



Didapro 10 - DidaSTIC

Didactique de l'informatique et des STIC

30 janv.-1 févr. 2024 Louvain-la-Neuve (Belgique)

Vers l'analyse de ressources d'apprentissage de la programmation à l'aide d'un référentiel de types de tâches

Sébastien Jolivet, Eva Dechaux,
Anne-Claire Gobard, Patrick Wang

Décrire des ressources : quels défis ?

Indexation (Loiseau, 2009) : « opération qui consiste à décrire et à caractériser un document à l'aide de représentations des concepts contenus dans ce document, c'est-à-dire à **transcrire en langage documentaire les concepts après les avoir extraits du document par une analyse** ».

Langage documentaire (Lefevre, 2000) : « (...) langage artificiel qui fournit une **représentation formalisée et univoque** des documents d'un corpus et des questions qui **intéressent un groupe d'utilisateurs**, (...) ».

Pour résumer : décrire une ressource, pour qui ? Pour quoi ? Comment ?

Décrire des ressources : nos choix - objectifs

Groupe d'utilisateurs : enseignants ou apprenants d'informatique / EAIH d'apprentissage de l'informatique

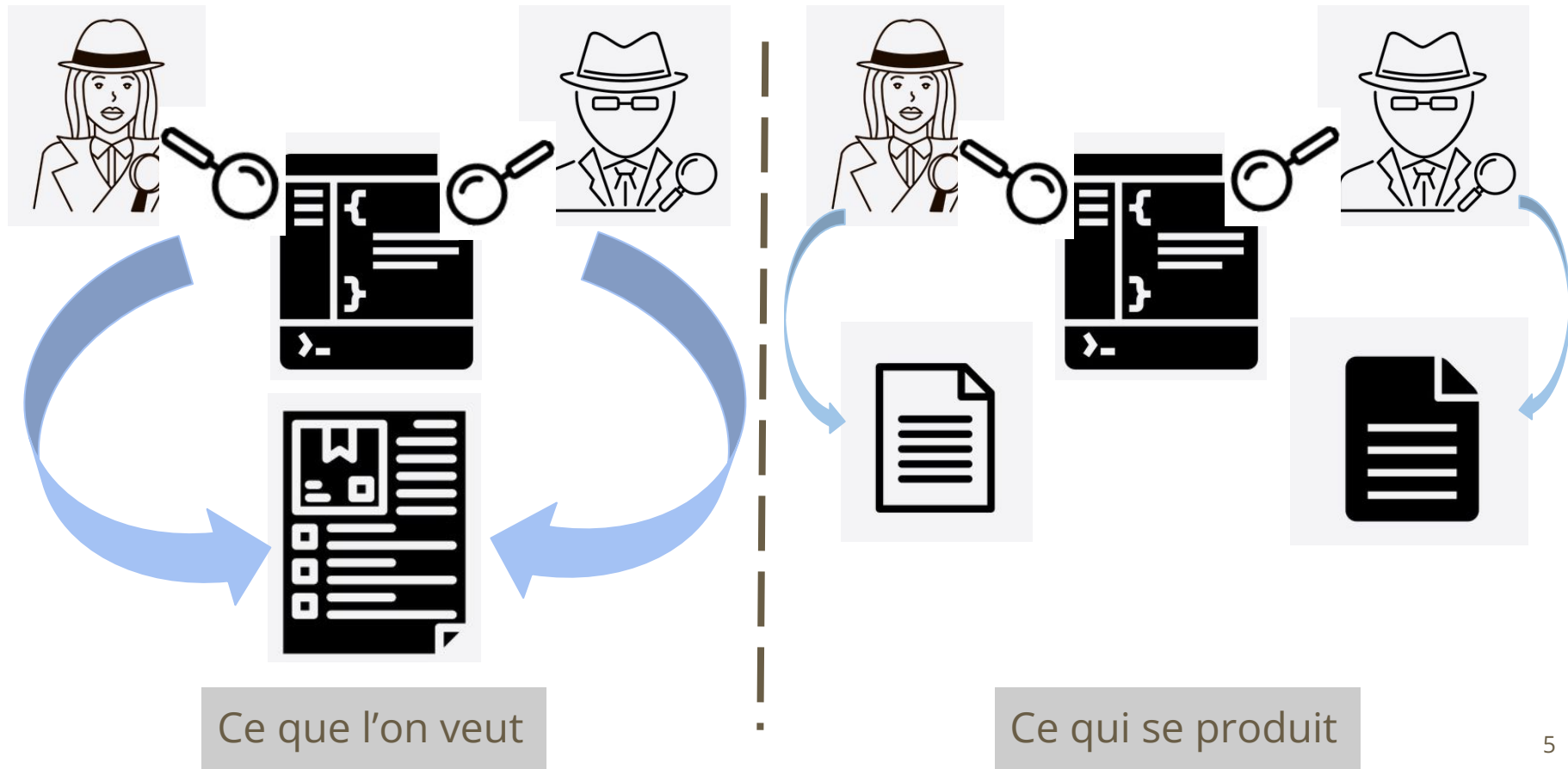
Finalité de la description : **outiller** les enseignants ou un EIAH pour le **choix de ressources**

