



TECHNOLOGIE · 4E

PRIORITÉ P1

Programmer une nouvelle fonctionnalité d'un objet technique

variables, conditions, boucles et entrées-sorties

🎯 À LA FIN DE CETTE FICHE, TU SAURAS...

- ✓ Tu sauras utiliser une variable, une condition et une boucle pour programmer une nouvelle fonctionnalité d'un objet technique.
- ✓ Tu sauras expliquer le rôle d'un capteur et d'un actionneur dans un programme qui pilote un objet technique.
- ✓ Tu sauras calculer le nombre d'instructions exécutées par une boucle et la valeur finale d'une variable après plusieurs répétitions.

TON PARCOURS DE RÉVISION — coche chaque partie une fois révisée !

- | | |
|--|---|
| ☐ 1 · Les mots à connaître | ☐ 7 · La carte mentale |
| ☐ 2 · Les règles essentielles | ☐ 8 · Trouve l'erreur |
| ☐ 3 · Méthode | ☐ 9 · Exercices type Brevet |
| ☐ 4 · Exemple type Brevet | ☐ 10 · QCM chrono (5 min) |
| ☐ 5 · Les pièges à éviter | ☐ 11 · Ta carte « 30 secondes » |
| ☐ 6 · Tips pour t'en souvenir | ☐ ∞ Corrections (pour les parents) |

👤 LE COIN DES PARENTS — pas besoin de connaître le cours, tout est justifié.

1. Faites lire « Les mots à connaître » : variable, opérateur logique, condition, boucle, liste, événement, capteur, actionneur.
2. Demandez à votre enfant d'expliquer la différence entre un capteur et un actionneur sur un appareil programmé de la maison.
3. Corrigez le QCM avec les pages détachables. Moins de 8/10 : refaire la fiche dans 3 jours.

1

Les mots à connaître *surligne ceux que tu confonds !*

Instruction — une commande simple qu'exécute le programme, l'une après l'autre ; plusieurs instructions forment un bloc, par exemple le corps d'une boucle.

Programme — un ensemble ordonné d'instructions qui pilote un objet technique, en combinant variables, conditions, boucles et entrées-sorties pour obtenir le comportement souhaité.

Objet technique — un produit fabriqué par l'être humain pour répondre à un besoin (une trottinette, une lampe automatique, un objet domestique programmable...).

Variable — un espace nommé qui stocke une valeur (un nombre, un texte...) pouvant changer pendant l'exécution du programme.

Opérateur logique — un opérateur (ET, OU, NON) qui combine ou inverse des conditions vraies ou fausses pour construire un test plus précis.

Condition — une instruction qui teste si une expression est vraie ou fausse, et qui choisit entre deux chemins d'exécution différents selon le résultat.

Boucle — une instruction qui répète un bloc d'instructions un certain nombre de fois ou tant qu'une condition reste vraie.

Liste et événement — une liste est une structure qui stocke plusieurs valeurs sous un seul nom, chacune accessible par sa position. Un événement est un fait déclencheur (appui sur un bouton, dépassement d'un seuil mesuré par un capteur...) qui lance l'exécution d'une instruction ou d'un bloc d'instructions.

Entrée-sortie — l'échange d'informations entre le programme et l'environnement de l'objet technique : une entrée reçoit une donnée mesurée par un capteur, une sortie commande un effet produit par un actionneur.

Capteur et actionneur — un capteur mesure une grandeur de l'environnement de l'objet technique (distance, température, présence d'un obstacle...) et transmet cette valeur au programme (entrée). Un actionneur produit en retour un effet physique commandé par le programme (sortie), comme allumer une lampe ou faire tourner un moteur.

2

Les règles essentielles *chaque règle avec son exemple*

- 1 **Programmer une nouvelle fonctionnalité d'un objet technique consiste à combiner des variables, des conditions, des boucles et des entrées-sorties pour obtenir le comportement souhaité.**

exemple → un programme qui allume une lampe dans le noir combine une entrée (un capteur), une condition (si la valeur mesurée est faible) et une sortie (allumer la lampe).

- 2 **Une condition choisit entre deux chemins d'exécution différents selon qu'un test est vrai ou faux ; les opérateurs logiques ET, OU et NON permettent de combiner plusieurs tests.**

exemple → « si valeur mesurée faible ET mouvement détecté, alors allumer la lampe » combine deux tests avec l'opérateur ET : la lampe ne s'allume que si les deux conditions sont vraies en même temps.

