



IAP - Introduction à l'Algorithmique et à la Programmation

Equipe pédagogique

Marie-José Caraty, Denis Poitrenaud, Julien Rossit, Camille Kurtz, Jacques Alès-Bianchetti, Denis Jeanneau

Cours n°6

Les fondamentaux de la programmation impérative

Analyse du projet (2/2)

http://www.tutorialspoint.com/c standard library/

DUT Informatique, 1ère année – IAP – Marie-José Caraty

2019-2020

2

CYCLE DE DEVELOPPEMENT LOGICIEL

Quelques dates qui vous intéresseront...

1/2

Date de remise des dossiers de développement

Le lundi 4 novembre au secrétariat (impératif, retard pénalisé)

Date des recettes

La semaine du 21 octobre en séance de IAP3 (2ème TP de IAP)

DUT Informatique, 1ère année – IAP – Marie-José Caraty

2019-2020

5.

1

CYCLE DE DEVELOPPEMENT LOGICIEL

Quelques dates qui vous intéresseront...

2/2

. RECETTE DE L'APPLICATION

Passage de la recette

Cette semaine, semaine du 14 octobre

Les deux séances de TP (IAP2 et IAP3) sont dédiées au projet

La semaine du 21 octobre

La première séance de TP (IAP2) est dédiée au projet

Recette de l'application

Validation par un observateur (le client)
en l'occurrence votre enseignant de TP
du Sprint de plus haut niveau atteint en phase de développement

Les JDT de recettes sont plus elaborés que les JDT de développement

TEST DE L'APPLICATION

Recette de l'application

Préparation à la recette

Préparez un projet Windows de votre application sur une machine de votre choix (machine de l'IUT, portable personnel accepté).

Recette

En présence de votre enseignant de TP, vous testerez votre programme avec un JDT de recette (inSPn.txt) correspondant au niveau n ($1 \le n \le 5$) du sprint le plus haut que vous avez validé en phase de développement avec les JDT de référence qui vous ont été communiqués pour chacun des sprints. Ce nouveau JDT (de recette) vous sera communiqué par votre enseignant. Vous nommerez run.txt le fichier de sortie.

Si votre fichier run.txt est identique au fichier résultat de référence refSpn.txt (donné par votre enseignant), votre logiciel est accepté : le Sprint#n est validé. Sinon votre logiciel est refusé, vous avez 5 minutes pour corriger vos erreurs. A la suite de ce délai, si vous ne passez toujours pas la recette, vous aurez à valider le Sprint#n-1.

Rem: utilisation d'un comparateur de fichier (diff) pour comparer les fichiers. Ce comparateur affiche les lignes qui diffèrent.

5

DUT Informatique, 1ère année - IAP - Marie-José Caraty

2019-2020

Objectif

Programmer un interpréteur de commandes

1. POURSUITE PAR EQUIPES EN PATINAGE DE VITESSE

Le problème posé

pour la gestion d'une compétition de poursuites par équipes

Dans la **poursuite par équipes**,

deux équipes (Eq1 et Eq2) formées chacune de trois athlètes doivent faire un nombre donné (t) de tours

Eq1

Le terrain de la compétition est une piste circulaire admettant deux points diamétralement opposés (P1 et P2) d'où partent respectivement les trois patineurs des deux équipes (Eq1 et Eq2) engagées dans la poursuite

A chaque tour de piste (passage aux points P1 de l'équipe Eq1 et au point P2 de l'équipe Eq2), le temps réalisé par l'équipe est celui obtenu par le dernier patineur de l'équipe passant le point de référence de l'équipe

Une compétition de poursuites par équipes consiste à organiser *n poursuites par équipes* pour les 2 x n équipes engagées dans la compétition

DUT Informatique, 1ère année – IAP – Marie-José Caraty

2019-2020



MANIPULATION D'UNE TABLE DE DONNEES

Programmer un interpréteur de commandes

Objectif Programmer un interpréteur de commandes pour la gestion d'une compétition de poursuites par équipes

