



IAP - Introduction à l'Algorithmique et à la Programmation

Equipe pédagogique

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

Cours Projet

Les fondamentaux de la programmation impérative

Analyse du projet

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

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

2021-2022

1. CLASSEMENT WTA DES JOUEUSES DE TENNIS

Le problème posé

Objectif Programmer un interpréteur de commandes

pour la gestion du classement WTA des joueuses

professionnelles de tennis

Classement Etabli à partir des résultats des 4 tournois

du Grand Chelem

1. Open d'Australie

2. Internationaux de France de tennis,

3. le tournoi de Wimbledon4. l'US Open de tennis

7 commandes Six commandes de gestion du classement WTA

et une commande de sortie de l'interpréteur (exit)

2 DUT Informatique, 1^{ère} année – IAP – Marie-José Caraty 2021-2022

1. CLASSEMENT WTA DES JOUEUSES DE TENNIS

Les commandes de gestion du classement WTA

Sept commandes

Six commandes de gestion du classement WTA et une commande de sortie de l'interpréteur (exit)

definir_nombre_tournois	[Sp1]	
enregistrement_tournoi	[Sp1]	
affichage_matchs_tournoi	[Sp1]	
affichage_joueuses_tournoi	i [Sp2]	
afficher_matchs_joueuses	[Sp3]	
afficher_classement	[Sp4]	(2 tournois)
afficher_classement	[Sp5]	(≥5 tournois)
exit	- - -	

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

DEVELOPPEMENT DE L'APPLICATION

Développement de l'application à base de Sprints

des cycles de développement appliqués à une fonctionnalité incrémentale de l'application

Cycle de développement logiciel

Quelque soit la taille de l'application considérée, spécification, conception, codage, tests unitaires et validation sont les phases types de tout cycle de développement logiciel pour aboutir à un code fiable, efficace et de qualité

1. a) Spécification

b) Plan de Validation

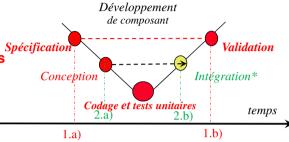
2. a) Conception

b) Plan d'intégration*

3. Codage et tests unitaires

4. Intégration*

5. Validation



Les différentes phases du cycle en V sont assimilables à une décomposition en sous-problèmes. Spécification et Conception permettent de concevoir des tests et leurs résultats attendus. Seule la réussite des tests en phase d'intégration puis de validation permet de valider le composant.

CYCLE DE DEVELOPPEMENT LOGICIEL

Développement par Sprints

(2/2)

6

8

En référence à la programmation agile pour lequel le cycle de développement (analyse fonctionnelle, spécification, codage, test) est court, des Sprints vous sont proposés pour le développement de votre projet.

(1) Chaque Sprint validé correspond à un très bon travail que vous pouvez démontrer. Vous devez en garder la trace : conservez les sources intactes au niveau de votre solution Projet.

Le plus haut sprint (#n, avec 1<=n<=5) validé en phase de recette, sera la base de votre notation de sprint (NS), ses sources seront à ajouter à votre Dossier de Programmation pour une évaluation fondée sur la qualité de code.

Lorsque vous passez à un autre Sprint, créez un nouveau projet où **vous recopierez au niveau des répertoires** toutes les entités logicielles (validées) du précédent projet qui sont nécessaires au nouveau sprint ou à y adapter.

(2) Dans le cas où votre programme ne fonctionne pas sur l'ensemble des fonctionnalités demandées du sprint, joignez ces sources pour un éventuel bonus lié à ce développement.

DEVELOPPEMENT DE L'APPLICATION

Développement par Sprints

(1/2)

Phase de développement

Deux fichiers in Sp#n.txt et out Sp#n.txt sont donnés pour un Sprint#n sur COMMUN (avec $1 \le n \le 5$)

Le premier fichier est un Jeu de Données de Test (JDT), il correspond aux entrées de l'application à tester.