Critère de description : **caractéristiques didactiques** de la ressource

Introduction : une ou des descriptions ?



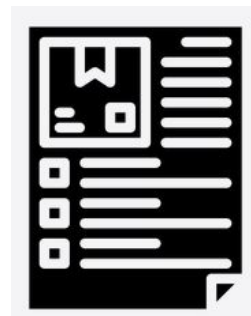
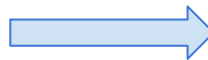
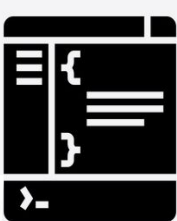
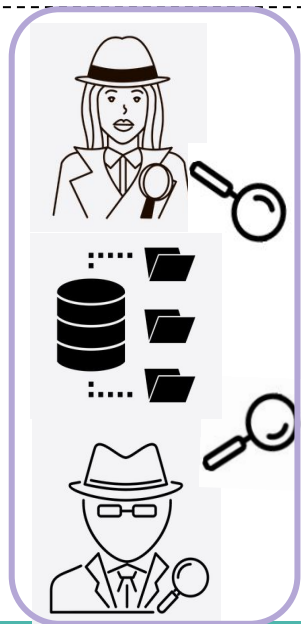
Introduction : une ou des descriptions ?



Réponse (très) partielle

Moyen pour répondre aux enjeux : **réaliser (une partie de) la description relativement à un référentiel** décrivant savoirs et savoir-faire relatifs à l'activité de programmation.

Référentiel des
savoirs et
savoir-faire en
programmation

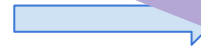
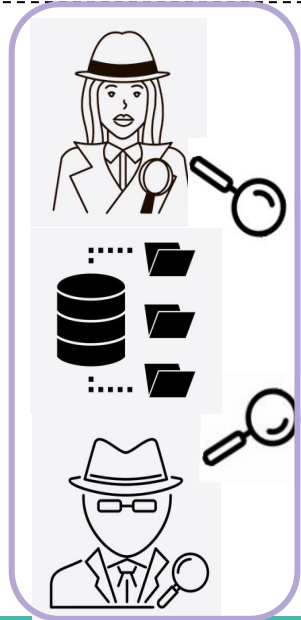


Ressource
décrite, de
manière
unique, selon
des
caractéristiques
didactiques

Réponse (très) partielle

Moyen pour répondre aux enjeux : **réaliser (une partie de) la description relativement à un référentiel** décrivant savoirs et savoir-faire relatifs à l'activité de programmation.

Référentiel des
savoirs et
savoir-faire en
programmation



Ressource
décrite, de
manière
unique, selon

techniques
didactiques

Encore faut-il disposer d'un référentiel...

Programmer : quel référentiel ?

Qu'est-ce que programmer ?