- 3 **Une boucle répète un bloc d'instructions plusieurs fois ; le nombre total d'instructions exécutées se calcule en multipliant le nombre d'instructions du bloc par le nombre de tours de la boucle.**

exemple → une boucle de 3 instructions répétée 6 fois exécute au total $3 \times 6 = 18$ instructions.

- 4 **Un capteur transmet au programme une donnée mesurée dans l'environnement de l'objet technique (entrée) ; un actionneur produit l'effet physique commandé par le programme en retour (sortie).**

exemple → un capteur transmet une valeur mesurée au programme (entrée) ; une lampe commandée par le programme s'allume ou s'éteint (sortie, via un actionneur).

- 5 Une variable ne change de valeur que lorsqu'une instruction du programme le lui demande explicitement ; sa valeur finale après une boucle dépend du nombre de tours où cette instruction s'est exécutée.**

exemple → une variable compteur commence à 0. Elle augmente de 1 à chaque tour où une condition est vraie : si cette condition a été vraie 7 fois sur 10 tours, le compteur vaut $7 \times 1 = 7$ à la fin de la boucle.

- 6 Une liste permet de traiter plusieurs valeurs avec le même bloc d'instructions, en les parcourant une par une dans une boucle.**

exemple → une liste de 5 mesures de distance est parcourue par une boucle qui affiche chacune d'elles, sans écrire 5 fois la même instruction d'affichage.

- 7 Un événement déclenche l'exécution d'une instruction ou d'un bloc d'instructions au moment où il se produit, sans attendre le tour suivant d'une boucle.**

exemple → l'appui sur un bouton d'arrêt d'urgence déclenche immédiatement l'arrêt du moteur, dès que l'événement se produit, sans attendre la fin du tour de boucle en cours.

3

Méthode — Calculer le nombre d'instructions exécutées par une boucle

Exemple suivi : une boucle de 3 instructions répétée 6 fois.

Étape 1 RELÈVE le nombre d'instructions contenues dans le corps de la boucle.

Étape 2 RELÈVE le nombre de tours (répétitions) effectués par la boucle.

Étape 3 MULTIPLIE le nombre d'instructions par le nombre de tours : 3×6 .

Étape 4 OBTIENS le nombre total d'instructions exécutées : $3 \times 6 = 18$.

4

Exemple type Brevet — rédigé pas à pas

Un programme pilote un objet technique équipé d'un capteur et d'un actionneur (une lampe). Une boucle contient 4 instructions (lire le capteur, comparer sa valeur à un seuil, commander la lampe selon le résultat, attendre une seconde) et se répète 10 fois pendant un test. a) Calcule le nombre total d'instructions exécutées par cette boucle. b) Une variable compteur commence à 0 et augmente de 1 à chaque tour où la lampe s'allume ; elle s'allume à 7 des 10 tours. Quelle est la valeur finale du compteur ?

Étape 1 — Relève le nombre d'instructions du corps de la boucle : 4, et le nombre de tours : 10.

Étape 2 — Multiplie le nombre d'instructions par le nombre de tours : 4×10 .

Étape 3 — Relève le nombre de tours où la condition (lampe allumée) est vraie : 7 sur les 10 tours.

Étape 4 — La variable compteur augmente de 1 à chacun de ces 7 tours, en partant de 0 : $0 + 7 \times 1$.

Conclusion : a) $4 \times 10 = 40$: la boucle exécute 40 instructions au total. b) $7 \times 1 = 7$ (en partant de 0) : le compteur vaut 7 à la fin du test, car il augmente uniquement lors des 7 tours où la lampe s'allume.

5 Les pièges à éviter *les erreurs classiques en contrôle...*

X Additionner le nombre d'instructions du corps de la boucle et son nombre de tours au lieu de les multiplier.

X Confondre une condition, qui choisit entre deux chemins selon un test vrai ou faux, et une boucle, qui répète des instructions.

X Croire qu'un capteur agit directement sur l'objet technique, alors que c'est l'actionneur qui produit l'effet ; le capteur ne fait que mesurer une grandeur.

X Oublier qu'une variable ne change de valeur que lorsqu'une instruction le lui demande explicitement : elle ne se met jamais à jour toute seule.

6 Tips — pour t'en souvenir

Instructions exécutées par une boucle = instructions du bloc répété $\times$ nombre de tours (ex. $2 \times 9 = 18$, ou $3 \times 6 = 18$).

Condition = choix entre deux chemins ; boucle = répétition d'instructions. Ne confonds pas les deux.

