

RCP : Référentiel de compétences en programmation - vue détaillée

[Voir la vue générale](#) [Voir la vue formulaire de sélection](#) [Voir la documentation](#)

Programmer avec des expressions (01)

- ▶ Évaluer la valeur et le type (entier, chaîne de caractères ou booléen) d'une expression comportant variables, constantes et opérateurs (E01)
- ▶ Modéliser les informations disponibles/à calculer par des variables de type élémentaire (entier, chaîne de caractères et booléen) (M01)
- ▶ Anticiper l'écriture d'une expression comportant variables, constantes et opérateurs (A01)
- ▶ Décomposer une expression complexe en identifiant une ou des variables intermédiaires (D01)
- ▶ Généraliser une expression en remplaçant une constante par une variable (G01)

Programmer avec des affectations et des séquences (02)

- ▶ Évaluer les effets de l'exécution d'un programme comportant une séquence d'affectations (E02)
- ▶ Modéliser les informations à calculer par une ou des variables intermédiaires (M02)
- ▶ Anticiper l'écriture d'un traitement séquentiel pour obtenir un résultat spécifié, la méthode étant donnée (A02)
- ▶ Décomposer un traitement complexe en identifiant une ou des variables intermédiaires (D02)

Programmer avec des boucles bornées (03)

- ▶ Évaluer les effets de l'exécution d'un programme comportant une boucle bornée (E03)
- ▶ Modéliser par une ou des variables les informations à modifier à chaque répétition (M03)
- ▶ Anticiper l'écriture d'une boucle bornée pour obtenir un résultat spécifié, la méthode étant donnée (A03)

- ▶ Décomposer un traitement complexe en identifiant une boucle bornée (D03)
- ▶ Généraliser un traitement séquentiel en le remplaçant par un traitement répétitif (G03)

Programmer avec des boucles non bornées (04)

- ▶ Évaluer les effets de l'exécution d'un programme comportant une boucle non bornée (E04)
- ▶ Modéliser par une ou des variables les informations à modifier à chaque répétition (M04)
- ▶ Anticiper l'écriture d'un traitement répétitif pour obtenir un résultat spécifié, la méthode étant donnée (A04)
- ▶ Décomposer un traitement complexe en identifiant une boucle non bornée (D04)

Programmer avec des constructions alternatives ou conditionnelles (05)

- ▶ Évaluer les effets de l'exécution d'un programme comportant une ou des constructions alternatives ou conditionnelles (E05)
- ▶ Modéliser par une variable booléenne la condition d'une alternative ou d'une conditionnelle (M05)
- ▶ Anticiper l'écriture d'un traitement alternatif ou conditionnel pour obtenir un résultat spécifié, la méthode étant donnée (A05)
- ▶ Décomposer un traitement complexe en plusieurs cas en utilisant des constructions alternatives ou conditionnelles (D05)

Programmer avec des fonctions (06)

- ▶ Évaluer le résultat de l'appel d'une fonction avec des paramètres donnés (E06)
- ▶ Modéliser un traitement par une fonction en spécifiant ses paramètres (M06)
- ▶ Anticiper l'écriture d'une fonction, les paramètres étant spécifiés (A06)
- ▶ Décomposer un traitement complexe en utilisant la composition de fonctions (D06)
- ▶ Généraliser un traitement, par la définition d'une fonction avec paramètres ou par l'ajout d'un paramètre à une fonction (G06)
- ▶ Abstraire un traitement par une définition de fonction avec paramètres et commentaire (X06)

Programmer avec des fonctions récursives (07)

- ▶ Évaluer le résultat de l'appel d'une fonction récursive (E07)
- ▶ Modéliser un traitement par une fonction récursive en spécifiant ses paramètres (M07)
- ▶ Anticiper l'écriture d'une fonction récursive, les paramètres étant spécifiés (A07)
- ▶ Généraliser une fonction récursive avec une fonction en paramètre (G07)

Programmer avec des tuples (08)

- ▶ Évaluer une expression permettant d'accéder aux informations d'un tuple (E08)
- ▶ Modéliser une information complexe par un tuple d'informations élémentaires (M08)
- ▶ Anticiper l'accès aux informations élémentaires d'un tuple (A08)
- ▶ Abstraire des données complexes en utilisant des tuples (X08)

Programmer avec des enregistrements à champs nommés (09)

- ▶ Évaluer une expression permettant d'accéder aux informations d'un enregistrement à champs nommés (E09)
- ▶ Modéliser une information complexe par un enregistrement à champs nommés (M09)
- ▶ Anticiper l'accès aux informations élémentaires d'un enregistrement à champs nommés (A09)
- ▶ Abstraire des données complexes en utilisant des enregistrements à champs nommés (X09)

Programmer avec des tableaux - construction et accès (10)

- ▶ Évaluer une expression ou les effets de l'exécution d'un programme utilisant un tableau (E10)
- ▶ Modéliser une série d'informations par un tableau donné en extension (M10)
- ▶ Anticiper les traitements à programmer pour accéder aux éléments d'un tableau par leurs indices et les modifier (A10)
- ▶ Généraliser l'écriture d'un tableau en utilisant la compréhension (G10)
- ▶ Abstraire les données complexes en utilisant des tableaux (X10)

Programmer des parcours de tableau (11)

- ▶ Évaluer les effets de l'exécution d'un programme itérant sur un tableau (E11)
- ▶ Modéliser un traitement de tableau en choisissant d'itérer sur les indices ou les éléments (M11)
- ▶ Anticiper un parcours linéaire de tableau (A11)
- ▶ Généraliser un traitement en l'appliquant à tous les éléments d'un tableau (G11)

Programmer avec des dictionnaires - construction et accès (12)

- ▶ Évaluer une expression ou les effets de l'exécution d'un programme utilisant un dictionnaire (E12)
- ▶ Modéliser l'association d'informations par un dictionnaire (M12)
- ▶ Anticiper le traitement permettant d'ajouter ou modifier une association à un dictionnaire (A12)
- ▶ Abstraire les données complexes en utilisant des dictionnaires (X12)

Programmer avec des dictionnaires - itérer (13)

- ▶ Évaluer les effets de l'exécution d'un programme itérant sur un dictionnaire (E13)
- ▶ Modéliser un traitement de dictionnaire en choisissant d'itérer sur les clés, les valeurs ou les paires (M13)
- ▶ Anticiper un traitement nécessitant d'itérer sur un dictionnaire (A13)

Programmer avec des interfaces et des implémentations (14)

- ▶ Modéliser les données et les traitements en spécifiant une structure de données par son interface (M14)
- ▶ Anticiper l'implémentation d'une structure de données (A14)
- ▶ Abstraire un traitement en utilisant une structure de données à spécifier. (X14)

Programmer avec des objets et des classes (15)

- ▶ Évaluer la création d'une instance, l'invocation de méthodes et l'accès aux attributs d'un objet (E15)
- ▶ Modéliser les données et les traitements en spécifiant une classe, des attributs et des méthodes (M15)
- ▶ Anticiper des traitements en implémentant les méthodes d'une classe (A15)
- ▶ Décomposer les données et les traitements en les regroupant dans des classes (D15)
- ▶ Généraliser une classe par l'ajout d'un attribut ou d'une méthode (G15)
- ▶ Abstraire un ensemble de données et de traitements en les encapsulant dans une classe en tant qu'attributs et méthodes (X15)

Sophie Chane-Lune, Sébastien Hoarau, Christophe Declercq, Laboratoire d'Informatique et de Mathématiques, Université de La Réunion, 2023-2025