Programming subtasks	Basic processes	Knowledge domains	Mental representations	External representations
Understanding the problem	COMPOSITION	Domain knowledge (e.g., statistics, banking)	Situation model	Requirements document Specifications document
Design		Design strategies Programming Algorithms and methods Design language	Solution model Plan representation	Design document
Coding		Programming language Programming conventions	Program representation	Code
Maintenance		All knowledge domains Debugging, testing strategies Frequent kinds of error	All representations	All documents

Figure 1: The tasks of programming.

Pennington & Grabowski, 1990

Les auteures signalent que ce découpage ne doit pas masquer la forte intrication des tâches des différentes catégories lors de l'activité de programmation

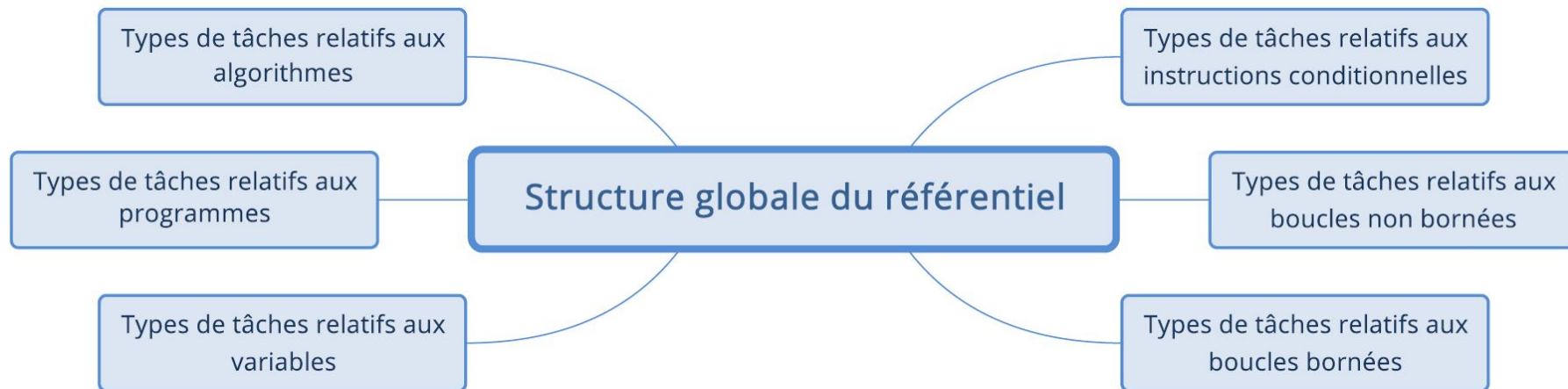
Qu'est-ce que programmer ?

Programming subtasks	Basic processes	Knowledge domains	Mental representations	External representations
Understanding the problem	C	Domain knowledge (e.g., statistics, banking)	Situation model	Requirements document Specifications document
Design	M P O S I T I O N	Design strategies Programming Algorithms and methods Design language	Solution model Plan representation	Design document
Coding	E N S I O N	Programming language Programming conventions	Program representation	Code
Maintenance	N	All knowledge domains Debugging, testing strategies Frequent kinds of error	All representations	All documents

Figure 1: The tasks of programming.