Une variable ne change que si une instruction le lui demande : elle ne se met jamais à jour toute seule.

7 La carte mentale *dessiner, c'est mémoriser !*

Carte mentale à dessiner : « PROGRAMMER UNE FONCTIONNALITÉ » au centre, 4 branches colorées. VARIABLE : stocke une valeur qui change. CONDITION : ET/OU/NON, choix vrai/faux. BOUCLE : répète des instructions (instructions $\times$ tours). ENTRÉE-SORTIE : capteur (mesure) / actionneur (effet).



ta carte mentale, de mémoire...

8 À toi de jouer — trouve l'erreur

Une boucle contient 5 instructions et se répète 4 fois. Un élève écrit : « $5+4=9$, la boucle exécute 9 instructions. »

→ *cherche l'erreur avant de lire la réponse !*

L'erreur : L'élève additionne le nombre d'instructions et le nombre de tours au lieu de les multiplier. Le nombre total d'instructions exécutées se calcule en multipliant le nombre d'instructions du corps de la boucle par son nombre de tours : $5 \times 4 = 20$. La boucle exécute donc 20 instructions au total, et non 9 comme l'écrit l'élève.

9 Exercices type Brevet *corrections en fin de fiche ✂*

Exercice 1. Une boucle contient 3 instructions et se répète 7 fois. Calcule le nombre total d'instructions exécutées.

Exercice 2. Explique la différence entre un capteur et un actionneur dans un objet technique programmé, avec un exemple pour chacun.

10 QCM — 10 questions 5 MIN MAX • OBJECTIF 8/10

Une seule bonne réponse par question, sans calculatrice — entraînement épreuve d'automatismes du Brevet 2026.

1. Une variable, dans un programme, est :

- ☐ A un espace nommé qui stocke une valeur pouvant changer pendant l'exécution du programme
- ☐ B une instruction qui s'exécute une seule fois
- ☐ C le nom donné au capteur d'un objet technique

2. Les opérateurs logiques ET, OU et NON servent à :

- ☐ A combiner ou inverser des conditions vraies ou fausses pour construire un test
- ☐ B additionner des valeurs numériques entre elles
- ☐ C nommer les variables d'un programme

3. Une condition, dans un programme, permet de :

- ☐ A répéter un bloc d'instructions un nombre fixe de fois
- ☐ B choisir entre deux chemins d'exécution différents selon qu'un test est vrai ou faux
- ☐ C mesurer une grandeur physique de l'environnement

4. Une boucle contient 4 instructions et se répète 10 fois. Le nombre total d'instructions exécutées est :

- ☐ A 14
- ☐ B 40
- ☐ C 6

5. Une boucle contient 3 instructions et se répète 7 fois. Le nombre total d'instructions exécutées est :

- ☐ A 10
☐ B 4
☐ C 21

6. Une liste, dans un programme, permet de :

- ☐ A stocker plusieurs valeurs sous un seul nom, accessibles chacune par leur position
☐ B exécuter une seule instruction plusieurs fois sans jamais la répéter
☐ C mesurer directement une grandeur physique

7. Un événement, dans un programme qui pilote un objet technique, est :

- ☐ A une valeur numérique stockée dans une variable
☐ B un fait déclencheur qui lance l'exécution d'une instruction
☐ C le nombre de tours effectués par une boucle

8. Un capteur et un actionneur ont pour rôle respectif de :

- ☐ A mesurer une grandeur de l'environnement pour le capteur, produire un effet physique commandé par le programme pour l'actionneur
☐ B stocker une valeur pour le capteur, répéter des instructions pour l'actionneur
☐ C exécuter le programme entier pour le capteur, l'arrêter pour l'actionneur

9. Une variable compteur commence à 0. Elle augmente de 1 à chaque tour où une condition est vraie, ce qui arrive à 7 des 10 tours d'une boucle. Sa valeur finale est :

- ☐ A 10
☐ B 3
☐ C 7

10. Programmer une nouvelle fonctionnalité d'un objet technique consiste à :

- ☐ A utiliser exclusivement des boucles, sans jamais de variable ni de condition
☐ B combiner des variables, des conditions, des boucles et des entrées-sorties pour obtenir le comportement souhaité
☐ C écrire uniquement le programme principal, sans jamais utiliser de capteur



11

En 30 secondes — ta carte de dernière révision

découpe-la et garde-la dans ta trousse — révision de veille de contrôle !