Le second correspond aux sorties attendues de l'application.

si votre programme fonctionne correctement, les résultats de référence doivent coïncider avec le résultat de votre run (trace d'exécution de votre programme).

En fin de la phase de développement du Sprint#n, vous comparerez les sorties de votre programme (run.txt) résultant de l'exécution de votre application à partir du jeu de données de test inSp#n.txt avec le résultat de référence qui vous sera communiqué (outSp#n.txt)

Pour comparer les deux solutions, vous utiliserez un comparateur de fichiers texte (e.g. diffchecker.com ou Diff.jar, archive de l'exécutable d'un programme Java de test d'identité de fichiers)

si les deux fichiers (run.txt et outSp#n.txt) sont identiques, votre application est 0-défaut pour ce test, vous pouvez passer au développement du Sprint suivant sinon corrigez les erreurs de votre application

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

2021-2022

5. CYCLE DE DEVELOPPEMENT LOGICIEL

Données nominales

Les données nominales sont des données « normales » c'est-à-dire valides au sens où elles sont sans erreur

Relativement au type,

relativement aux différentes spécifications des commandes (longueur autorisée pour les chaînes de caractères, domaine de variation des paramètres)

Cohérence de l'ensemble des paramètres de commande

Pour ce projet, les Jeux de Données de Test (JDT) des 5 sprints seront constitués de données nominales On ne vérifiera donc pas la cohérence des données de JDT

5. CYCLE DE DEVELOPPEMENT LOGICIEL

Données nominales

Les constantes du problème

Le nombre maximum de tournois	10
Le nombre de matchs par tournoi	127
Le nombre de joueuses par tournoi	128
La taille maximale d'une chaine de caractères saisie	30

Pour leur déclaration, on utilisera les directives du préprocesseur Exemple: Le nombre maximum de tournois est 10 #define maxTournois 10 // Nb maximum de tournois

Ces directives sont appelés macros (elles peuvent exprimer un calcul) Le traitement du préprocesseur est alors de remplacer dans le source toute occurrence du mot maxTournois par 10 (littéral, constante entière)

9

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

10

12

ANALYSE - Préreguis

Lecture des données

La lecture des données du problème se fera à partir d'une chaîne de caractères (mot) de taille 30 au plus

Déclaration d'une chaîne de caractères (mot) de taille maximale lqMot

```
// chaîne de caractères (mot) de taille max lgMot
char mot[lqMot+1]
```

La chaîne est stockée dans un tableau de caractères (mot) de taille permettant de stocker la chaîne de lgMot caractères et le code de fin de chaîne ('\0') d'où la taille utile lgMot+1

2021-2022

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

2021-2022

Sprint#1



5. CYCLE DE DEVELOPPEMENT LOGICIEL

Sprint #1 - inSp1.txt et outSp1.txt

inSp1.txt

definir_nombre_tournois 1 enregistrement_tournoi Paris 2021

Barty Pera Linette Paquet

Krejcikova Sakkari Pavlyuchenkova Zidansek Krejcikova Pavlyuchenkova affichage_matchs_tournoi exit

outSp1.txt

Paris 2021 64emes de finale Barty Pera Linette Paquet

demi-finales Krejcikova Sakkari Pavlyuchenkova Zidansek Krejcikova Pavlyuchenkova

Voir sur COMMUN les fichiers : inSp1.txt et outSp1.txt

Les sprints de l'application

Cinq sprints représentant cinq incréments de fonctionnalité de l'application

2021-2022

Développement par sprints - Sprint #1

Créez un projet de nom Sprint1 au sein d'une Solution/Projet nommée par exemple ClassementWTA

