

\* Represent rational numbers

\*/

class Rational {

public:

//methods

/\*\*

\* Add an integer number to the rational number

\*/

void add(int val);

/\*\*

\* multiply with a rational number

\* r rational number

\*/

void mul(Rational r);

private:

//fields (members)

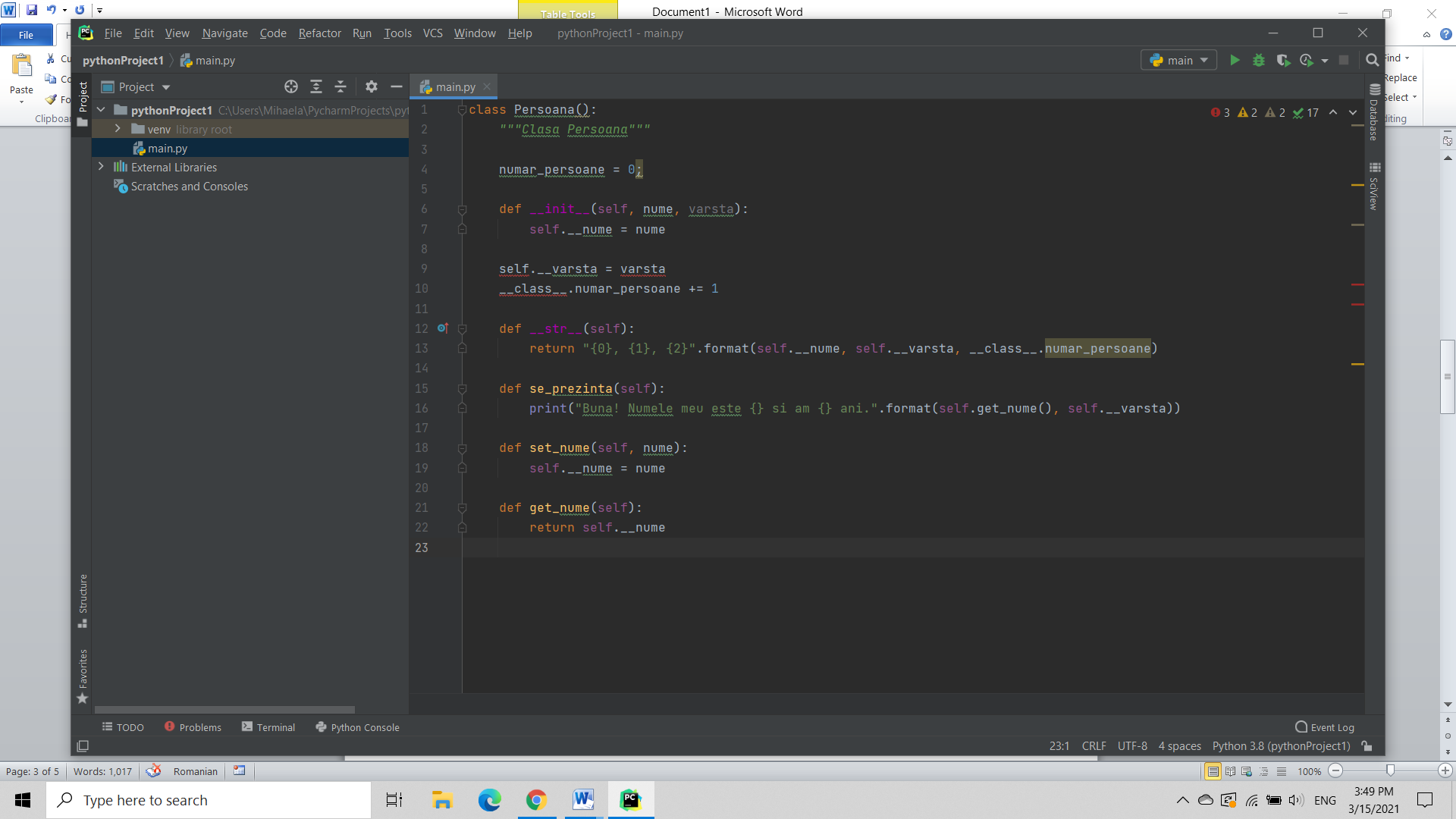
int a;

int b;

};

