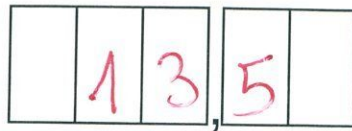


NOM CUESTA

Prénom Quentin

Promo 2017 U M1

Date 14/04/16



CUESTA Quentin
M1 - 2015

MATIÈRE Développement Mobile

Android :

1,5/3

Android
105

6,5/10
7/10

• Question 1: Nous devons créer un broadcaster avec l'intent "com.twitter.tweet" pour pouvoir envoyer une paire (key, value) à l'application Twitter. "message" sera la clé de l'envoi. Nous supposons que Twitter possède un ~~listener~~ broadcast receiver pour attrapper l'envoi de données avec la clé "message".

• Question 2:

<TextView:

<android:android:id="@+id/lastTweet" />

<android:content=" " />

<android:width="match-parent" />

<android:height="match-parent" />

/>

String lastTweet = # receiveLastTweet();

TextView t = (TextView) findViewById(R.id.lastTweet);

t.setText(lastTweet);

filter.

0,5

• Question 3: Nous ne pouvons pas faire cela car les tweets sont broadcastés dans le téléphone par l'application Twitter.

N'importe quelle application ayant un broadcastlistener peut récupérer les tweets. C'est comme la radio.

• Question 4:

3/5

```
for (int i = 0; i < k; i++) {  
    buffer = Ui  
    Ui = MyRule.compute(U0, Ui);  
    U0 = buffer;  
}  
  
int result = Uk;
```

• Question 5:

~~// j'ai oublié de créer un autre EditText pour saisir k~~
~~mais il est égal~~
~~identique aux~~
~~deux autres, il~~
~~si a que l'id qui~~
~~et le hint qui~~
~~change.~~

< EditText :

< android:id = "U₀" />

< android:width = match_parent />

< android:height = match_parent />

< android:hint = "U₀" />

/>

< EditText :

< android:id = "U₁" />

< android:width = match_parent />

< android:height = match_parent />

< android:hint = "U₁" />

/>

< Button :

< android:id = "compute" />

< android:width = match_parent />

< android:height = match_parent />

< android:text = "compute" />

/>

- Question 6: Nous ne pouvons pas se mettre dans un Button car l'algorithme s'exécute dans une tâche Asynchrone. Ce qui veut dire qu'à chaque fois que l'utilisateur va appuyer sur le bouton nous aurons une nouvelle tâche asynchrone. Déjà que les tâches peuvent être longues alors si nous en lançons beaucoup notre application va se faire kill en mémoire car elle en utilisera trop.

- Question 7:

```
public class AsyncCompute (int 03 k, MyRule computeUn(int), int p)
```

extends

AsyncTask {

```
int k, U0, U1 = 0;
```

```
int 03 U0, U1;  
int R = 0;
```

```
public void preExecute () {}
```

```
public void Update () {}
```

```
public void onExecute () {
```

```
int buffer = 0;
```

```
for for (int i = 0; i < k; i++) {
```

```
buffer = U1;
```

```
U1 = MyRule.computeUn(U0, U1);
```

```
U0 = buffer;
```

```
}
```

```
R = U1
```

```
{
```

```
public void postExecute () {
```

```
return Result(R);
```

```
}
```

```
public async compute () {
```

```
k = U[0]; U0 = U[1]; U1 = U[2];
```

```
{
```

```
}
```

U : données passé en paramètres (données d'entrées)

V : action à effectuer

W : variable à retourner.

) type ??

Q. Question 8 : On peut lui envoyer une notification.

Q. Question 9 : C'est une application de réception de SMS. La class étends `IntentReceiver` pour écouter grâce au `Intent-filter` la réception des SMS. Lorsque que cette application reçoit un SMS, elle crée un message grâce à `buf` qui est un `StringBuilder` pour afficher le SMS à l'utilisateur.

cos :

Q. Question 1 : `Character` et `Inventory` seront des classes car nous pouvons trouver tout un tas de fonctions à leur ajouter. Les structures ne contiennent que des attributs.

~~public class~~ ^{class} `Character()` ←

`var Character (`

`let pseudo, sex, age, type;`

`var HP, MP, location;`

`var inventory = new Inventory();` ;) ;

`var Inventory (`

`var object: [int, Object];` ;) ;

NOM CUESTA

Prénom Quentin

Promo 9017 LL M1

Date 14/04/16

MATIÈRE Développement Mobile

Question 2:

```
func initNM ( ps: String, s: String, a: Int, typet: String) {  
  let pseudo: String? ps: "John Doe";  
  let sex: String? s: "Unknown";  
  let age: Int? a: 18;  
  let type: String? t: "warrior";
```

}

```
func initM ( hp: Int, mp: Int, l: [Int] )
```

```
func initM ( l: [Int] ) {
```

```
  var hp = 20;
```

```
  mp = 10;
```

```
  x = l[0];
```

```
  y = l[1];
```

```
  inventory = "empty";
```

```
}
```

1,5

Question 3:

pseudo:	<input type="text"/>
sex:	<input type="text"/>
age:	<input type="text"/>
type:	<input type="text"/>
HP:	<input type="text"/>
MP:	<input type="text"/>
location:	<input type="text"/> <input type="text"/>
inventory:	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>
eco:	<input type="text"/>

0,25

0,125

Question 5:

func smallHPPotition () {

switch character. HP

case 1... 19

character. HP = 20;

case 20... 29

character. HP = 30;

case 30... 39

character. HP = 40;

case 40... 49 ~~HP = 50~~

character. HP = 50;

⋮

case 90... 99

character. HP = 100;

case 0

return;

modify character. modify inventory
return;

1,5

ok

?

• Question 6

```
func recoverHP (way: string) {  
  if way == "small HP" {  
    smallHPotion();  
  }  
  if way == "full HP" || way == "fountain"  
    character.HP = 100;  
}
```

(fountain) 1,75

• Question 7

```
func pickItem (items: [object]) {  
  let item: object? = items[0];  
  if !item  
    print("There is nothing interesting here, Dude!");  
  else  
    character.inventory.add(item);  
}
```

2