Premier Sprint

- Analyse fonctionnelle : Limitation à des données de test nominales. Implémentez les commandes "definir nombre tournois". "enregistrement tournoi", "affichage_matchs_tournoi" et "exit"
- Spécification : Définissez les types/structures de données
- a) le type Joueuse ayant pour attributs le nom de la joueuse et ses points cumulés
- b) le type Match ayant pour attributs *l'index de la gagnante* et *l'index de la perdante*
- c) le type Tournoi ayant pour attribut son nom, sa date et le tableau des 127 matchs
- d) le type TournoisWTA le tableau des tournois

Codage: Prototypez et codez (a) la fonction *definir_nombre_tournois* qui permet de définir le nombre de tournois, (b) la fonction *enregistrement_tournoi* qui enregistre les matchs d'un tournoi, (c) la fonction affichage matchs tournoi qui affiche les matchs d'ur tournoi dans l'ordre de l'enregistrement, (d) la fonction exit et (e) le main(): boucle sur les quatre commandes "definir nombre tournois", " enregistrement tournoi", "affichage matchs tournoi" et "exit".

13

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

2021-2022

Le programme principal

ANALYSE - Le main ()

main du Sprint#1 à coder

Boucle de traitement des commandes se terminant à la commande « exit » À compléter de sprint en sprint jusqu'au Sprint #5

Pour identifier la commande, vous utiliserez la fonction stromp de comparaison des chaînes de caractères (cf. <stdlib.h>)

```
int main() {
  while (1) { // ou do {...} while(1);
      // Lecture de la commande (mot)
      scanf("%s", mot);
      // si la commande est "definir_nombre_tournois"
            appeler la fonction definir nombre tournois
      // si la commande est "exit"
      if (strcmp(mot, "exit") == 0) {
        exit(0); // sortie du programme principal
  system("pause"); return 0;
```

Développement par sprints - Sprint #1

C3. Commande d'enregistrement des résultats d'un tournoi

Une ligne composée de la chaîne de caractères "enregistrement tournoi" suivie du nom et de l'année du tournoi. Les résultats du tournoi sont sous la forme de 127 lignes, chacune contenant le résultat d'un match sous forme de deux noms de joueuses, en premier la gagnante, en second la perdante. Les 64 premières lignes correspondent aux matchs des 64 èmes de finale, les 32 lignes suivantes à ceux des 32èmes de finale, les 16 lignes suivantes aux 16èmes de finale, les 8 lignes suivantes aux 8èmes de finale, puis 4 lignes pour les quarts de finale, 2 lignes pour les demi-finales et enfin la dernière ligne pour la finale.

C4. Commande d'affichage des matchs d'un tournoi

Une ligne composée de la chaîne de caractères "affichage matchs tournoi" suivie du nom et de la date d'un tournoi. La commande affiche sur la sortie standard le nom et la date du tournoi sur la première ligne suivis des 127 matchs du tournoi sous la forme d'une ligne par match (nom de la gagnante suivi de celui de la perdante). On ajoutera à ces lignes les lignes indiquant les phases du tournoi (de 64èmes de finale à finale). L'ordre d'affichage est celui de l'enregistrement des matchs. Dans le cas où le tournoi n'existe pas, « tournoi inconnu » est affiché.

5. LES SPRINTS DE L'APPLICATION

Développement par sprints - Sprint #1

Test: Testez votre application par redirection des entrées à partir du fichier inSp1.txt (JDT du Sprint#1) et sa sortie vers le fichier run.txt. Comparez votre fichier run.txt au fichier des résultats de référence outSp1.txt

C1. Commande de sortie du programme

Une ligne composée de la chaîne de caractères "exit".

C2. Commande de définition du nombre de tournois

Une ligne composée de la chaîne de caractères "definir nombre tournois" suivie du nombre de tournois (entier supérieur ou égal à 1 et inférieure ou égal à 10).