**Python**

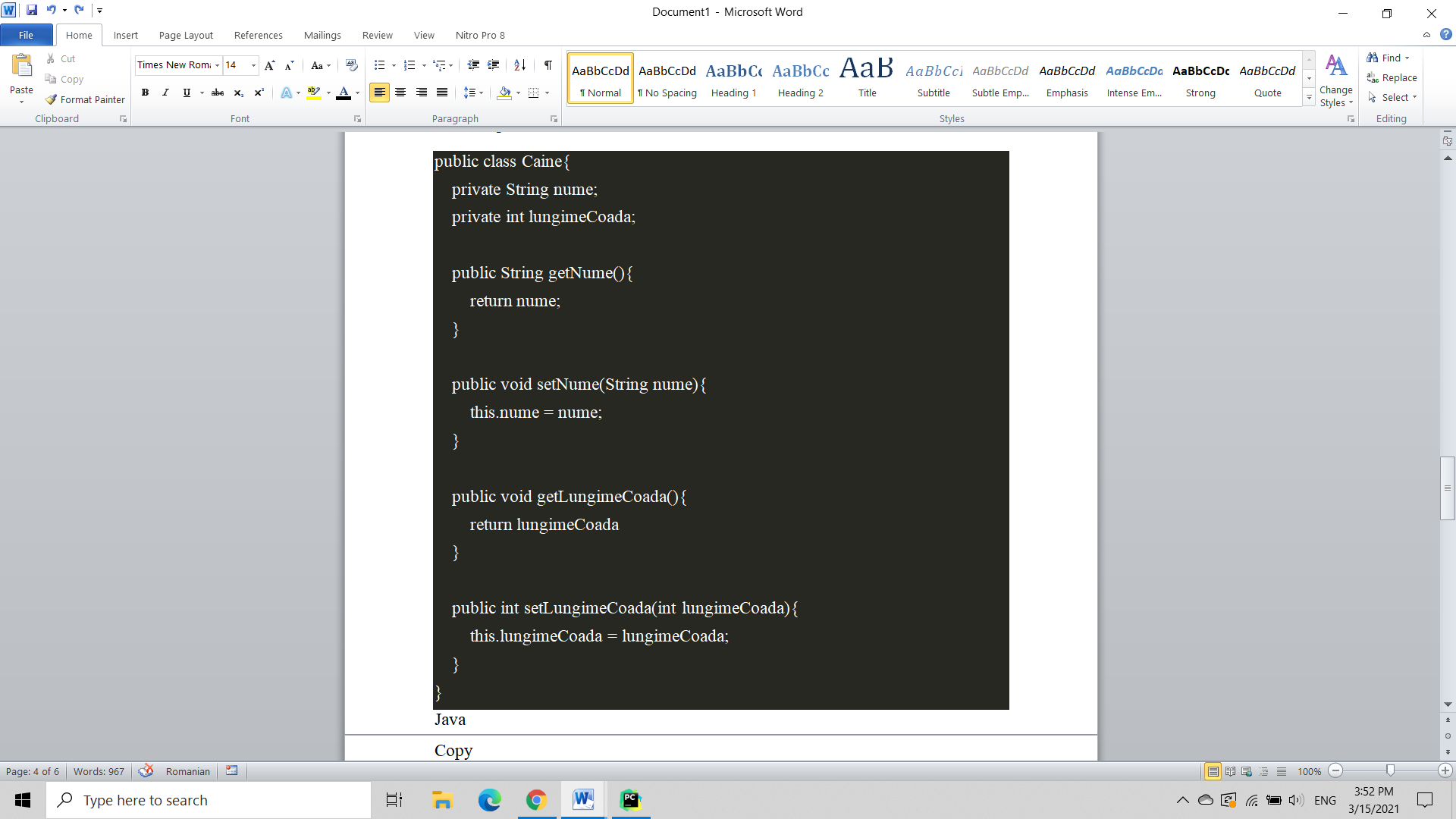
Exemplu:



Pentru citirea sau modificarea datelor membre protejate in clase pot fi definite metode de instante de tip set()/get(), care pot fi apelate din orice punct al domeniului de definitie al clasei.

