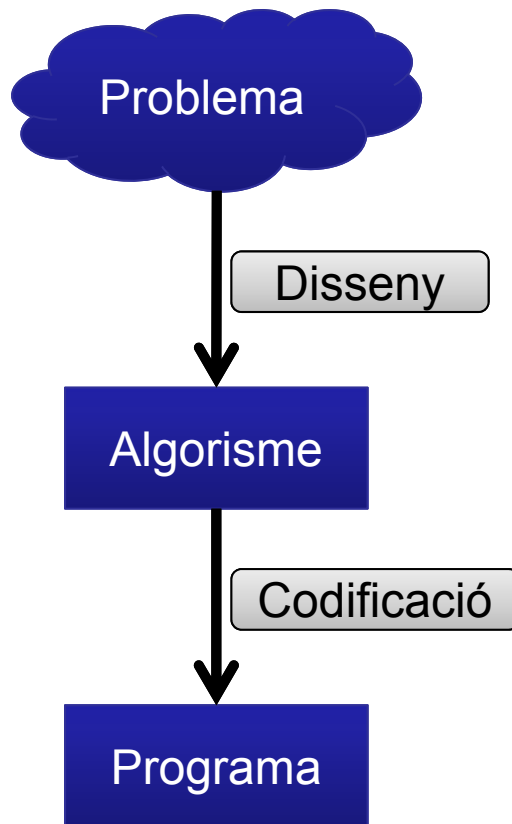


Pràctiques de Programació

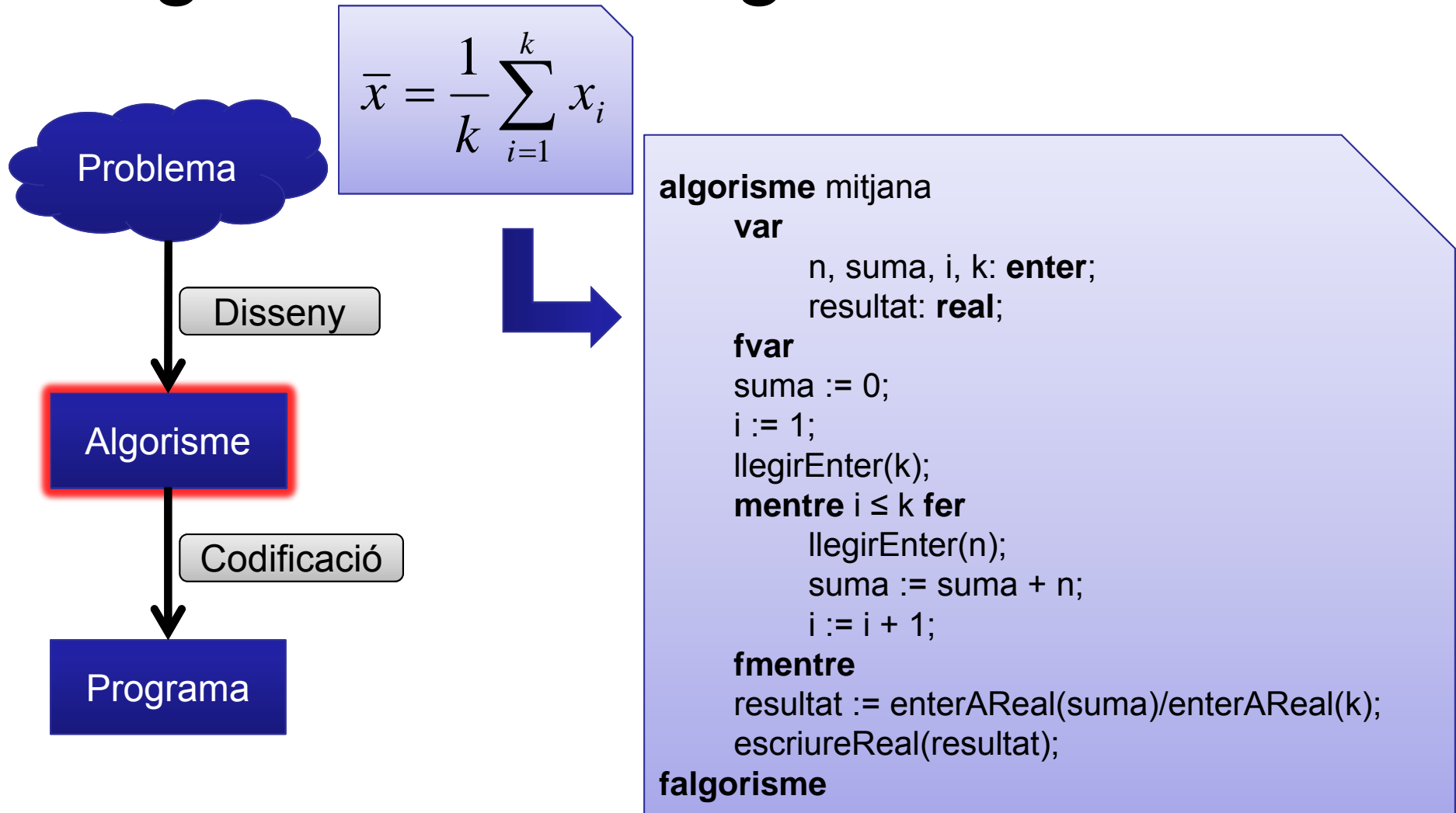
Algorismes i Programes

Algorismes i Programes

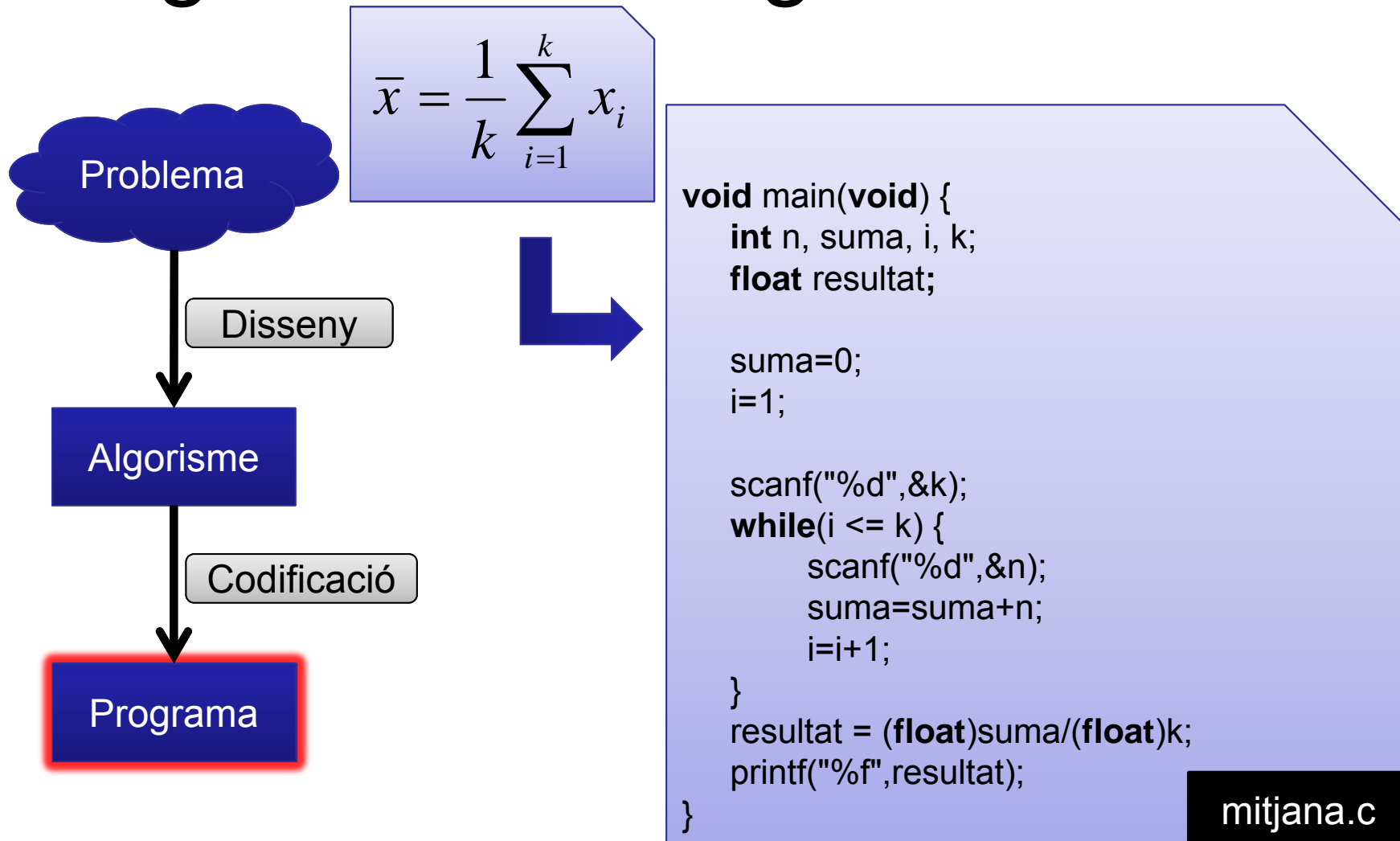


- **Algorisme:** Descripció “formal” dels passos a seguir per a resoldre un problema donat.
 - No ambigu i precís.
 - Llenguatge proper als humans.
- **Programa:** Traducció d’un algorisme a un llenguatge de programació.
 - No ambigu i precís.
 - Llenguatge proper a la màquina.
 - Podem utilitzar diferents llenguatges per un mateix algorisme (C, JAVA, PHP, ...).