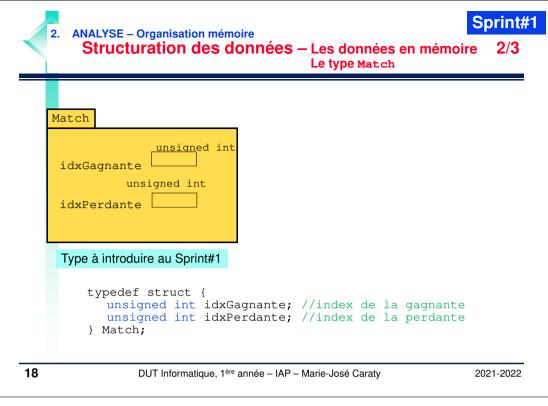
DUT Informatique. 1ère année – IAP – Marie-José Caraty

16

14

Sprint#1

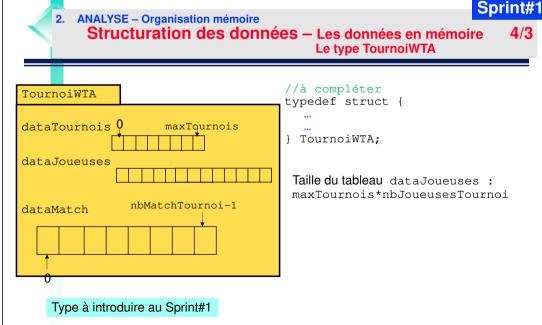
LES SPRINTS DE L'APPLICATION



2. ANALYSE - Organisation mémoire
Structuration des données - Les données en mémoire
Le type Tournoi

typedef struct {
...
} Tournoi;
//à compléter

Accès aux attributs du 2ème match du tournoi dataMatch[1]
Accès à la gagnante du 1er match dataMatch[0].idxGagnante



Type à introduire au Sprint#1

20

Sprint#1 - Prototypage des fonctions

Sprint#1

```
void enregistrement_tournoi (Tournoi* ins);
void enregistrement tournoi (const Tournoi * ins);
// Rappel :
// void enregistrement tournoi (Tournoi ins);
// est un prototype possible
// néanmoins, ce prototype n'est pas optimisé en raison
// de la copie du paramètre effectif dans le paramètre formel
// à l'appel de la fonction enregistrement tournoi
  on préfèrera passer l'adresse de ins (réduisant un appel
// à la copie de 4 octets)
// Idem pour affichage matchs tournoi, auguel on ajoutera const
// pour éviter toute modification du paramètre d'entrée (ins)
// dans la fonction
```

Spécification: les champs d'information d'une commande

sont lus dans la fonction de traitement de la commande

21

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

2021-2022

Sprint#2

5. CYCLE DE DEVELOPPEMENT LOGICIEL

inSp2

Sprint #2 - inSp2.txt et outSp2.txt

definir nombre tournois 1 enregistrement_tournoi Paris 2021 Barty Pera Linette Paquet Krejcikova Sakkari Pavlyuchenkova Zidansek Krejcikova Pavlyuchenkova

afficher_matchs_joueuse Paris 2021 Badosa

outSp2

Badosa Davis Badosa Kovinic Badosa Bogdan Badosa Vondrousova Zidansek Badosa

Voir sur COMMUN les fichiers : inSp2.txt et outSp2.txt

22

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

2021-2022

CYCLE DE DEVELOPPEMENT LOGICIEL

Développement par sprints – Sprint #2

(1/3)

5. LES SPRINTS DE L'APPLICATION

Développement par sprints – Sprint #2

Sprint#2

Créez un projet de nom Sprint2 au sein de la même Solution que le Sprint1 : ClassementWTA. Au niveau des répertoires des projets, recopiez le source du Sprint1 dans le répertoire des sources de Sprint2. Adaptez le source au Sprint#2.