**Java**

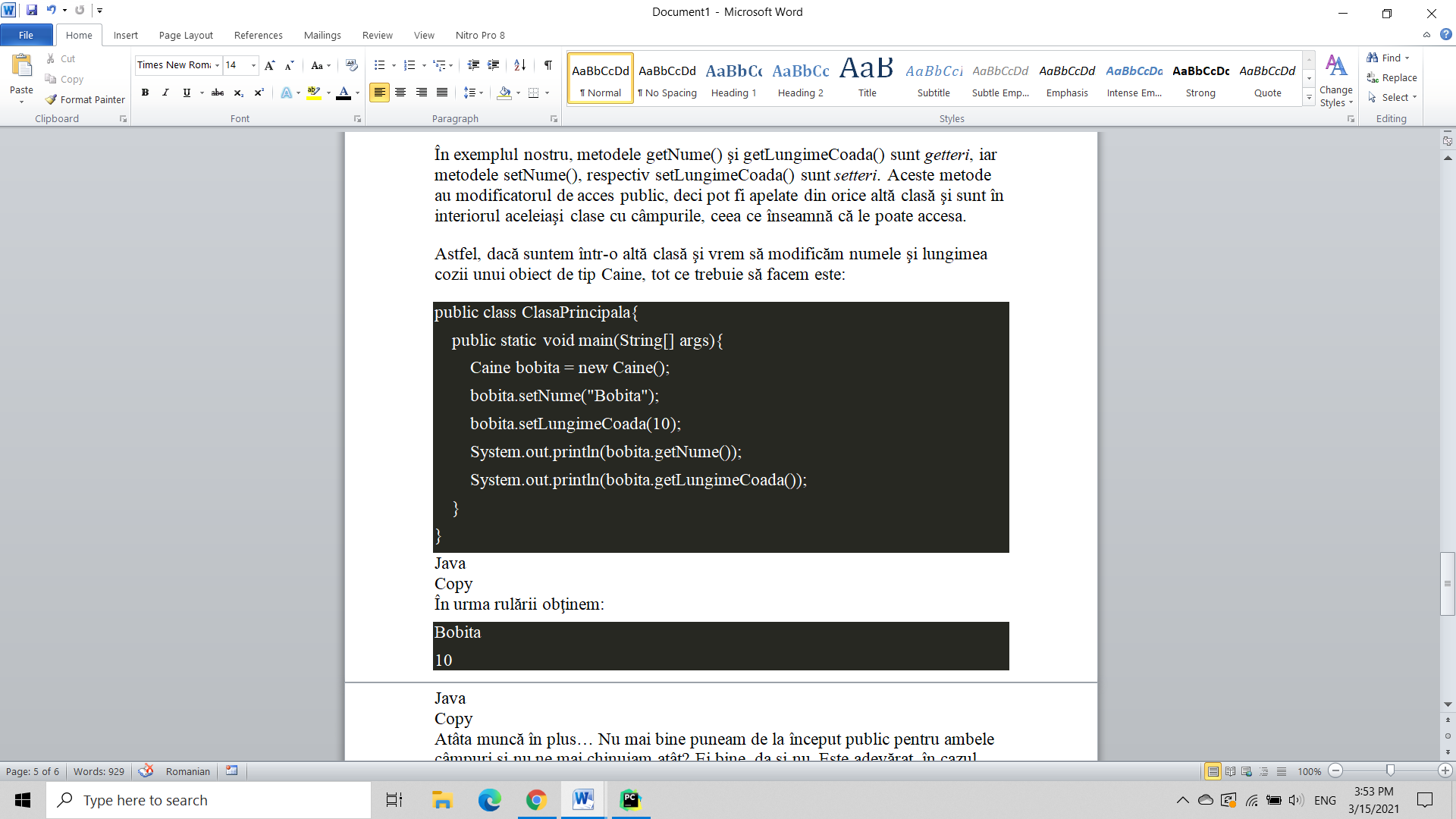
De exemplu:



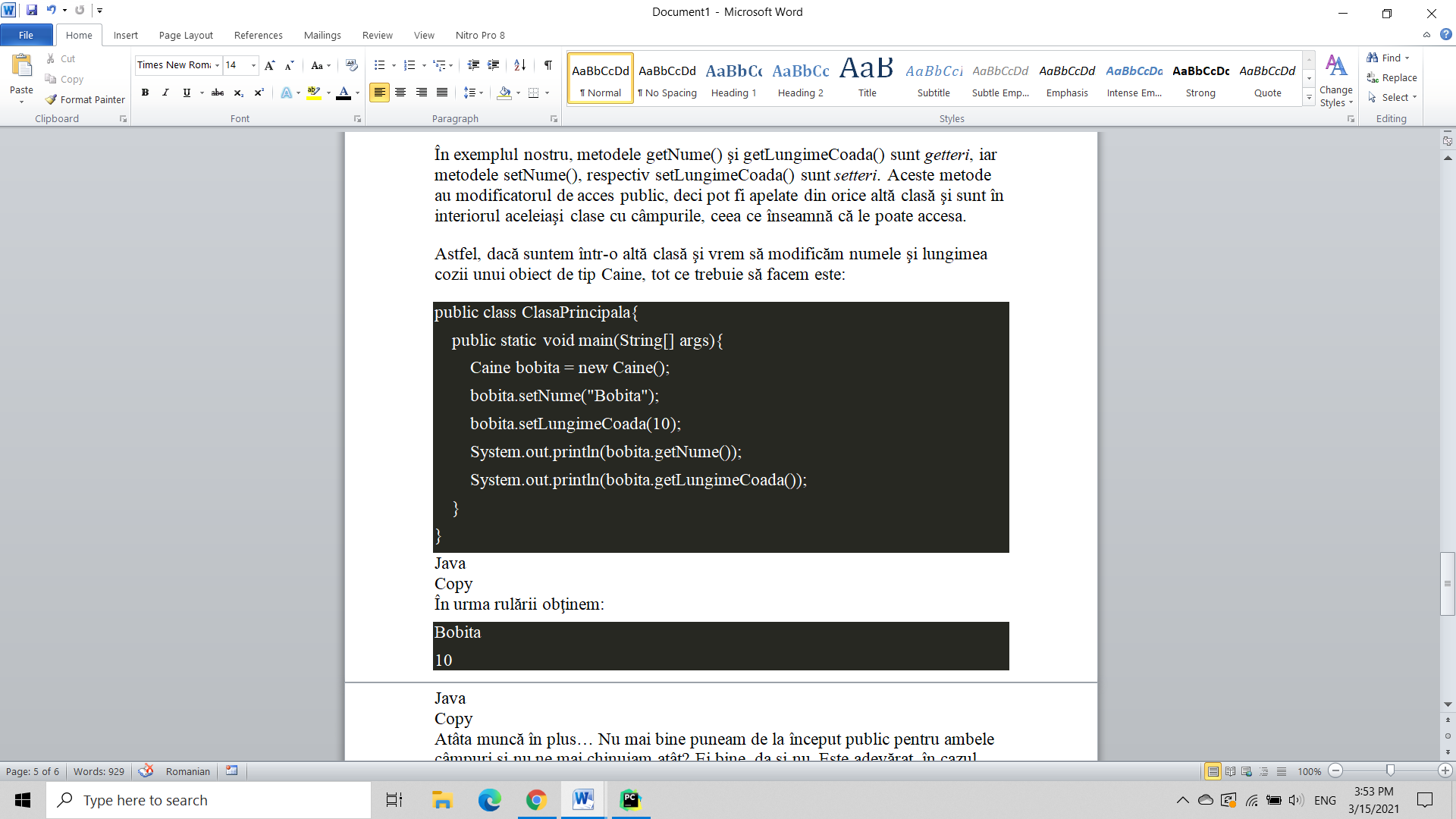
Metoda care returnează valoarea câmpului poartă numele de **getter**, iar cea care care modifică valoarea câmpului se numeşte **setter**.

În exemplul nostru, metodele getNume() şi getLungimeCoada() sunt *getteri*, iar metodele setNume(), respectiv setLungimeCoada() sunt *setteri*. Aceste metode au modificatorul de acces public, deci pot fi apelate din orice altă clasă şi sunt în interiorul aceleiaşi clase cu câmpurile, ceea ce înseamnă că le poate accesa.