Pennington & Grabowski, 1990

Clés de structuration du référentiel

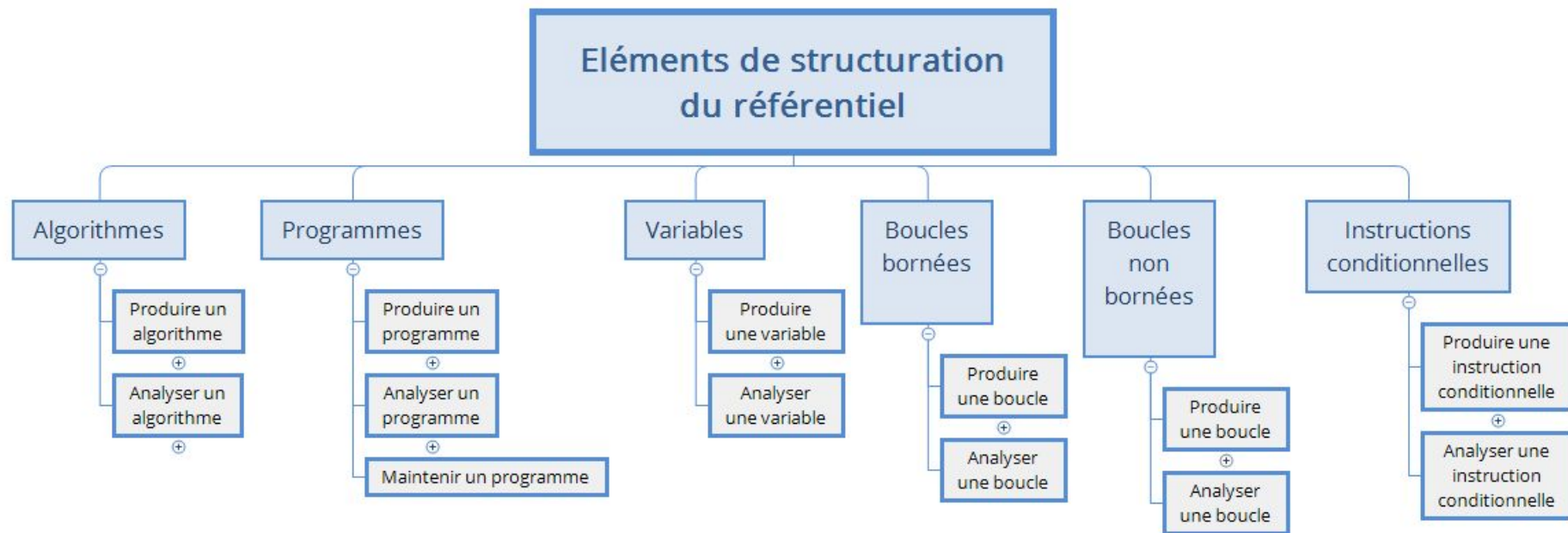


Accès au référentiel : https://link.infini.fr/ref_prog

Le référentiel est placé sous licence

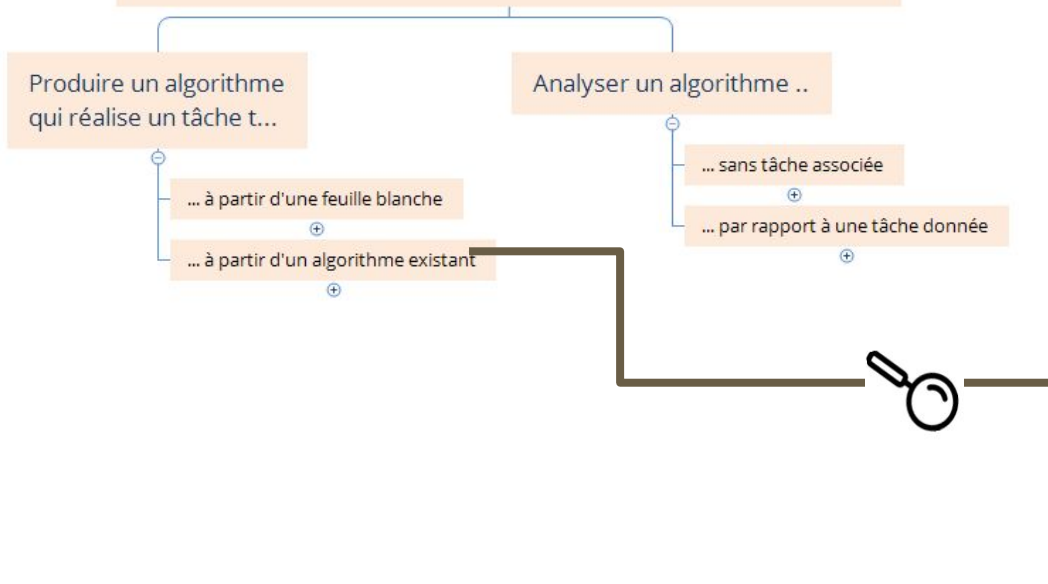


Clés de structuration du référentiel

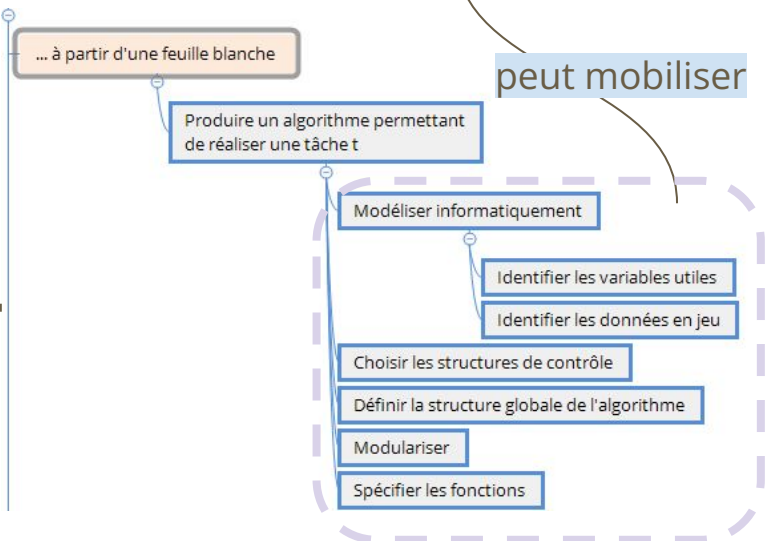


Zoom sur les algorithmes

Types de tâches relatifs aux algorithmes



Produire un algorithme permettant de réaliser une tâche t



Conclusions sur le référentiel, apports et suites

(Re)mettre en évidence la complexité, pour des novices, de types de tâches parfois perçues comme “évidentes”

Lier savoir-faire et savoirs : chaque type de tâches identifié peut-être relié à différents savoirs

Granularité et couverture : les niveaux de granularité et de couverture sont nécessairement relatif aux besoins auxquels le référentiel veut répondre (niveau scolaire, utilisations...). Mais l'étendre et/ou le raffiner ne remet pas en cause l'existant

Types de tâches didactiques : certains types de tâches sont considérés comme didactiques dans la mesure où il n'ont pas d'existence en dehors d'une institution d'enseignement

Retour sur la description de ressources

Mise en œuvre possible – Description d'un exercice

Objectif de cette partie : illustrer la capacité du référentiel à contribuer à une **description didactique de ressources de type exercice**.

Un exercice est caractérisé par un **énoncé** qui permet de décrire un **problème** à résoudre et de communiquer ce problème.

Approche utilisée :

- Identification d'un exercice *souvent* demandé aux élèves
- Construction d'une solution typique
- Identification des types de tâches

Exemple d'exercice – Le juste prix

Écrire un programme qui permet à l'utilisateur de deviner un nombre entier entre 1 et 100, choisi aléatoirement par l'ordinateur. Il fait des propositions et indique si le nombre est plus grand, plus petit ou affiche gagné si le nombre est trouvé. La partie s'arrête dès que c'est gagné.