Deuxième Sprint

- Analyse fonctionnelle : Limitation à des données de test nominales. Implémentez la commande "afficher matchs joueuse".
- Spécification : Développez la fonctionnalité associée à la nouvelle commande : mémoriser un temps de chronométrage de la course et afficher les résultats des matchs auxquels a participé une joueuse donnée dans un tournoi donné.
- Codage: Prototypez et codez (a) la fonction afficher_matchs_joueuse et (b) mettre à jour le main () avec cette autres commande. Les champs d'information d'une commande sont lus dans la fonction de traitement de la commande.
- Test : Testez votre application par redirection des entrées à partir du fichier inSp2.txt (JDT du Sprint#2) et sa sortie vers le fichier run.txt. Comparez votre fichier run.txt au fichier des résultats de référence outSp2.txt: si les deux fichiers coïncident, votre Sprint#2 est validé, vous pouvez passer au Sprint#3 sinon corrigez les erreurs.

C6. Commande d'affichage des matchs d'une joueuse

Une ligne composée de la chaîne de caractères "afficher matchs joueuse" suivie du nom et de la date d'un tournoi ainsi que le nom de la joueuse. La commande affiche sur la sortie standard les matchs du tournoi auxquels a participé la joueuse sous la même forme et le même ordre d'affichage que l'enregistrement des matchs. Dans le cas où le tournoi n'existe pas, « tournoi inconnu » est affiché. Dans le cas où la joueuse n'existe pas, « joueuse inconnue » est affiché.

24

Sprint#2

Sprint#2 - Prototypage des fonctions

(3/3)

Sprint#2

void afficher matchs joueuse(const TournoisWTA*);

Spécification : les champs d'information d'une commande sont lus dans la fonction de traitement de la commande

25

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

2021-2022

5. CYCLE DE DEVELOPPEMENT LOGICIEL

Sprint #3 - inSp3.txt et outSp3.txt

outSp3

Zheng 45

Zhu 10 Zidansek 720

inSp3 definir_nombre_tournois 1 Paris 2021 enregistrement_tournoi Paris 2021 Alexandrova 45 Barty Pera Andreescu 10 Linette Paquet Anisimova 10 Arruabarrena 10 Kreicikova Sakkari Zavatska 10 Pavlyuchenkova Zidansek Zhang 10 Krejcikova Pavlyuchenkova

Voir sur COMMUN les fichiers : inSp3.txt et outSp3.txt

26

28

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

2021-2022

CYCLE DE DEVELOPPEMENT LOGICIEL

Développement par sprints - Sprint #3

(1/3)

Sprint#3

Créez un projet de nom Sprint3 au sein de la même Solution que les projets Sprint1 et Sprint2: ClassementWTA. Au niveau des répertoires des projets, recopiez le source du Sprint2 dans le répertoire des sources du Sprint3. Adaptez le source au Sprint#3.

Troisième Sprint

- Analyse fonctionnelle : Limitation à des données de test nominales. Implémentez la commande "affichage joueuses tournoi"
- Spécification : Développez la fonctionnalité associée à la nouvelle commande : affichage par ordre alphabétique des joueuses et des points associés d'un tournoi.
- Codage: Prototypez et codez (a) la fonction affichage_joueuses_tournoi (b) mettre à jour le main () avec cette commande. Les champs d'information d'une commande sont lus dans la fonction de traitement de la commande.
- Test: Testez votre application par redirection des entrées à partir du fichier inSp3.txt (JDT du Sprint#3) et sa sortie vers le fichier run.txt. Comparez votre fichier run.txt au fichier des résultats de référence outSp3.txt: si les deux fichiers coïncident, votre Sprint#3 est validé, vous pouvez passer au Sprint#4 sinon corrigez les erreurs.

5. LES SPRINTS DE L'APPLICATION

affichage_joueuses_tournoi Paris 2021

Développement par sprints – Sprint #3

Sprint#3

C5. Commande d'affichage des 128 joueuses d'un tournoi

Une ligne composée de la chaîne de caractères :

"affichage_joueuses_tournoi" suivie du nom et de la date d'un tournoi.

La commande affiche sur la sortie standard les noms des 128 joueuses suivis du nombre de points gagnés dans le tournoi.

L'ordre d'affichage est l'ordre lexicographique.