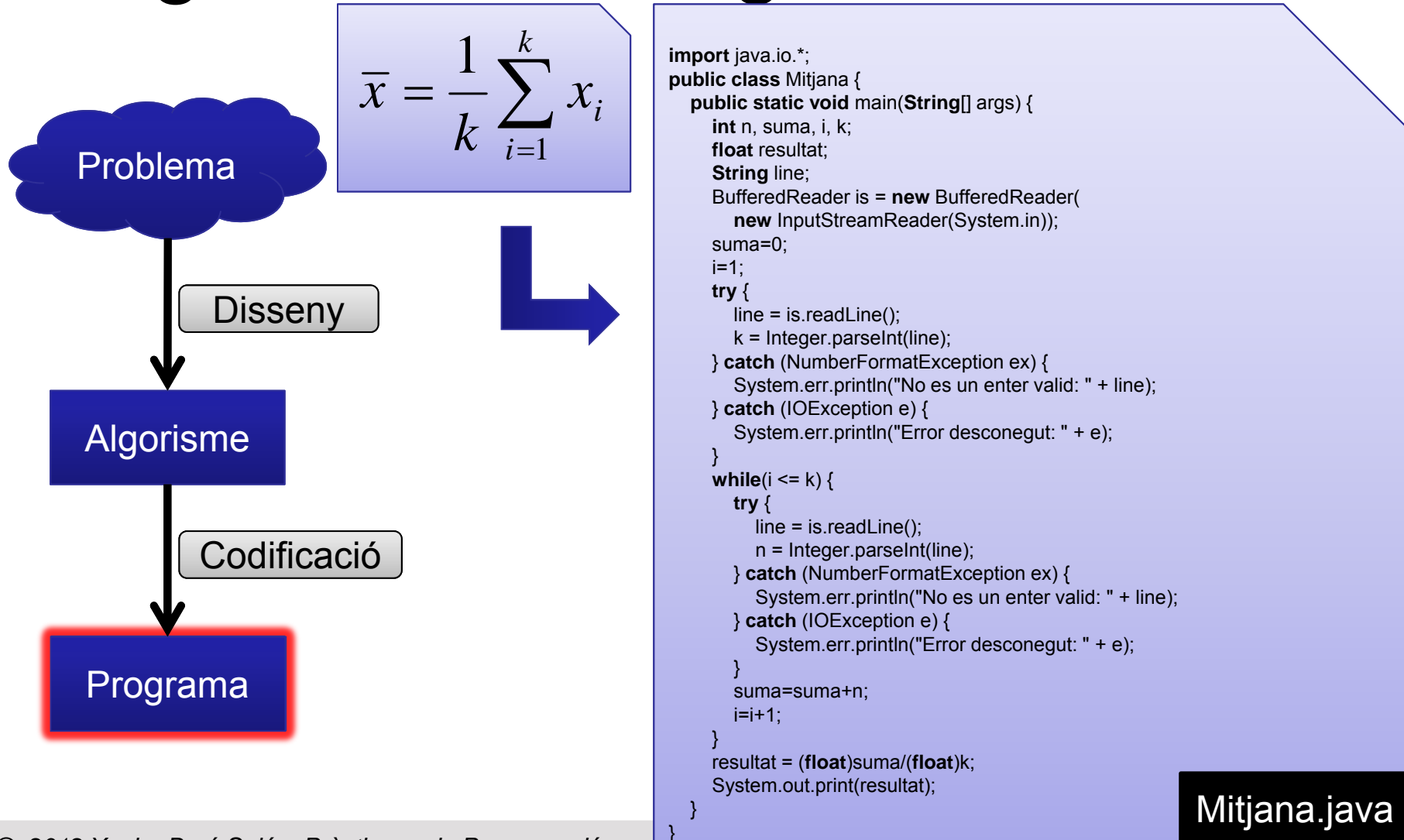
Algorismes i Programes



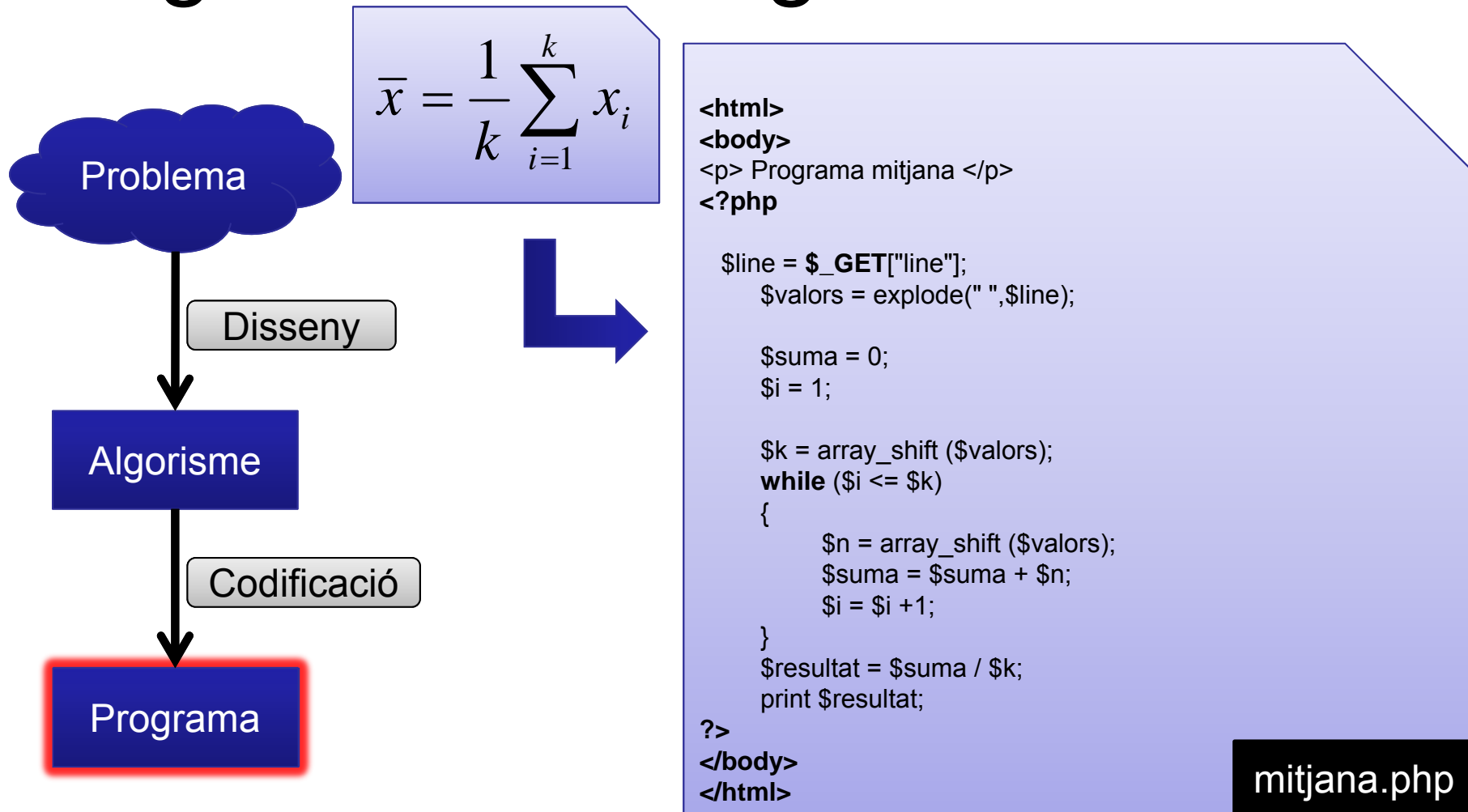
Algorismes i Programes



Algorismes i Programes



Algorismes i Programes



Algorismes i Programes

- **Comentaris finals:**

- **Els algorismes** us ajudaran a organitzar els passos per resoldre un determinat problema. Tot i que a les activitats teòriques es demana treballar amb algorismes, a les parts pràctiques, podeu treballar directament amb el llenguatge de programació, fent el disseny i la codificació en el mateix pas. De totes maneres, fins que no adquireu molta pràctica programant, us serà d'ajuda pensar en la estructura del llenguatge algorísmic per evitar que els vostres programes siguin caòtics i desorganitzats. En problemes complexos, també és aconsellable fer un bon disseny inicial.