Exemple d'exercice – Le juste prix

Écrire un programme qui permet à l'utilisateur de deviner un nombre entier entre 1 et 100, choisi aléatoirement par l'ordinateur. Il fait des propositions et indique si le nombre est plus grand, plus petit ou affiche gagné si le nombre est trouvé. La partie s'arrête dès que c'est gagné.

```
import random
devine = 0
prix = random.randint(1,100)
while prix != devine:
    devine = int(input("Devinez le prix : "))
    if devine > prix :
        print("C'est moins !")
    elif devine < prix :
        print("C'est plus !")
    else:
        print("Bravo, le prix est bien", devine)
```

Exemple d'exercice – Le juste prix

- Types de tâches Algo/Prog:
- Concevoir un algo/prog réalisant une tâche t
 - Exécuter un programme
 - Tester un programme

```
import random
devine = 0
prix = random.randint(1,100)
while prix != devine:
    devine = int(input("Devinez le prix : "))
if devine > prix :
    print("C'est moins !")
elif devine < prix :
    print("C'est plus !")
else:
    print("Bravo, le prix est bien", devine)
```

Exemple d'exercice – Le juste prix

```
import random
devine = 0
prix = random.randint(1,100)
while prix != devine:
    devine = int(input("Devinez le prix : "))
if devine > prix :
    print("C'est moins !")
elif devine < prix :
    print("C'est plus !")
else:
    print("Bravo, le prix est bien", devine)
```

Types de tâches Algo/Prog:

- Concevoir un algo/prog réalisant une tâche t
- Exécuter un programme
- Tester un programme

Types de tâches sur les concepts:

- Déclarer une **variable**
- Affecter une valeur à une variable
- Utiliser une variable dans une **expression**
- Concevoir une **structure conditionnelle**...
- Concevoir une boucle non-bornée...