Dans le cas où le tournoi n'existe pas, "tournoi inconnu " est affiché.



Sprint#3

inSp4

Barty Pera

Linette Paquet

Barty Suarez Blinkova Babos

Krejcikova Sakkari Pavlyuchenkova Zidansek Kreicikova Pavlvuchenkova

KaPliskova Sabalenka

afficher classement

Barty KaPliskova

exit

definir_nombre_tournois 2

enregistrement tournoi Paris 2021

enregistrement_tournoi Londres 2021

Sprint#3 - Prototypage des fonctions

(3/3)

5. CYCLE DE DEVELOPPEMENT LOGICIEL

Sprint #4 - inSp4.txt et outSp4.txt

Sprint#4

Sprint#3

void affichage joueuses tournoi(TournoisWTA*);

Spécification : les champs d'information d'une commande sont lus dans la fonction de traitement de la commande

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

2021-2022

Sprint#4

outSp4

Krejcikova 2180 Barty 2045 Pavlyuchenkova 1290 KaPliskova 1245 Sabalenka 810

Stosur 10 Swan 10 Teichmann 10 Voegele 10 Volvnets 10 YWang 10 Zavatska 10

Sakkari 765

Voir sur COMMUN les fichiers : inSp4.txt et outSp4.txt

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

2021-2022

29

CYCLE DE DEVELOPPEMENT LOGICIEL

Développement par sprints – Sprint #4

(1/3)

Créez un projet de nom Sprint4 au sein de la même Solution que les autres Sprints: Competition. Au niveau des répertoires des projets, recopiez le source du Sprint#3 dans le répertoire des sources de Sprint#4. Adaptez le source au Sprint#4.

Quatrième Sprint

- Analyse fonctionnelle : Limitation à des données de test nominales et à un nombre de tournois entre 1 et 4. Implémentez la commande " afficher classement ".
- Spécification : Développez la fonctionnalité associée à la commande : afficher le classement WTA des joueuses pour un nombre de tournois entre 1 et 4
- Codage: (a) Prototypez et codez la fonction afficher classement (b) mettre à jour le main () avec cette nouvelle commande. Les champs d'information d'une commande sont lus dans la fonction de traitement de la commande.
- Test : Testez votre application par redirection des entrées à partir du fichier inSp4.txt (JDT du Sprint#4) et sa sortie vers le fichier run.txt. Comparez votre fichier run.txt au fichier des résultats de référence outSp4.txt : si les deux fichiers coïncident, votre Sprint#4 est validé, vous pouvez passer au Sprint#5 sinon corrigez les erreurs.

5. LES SPRINTS DE L'APPLICATION

Développement par sprints – Sprint #4

Sprint#4

C7. Commande d'affichage du classement

Une ligne composée de la chaîne de caractères "afficher classement". La commande affiche sur la sortie standard le classement des joueuses dont le nombre de points cumulés est non-nul sous la forme d'une ligne par joueuse contenant son nom suivi du nombre de points cumulés selon les règles WTA. L'ordre d'affichage est celui du nombre de points avec en cas d'égalité l'ordre lexicographique. Dans le cas où aucun tournoi n'est enregistré. « pas de classement » est affiché.



32

Sprint#4

void afficher_classement(TournoisWTA*);

Spécification : les champs d'information d'une commande sont lus dans la fonction de traitement de la commande

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

2021-2022

outSp5

Osaka 4045 Barty 2405 Kreicikova 2225 Brady 2010 SWilliams 1630 Townsend 10 Vikhlyantseva 10 Volvnets 10 Wickmayer 10 Kreicikova 2585 Bart.v 2495 Raducanu 2180 Osaka 2135 Sabalenka 1710 Swan 10 Tomova 10 Yastremska 10 Zavatska 10

losé Caraty 2021-2022

Sprint#5

CYCLE DE DEVELOPPEMENT LOGICIEL

Développement par sprints – Sprint #5

(1/4)

Créez un projet de nom Sprint5 au sein de la même Solution que les Sprints 1 à 4 : Competition. Au niveau des répertoires des projets, recopiez le source du Sprint4 dans le répertoire des sources de Sprint5. Adaptez le source au Sprint#5.