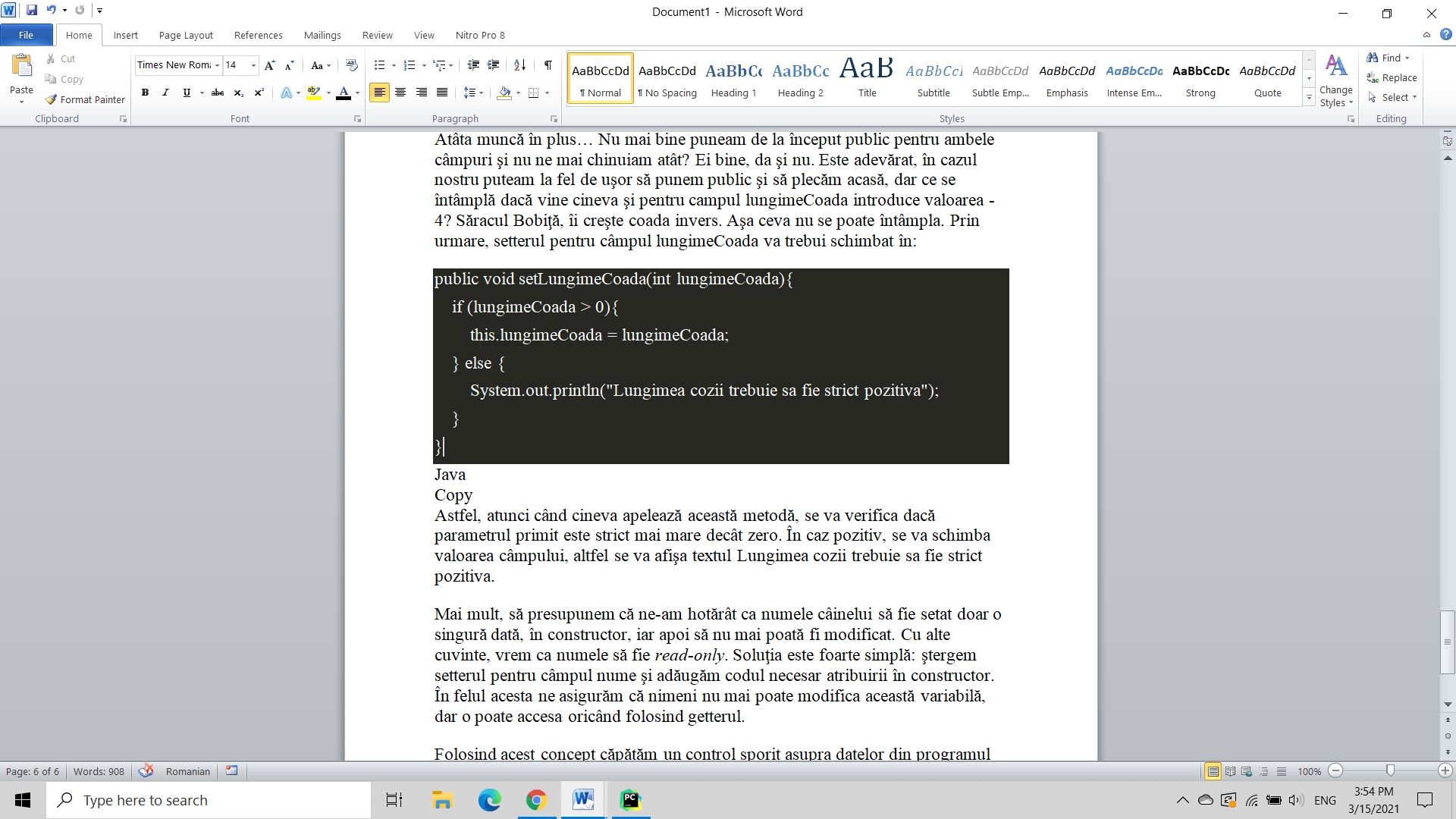
Astfel, dacă suntem într-o altă clasă şi vrem să modificăm numele şi lungimea cozii unui obiect de tip Caine, tot ce trebuie să facem este:



În urma rulării obţinem:



Atâta muncă în plus… Nu mai bine puneam de la început public pentru ambele câmpuri şi nu ne mai chinuiam atât? Ei bine, da şi nu. Este adevărat, în cazul nostru puteam la fel de uşor să punem public şi să plecăm acasă, dar ce se întâmplă dacă vine cineva şi pentru campul lungimeCoada introduce valoarea -4? Săracul Bobiţă, îi creşte coada invers. Aşa ceva nu se poate întâmpla. Prin urmare, setterul pentru câmpul lungimeCoada va trebui schimbat în:



Astfel, atunci când cineva apelează această metodă, se va verifica dacă parametrul primit este strict mai mare decât zero. În caz pozitiv, se va schimba valoarea câmpului, altfel se va afişa textul Lungimea cozii trebuie sa fie strict pozitiva.

2.Definirea unei clase cu 2 constructori,cu un destructor,cu 2 accesori,2 modificatori si inca 1-2 operatii.

Class manual

{

Public:

Char numemanual[40];

Char autor1[40];

char autor2[40];

int nrpagini;

double pret;

void citeste\_ manual (char \*numemanual,char \*autor1,char \*autor2,int \*nrpagini,double \* pret);

void afisare\_manual ();

};