Exploitation pour la description d'EIAH

TABLE 1. Effectif des types de tâches de la catégorie *Programme* par EIAH.

Types de tâches		A.P.	Citi	Pyra
Analyser un code existant		0	0	0
Feuille blanche	Copier et exécuter	0	6	0
	Concevoir un programme	26	43	8
Code existant : Modifier un programme P en P' qui réalise une tâche t	P est erroné	1	0	0
	P réalise t' proche de t	15	0	0
	P contient des trous	3	0	0
Nombre d'exercices décrits		45	49	8

Identifier et
comparer le
présent

Exploitation pour la description d'EIAH

TABLE 1. Effectif des types de tâches de la catégorie *Programme* par EIAH.

Types de tâches		A.P.	Citi	Pyra
Analyser un code existant		0	0	0
Feuille blanche	Copier et exécuter	0	6	0
	Concevoir un programme	26	43	8
Code existant : Modifier un programme P en P' qui réalise une tâche t	P est erroné	1	0	0
	P réalise t' proche de t	15	0	0
	P contient des trous	3	0	0
Nombre d'exercices décrits		45	49	8

Mais aussi
détecter
l'absent

Identifier et
comparer le
présent

Bilan de l'analyse de ressources

Critère de description : **caractéristiques didactiques** de la ressource

Bilan de l'analyse de ressources

Critère de description : **caractéristiques didactiques** de la ressource

Description (partielle) relativement à un référentiel décrivant savoirs et savoir-faire relatifs à l'activité de programmation, mais ...

- Ensemble non-ordonné de types de tâches
- Pas de prise en compte de multiples solutions
- Pas ou peu de prise en compte de l'énoncé
- Fait émerger quelques prérequis, mais pas forcément tous

Bilan de l'analyse de ressources

```
import random
devine = 0
prix = random.randint(1,100)
while prix != devine:
    devine = int(input("Devinez le prix : "))
if devine > prix :
    print("C'est moins !")
elif devine < prix :
    print("C'est plus !")
else:
    print("Bravo, le prix est bien", devine)
```

- Ensemble non-ordonné de types de tâches
- Pas de prise en compte de multiples solutions
- Pas ou peu de prise en compte de l'énoncé
- Fait émerger quelques prérequis, mais pas forcément tous

Types de tâches sur les concepts:

- Déclarer une **variable**
- Affecter une valeur à une variable
- Utiliser une variable dans une **expression**
- Concevoir une **structure conditionnelle**...
- Concevoir une boucle non-bornée...

Conclusion générale

Quelques perspectives et enjeux pour conclure

Point de vue de l'**utilisateur humain** :

- Lier des ressources avec des curriculums
- Organiser son enseignement
- Faire de l'(auto)évaluation
- Identifier les erreurs possibles

Quelques perspectives et enjeux pour conclure

Point de vue de l'**utilisateur humain** :

- Lier des ressources avec des curriculums
- Organiser son enseignement
- Faire de l'(auto)évaluation
- Identifier les erreurs possibles

Point de vue d'une **ressource de type EIAH** :

- Faire de l'apprentissage adaptatif
- Fournir des feedbacks automatiques
- Indexer des ressources



Didapro 10 - DidaSTIC

Didactique de l'informatique et des STIC

30 janv.-1 févr. 2024 Louvain-la-Neuve (Belgique)

Vers l'analyse de ressources d'apprentissage de la programmation à l'aide d'un référentiel de types de tâches

Sébastien Jolivet, Eva Dechaux,
Anne-Claire Gobard, Patrick Wang