Cinquième Sprint

- Analyse fonctionnelle : Limitation à des données de test nominales et à un nombre de tournois entre 1 et 10. Implémentez la commande " afficher_classement ".
- Spécification: Développez la fonctionnalité associée à la commande: afficher le classement WTA des joueuses pour un nombre de tournois entre 1 et 10.
- Codage: (a) Prototypez et codez la nouvelle fonction afficher_classement (b) mettre à jour le main () avec cette nouvelle commande. Les champs d'information d'une commande sont lus dans la fonction de traitement de la commande..
- Test: Testez votre application par redirection des entrées à partir du fichier inSp5.txt (JDT du Sprint#5) et sa sortie vers le fichier run.txt. Comparez votre fichier run.txt au fichier des résultats de référence outSp5.txt: si les deux fichiers coïncident, votre Sprint#5 est validé. Bravo!

5. LES SPRINTS DE L'APPLICATION

5. CYCLE DE DEVELOPPEMENT LOGICIEL

inSp5

enregistrement_tournoi New-York 2020

enregistrement_tournoi Melbourne 2021

enregistrement_tournoi Paris 2021

enregistrement_tournoi Londres 2021

enregistrement_tournoi New-York 2021

definir_nombre_tournois 5

Krejcikova Pavlyuchenkova

KaPliskova Kalinina

Osaka Azarenka

Barty Kovinic

Osaka Brady

Barty Pera

Barty Suarez

Barty KaPliskova

Barty Zvonareva

exit.

36

afficher_classement

Raducanu Fernandez

afficher_classement

Sprint #5 - inSp5.txt et outSp5.txt

Développement par sprints - Sprint #5

(2/4)

Sprint#5

C7. Commande d'affichage du classement

Une ligne composée de la chaîne de caractères "afficher_classement". La commande affiche sur la sortie standard le classement des joueuses dont le nombre de points cumulés est non-nul sous la forme d'une ligne par joueuse contenant son nom suivi du nombre de points cumulés selon les règles WTA. L'ordre d'affichage est celui du nombre de points avec en cas d'égalité l'ordre lexicographique. Dans le cas où aucun tournoi n'est enregistré. « pas de classement » est affiché.

5. CYCLE DE DEVELOPPEMENT LOGICIEL

Sprint#5

Sprint#5 - Prototypage des fonctions

(3/4)

Sprint#5

void afficher_classement(TournoisWTA*);

Spécification : les champs d'information d'une commande sont lus dans la fonction de traitement de la commande

37

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

2021-2022

38

5. CYCLE DE DEVELOPPEMENT LOGICIEL

Sprint#5 - Algorithme de tri

(4/4)

Sprint#5

Soit un tableau de n éléments (indexés de 1 à n), le principe du tri par insertion est le suivant : construire une liste triée constitué au début du 1er élément du tableau et que l'on agrandira en y **insérant un à un** les autres éléments (de gauche à droite). Il s'agit d'un **tri stable** conservant l'ordre d'apparition des éléments égaux dans le tableau.

Algorithme de tri par sélection

```
// tri d'un tableau t de taille n (indexé de 1 à n)
procedure tri insertion(tableau t, entier n)
debut_procedure
    pour i allant de 2 à n
      v \leftarrow t[i]
       j ← i
       tant que j > 1 et t[j - 1] > v
         t[j] \leftarrow t[j-1]
         j ← j - 1
       fin tant que
      T[j] ← v
    fin pour
fin_procedure
```

Attention: vous aurez à adaptez l'algorithme à la déclaration en langage C d'un tableau de taille n (indexé de 0 à n-1)

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

2021-2022