Les commandes sont codées sous forme de chaînes de caractères et entrées en utilisant l'entrée standard ou par redirection d'un fichier texte sur l'entrée standard

Interpréteur des 8 commandes

•	/ 4	0 : 1//4)
■ exit	(1.	Sprint#1)
<pre>— definir_parcours</pre>	(2.	Sprint#4)
definir_nombre_épreuves	(3.	Sprint#5)
<pre>- inscrire_equipe</pre>	(4.	Sprint#1)
<pre>afficher_equipes</pre>	(5.	Sprint#1)
<pre>- enregistrer_temps</pre>	(6.	Sprint#2)
<pre>afficher_temps</pre>	(7.	Sprint#2)
<pre>- afficher_temps_equipes</pre>	(8.	Sprint#3)
<pre>— detection_fin_poursuite</pre>	(7.	Sprint#4)
<pre>— detection_fin_competition</pre>	(8.	Sprint#5)

1.

8

MANIPULATION D'UNE TABLE DE DONNEES

Spécifications

(1/2)

Toute spécification donnée doit être suivie sous peine de pénalisation

Déclaration et documentation des constantes utilisées

Objectif: Eviter les nombres magiques par le nommage et dimensionnement des constantes utilisées

Nombre magique : sont considérés comme nombre magique toute constante numérique littérale différente de 0 et de 1

Spécifications

(2/2)

Autre spécification

 $\mbox{\tt ``Les}$ champs d'information d'une commande seront lus dans la fonction correspondante $\mbox{\tt ``}$

Objectif: Structuration du code par unification

du traitement des commandes

9

DUT Informatique, 1ère année - IAP - Marie-José Caraty

2019-2020

MANIPULATION D'UNE TABLE DE DONNEES

Structuration des données

La structuration des données est obligatoire

Pénalisation en cas de non structuration : 0/20

Rappel

Une structure de données permet de regrouper les données liées à un type d'information

Raison majeure de la nécessité de structuration des données Les fonctions, leur prototypage et leur appel

Passer en paramètre une variable de type structuré permet un accès (en lecture ou écriture suivant le type de passage [in], [out] ou [in out]) à chacun de ses champs

10

DUT Informatique, 1ère année – IAP – Marie-José Caraty

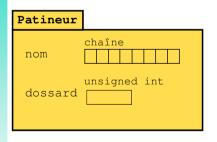
2019-2020

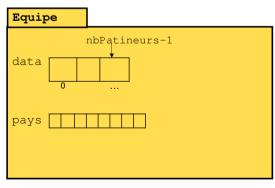
1.

MANIPULATION D'UNE TABLE DE DONNEES

Structuration des données (Srint#1)

(1/1)





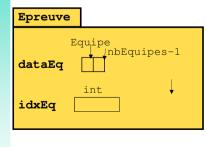
MAN

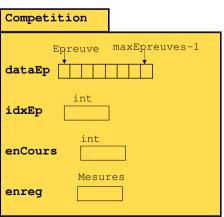
MANIPULATION D'UNE TABLE DE DONNEES

Structuration des données (Sprint#2)

(2/3)

Les types Epreuve, Competition





Les types utiles au Sprint#2

12

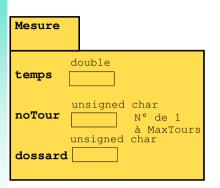
Enregistrement (maxEnr)

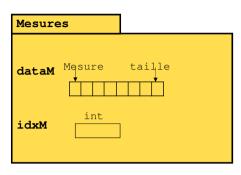
Les types utiles au Sprint#1

Structuration des données (Srint#2)

(2/3)

Les types Mesure et Mesures





taille <- maxEpreuves * maxTours * nbPatineurs * nb Equipes idxM // nombre d'enregistrements // de mesures

Les types utiles au Sprint#2

13

DUT Informatique, 1ère année – IAP – Marie-José Caraty

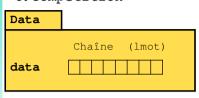
2019-2020

MANIPULATION D'UNE TABLE DE DONNEES

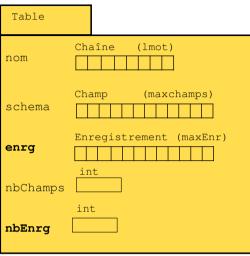
Structuration des données (Sprint#2)