- **Un bon programa** no és *només* aquell que funciona, o sigui, que es compila i s'executa sense errors, obtenint els resultats esperats. Un bon programa *també* ha de ser llegible, òptim, mantenible, estable, A continuació us donem alguns consells i bones pràctiques:
 - Comenteu bé el codi, amb comentaris que ajudin a entendre què s'està fent, i evitant comentaris obvis.
 - Utilitzeu noms de variables entenedors, que facilitin entendre per a què s'utilitza cada variable
 - Feu controls d'errors, comprovant que els valors dels paràmetres estan dins dels rangs que espereu, i desconfiant sempre de qualsevol entrada externa al vostre programa, especialment de la entrada per teclat d'informació per part d'un usuari. Si un pas del vostre programa pot fallar, verifiqueu que ha anat bé abans de continuar.
 - Intenteu pensar en el cost computacional de les vostres operacions, especialment dins de composicions iteratives. Cap al final de l'assignatura veurem maneres de calcular aquest cost de manera formal, però el sentit comú us ajudarà si pareu a penar-hi.
 - Feu un codi modular, amb funcions i accions per aquelles operacions que siguin generals, o que es repeteixen moltes vegades. Reutilitzar codi estalvia molt de temps i facilita la detecció d'errors.
 - Proveu el vostre codi, tant pel que fa a resultats finals, com a resultats parcials, provant les vostres funcions i accions en casos coneguts. Heu d'intentar buscar la pitjor situació, i comprovar que els vostres mecanismes de control d'errors funcionen.