- Programmer une fonctionnalité combine variables, opérateurs logiques (ET, OU, NON), conditions, boucles et entrées-sorties (capteurs, actionneurs).
- Capteur = mesure une grandeur (entrée). Actionneur = produit un effet commandé (sortie).
- Instructions exécutées par une boucle = instructions du bloc $\times$ nombre de tours (ex. $4 \times 10 = 40$).

PLAN DE RÉVISION ESPACÉE — relis la carte et refais le QCM : ☐ J+1 ☐ J+3 ☐ J+7 ☐ J+14



QCM en ligne

VA PLUS LOIN — TESTE-TOI SUR CETTE NOTION

Scanne le QR code (ou clique dessus) : le QCM chronométré de cette notion, note immédiate, chaque erreur expliquée. Gratuit, sans inscription.

Pour t'entraîner — QCM en nombre, évaluations, explications détaillées, analyse de tes points faibles — l'abonnement Réussito (sans engagement) : via le QR code ou sur reussito.fr/abo

Débloque toutes les fiches de 4e : via le QR code ou sur reussito.fr/4e



L'élève travaille sans les solutions sous les yeux ; vous corrigez ensemble avec ces pages. Chaque réponse est justifiée : pas besoin de connaître le cours.

Correction des exercices

Exercice 1. Multiplie le nombre d'instructions du corps de la boucle par son nombre de tours : $3 \times 7 = 21$. La boucle exécute 21 instructions au total.

Exercice 2. Un capteur mesure une grandeur de l'environnement de l'objet technique, par exemple la distance à un obstacle. Un actionneur produit un effet physique commandé par le programme, par exemple une lampe qui s'allume ou un moteur qui tourne.

Réponses du QCM — expliquées une par une

- 1 → A** Une variable est un espace nommé qui stocke une valeur pouvant changer pendant l'exécution du programme. *Piège : B décrit une instruction ordinaire, pas une variable. C confond variable et capteur, deux notions différentes.*
- 2 → A** Les opérateurs logiques ET, OU et NON combinent ou inversent des conditions vraies ou fausses pour construire un test plus précis. *Piège : B décrit une opération de calcul (une addition), pas un opérateur logique. C n'a aucun rapport avec les opérateurs logiques.*
- 3 → B** Une condition teste si une expression est vraie ou fausse et choisit entre deux chemins d'exécution différents selon le résultat. *Piège : A décrit une boucle, pas une condition. C décrit le rôle d'un capteur.*
- 4 → B** 4×10 fait 40 : on multiplie le nombre d'instructions du corps de la boucle par son nombre de tours. *Piège : 14 additionne à tort 4 et 10 au lieu de les multiplier. 6 soustrait à tort 4 de 10.*
- 5 → C** 3×7 fait 21 : on multiplie le nombre d'instructions du corps de la boucle par son nombre de tours. *Piège : 10 additionne à tort 3 et 7 au lieu de les multiplier. 4 soustrait à tort 3 de 7.*
- 6 → A** Une liste stocke plusieurs valeurs sous un seul nom, chacune accessible par sa position. *Piège : B décrit vaguement une boucle sans en donner le mécanisme correct. C décrit le rôle d'un capteur.*
- 7 → B** Un événement est un fait déclencheur (appui sur un bouton, dépassement d'un seuil...) qui lance l'exécution d'une instruction ou d'un bloc d'instructions. *Piège : A confond événement et variable. C confond événement et nombre de répétitions d'une boucle.*
- 8 → A** Un capteur mesure une grandeur de l'environnement de l'objet technique ; un actionneur produit un effet physique commandé par le programme. *Piège : B attribue à tort au capteur et à l'actionneur les rôles d'une variable et d'une boucle. C n'a aucun rapport avec les fonctions réelles du capteur et de l'actionneur.*
- 9 → C** $0 + 7 \times 1$ fait 7 : le compteur augmente uniquement lors des 7 tours où la condition est vraie. *Piège : 10 confond le nombre total de tours de la boucle avec le nombre de fois où la condition est vraie. 3 correspond au nombre de tours où la condition est fausse ($10 - 7$), pas au nombre de fois où le compteur augmente.*
- 10 → B** Programmer une nouvelle fonctionnalité consiste à combiner des variables, des conditions, des boucles et des entrées-sorties pour obtenir le comportement souhaité. *Piège : A ignore qu'un programme combine plusieurs briques différentes, pas seulement des boucles. C ignore que les capteurs (entrées) sont indispensables à un programme qui doit réagir à son environnement.*