(3/3)

Ajouter les structures de données décrivant les types Epreuve et Compétition



Enregistrement Data (maxChamps) colonne



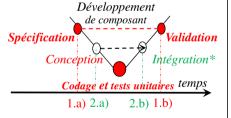
14

DUT Informatique, 1ère année – IAP – Marie-José Caraty

2019-2020



DEVELOPPEMENT DE L'APPLICATION Développement agile



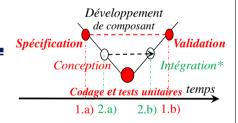
Le développement « agile » est l'un des grands principes du Génie logiciel

> Dépasse le cadre de la programmation/codage Méthodologie de développement « sûr »



16

DEVELOPPEMENT DE L'APPLICATION Développement agile



Choix d'un développement agile Développement par Sprints, chacun représentant un incrément de fonctionnalité

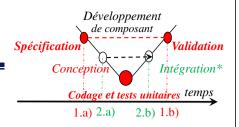
de l'application

Cinq sprints sont définis et pour chacun des Sprints :

- 1) une analyse fonctionnelle
- 2) une spécification
- 3) un codage correspondant
- 4) un test de validation



DEVELOPPEMENT DE L'APPLICATION Validation d'un Sprint



Principe de validation d'un **Sprint#***n* à partir : Jeu de Données de Test (JDT) : inSpn.txt et des sorties attendues : out SPn.txt

L'exécution de l'exécutable du Sprint#n par redirection

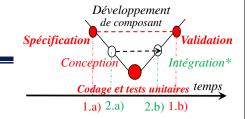
- des entrées à partir de inSpn.txt
- des sorties vers run .t.xt.

Si outSPn.txt coïncide avec run.txt le Sprint#n est 0-défaut, il est validé

17

2019-2020

DEVELOPPEMENT DE L'APPLICATION Validation d'un Sprint



A maîtriser

La redirection des entrées et sorties (cf. Aide à projet sur COMMUN).

PS: Vous pouvez glisser le nom de l'exécutable (nommé "Application" dans le dossier <debug> sous Visual Studio) dans la fenêtre de commande DOS (cmd)

De même, vous pourrez glisser le fichier "inSp#.txt" (où # est le n° de Sprint) dans cette fenêtre

DUT Informatique. 1ère année – IAP – Marie-José Caraty

DUT Informatique, 1ère année - IAP - Marie-José Caraty

2019-2020

4. QUALITE

Eléments de qualité d'un projet

Qualité d'un projet

- Architecture logicielle
- Structuration des données du programme
- Analyse des fonctionnalités/fonctions
- Code
- **Documentation**
- Livrable (dossier de programmation)
- Notion différée : Architecture logicielle

4. QUALITE

18

20

IOCCC-International Obfuscated C Code Contest https://fr.wikipedia.org/wiki/International Obfuscated C Code Contest

IOCCC' 1987 - The winner was...

```
#define -F<00||--F-00--;
int F=00,00=00; main() (F 00(); printf("%1.3f\n",4.*-
F/00/00);}F_00()
 getch();
```

4. QUALITE

Dossier de développement logiciel

□ Respect des spécifications

Respect des spécifications données pour le dossier de développement logiciel (livrable)

Respect du cahier des charges de présentation

Présence dans le dossier a) une page de garde indiquant le nom et le groupe des membres du binôme, l'objet du dossier, illustration, b) une table des matières de l'ensemble du dossier (incluant les annexes) avec la pagination de toutes les rubriques, la pagination est continue du début à la fin du dossier c) présentation de l'application, e) organisation des tests, f) bilan de validation des tests de développement, g) bilan de projet. Brochez vos dossiers.

☐ Présentation de l'application

L'art de synthétiser le projet dans un format donné (1 page). On devra trouver dans cette présentation son rôle fonctionnel (ce que fait l'application), ses entrées et ses sorties (au moins pour le Sprint validé).

22

DUT Informatique, 1ère année - IAP - Marie-José Caraty

2019-2020

4. QUALITE

21

Qualité de code

(1/2)

2019-2020

☐ Lisibilité du code

Code lisible

Code parfaitement indenté (tabulation)

Ligne de code limitée à 80 colonnes

Documentation

Du fichier source (c'est le cartouche)

(en début de fichier : nom de fichier, nom et groupe(s) des auteurs,

DUT Informatique, 1ère année - IAP - Marie-José Caraty

date de création)

Des types structurés et de leurs champs

Des variables importantes

Des fonctions :

rôle de la fonction, des paramètres formels, leur mode ([in], [out] et [in-out]) rôle du paramètre de retour éventuel ([return])

/*

* Calcul du maximum de deux valeurs

* [in] x, 1ere valeur

* [in] y, 2^{ème} valeur

* [out] mx, le maximum

void max(int x, int y, int* mx);

24

7. QUALITE DE CODE

Qualité de code

(2/2)

Commentaires dans le code

Savoir commenter les parties « complexes » du code, les structures de données,...

☐ Absence de nombres magiques

Absence de littéraux (autre que 0 et 1), utilisation des constantes ou de #define motivée par des changements possibles des constantes considérées

Longueur des fonctions

Le main doit être court (à très court).

Un main est à un niveau macroscopique (constitué d'appels de fonction).

Un main trop long traduit un manque au niveau de l'analyse (des fonctions auraient dû être conçues et être appelées par le main).

Par exemple, on n'y développe pas une interface (e.g., un menu).

Une fonction ne doit pas dépasser une vingtaine de lignes. Une fonction trop longue indique souvent un manque d'analyse fonctionnelle et s'accompagne de redondance : dans ce cas, on relit et on analyse le code en visant un niveau mascroscopique (de l'algorithme) pour introduire les fonctions (à coder) et qui seront à appeler (la longueur du code diminuera).



Les tests

Code redondant

Introduire la/les fonction(s) qui généralisent les traitements.

Tests

D'une manière générale, vous devez préciser votre stratégie de test : comment sont organisés vos tests.

Un test comprend: un objectif (ce que l'on veut tester), un JDT (jeu de test). un résultat attendu (résultat de référence), un résultat (trace d'exécution de votre programme) et un bilan de validation (ce que valide le test). Il faut rédiger cette partie.

Dans le rapport, on doit trouver une partie de « Bilan de validation des tests » où on résume tous les tests et ce qu'ils valident.

But : vérifier votre compréhension des tests.

Des JDT « personnels » servent à améliorer la couverture des tests (non demandés)

Bilan de projet

Retour d'expérience. Les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré

25 DUT Informatique, 1ère année – IAP – Marie-José Caraty 2019-2020

26

CYCLE DE DEVELOPPEMENT LOGICIEL

Evaluation des projets - Barème indicatif

[15 pts] NS Note de Sprint (du plus haut niveau #n) validé lors de la recette [1 pt] MOD Modulation d'un Sprint de niveau n validé relativement au source du sprint#n+1 (si présent dans le dossier) Présentation de l'application

[1 pt] PA [1 pt] BV Bilan de validation 1 pt BP Bilan de projet Qualité du code 2 pts] QC

[-1 pt]PN Pénalités de non respect des spécifications (cf. texte du projet) Ex. N° de groupe absent sur la page de garde, pagination absente...

Barème établi sur le Sprint de plus haut niveau validé à la recette

Barème indicatif
NS [/15 points] Barème ér
Sprint 5 : 15 points
Sprint 4 : 12 points
Sprint 3 : 09 points
Sprint 2 : 06 points
Sprint 1 : 03 points
Sprint 1 non atteint : 0 point

ATTENTION 0/20 si le dossier de développement n'est pas rendu 0/20 si le code n'est pas présent dans le dossier

2019-2020

DUT Informatique. 1ère année – IAP – Marie-José Caraty