

### Contenu

1. Différents supports .....	1
2. Contraintes .....	3
3. Choix de développement .....	4
4. Développement natif : choix du support .....	5
5. Environnement de développement natif .....	5
6. Exercices récapitulatifs .....	7

---

Vous avez jusqu'à maintenant créé des applications qui avaient pour vocation de s'exécuter sur un ordinateur, soit directement, soit à travers un navigateur. Cette fiche va vous présenter les autres possibilités de développement.

## 1. Différents supports

Quand on parle d'informatique embarquée, on pense souvent aux smartphones. Bien évidemment l'informatique embarquée est loin de se limiter à ce type de supports. Cependant, c'est effectivement l'exemple que l'on développera dans la mise en pratique, car c'est certainement ce qui vous intéresse le plus et c'est aussi un domaine qui prend une réelle ampleur et qui est très demandé par les entreprises. Le fait de savoir développer des applications pour smartphones va donc être valorisant sur votre CV. C'est aussi bien sûr une compétence attendue à l'examen.

### 1A. Smartphones

Les Smartphones sont des téléphones intelligents qui offrent, en plus des fonctions classiques d'un téléphone, des fonctions d'un assistant numérique personnel (PDA). Le fonctionnement est généralement basé sur un écran tactile, éventuellement un clavier. Ils disposent aussi accessoirement d'un gyroscope (pour capter les mouvements), d'un accéléromètre (pour capter la vitesse de déplacement) et éventuellement d'autres capteurs (lumière...).

Tous les smartphones sont programmables, mais de différentes façons, avec des langages différents et sur des systèmes d'exploitation différents. Il existe plusieurs OS mais deux se détachent très nettement des autres : Android et IOS (pour iPhone).

#### Android

Le système d'exploitation Android, développé par Google est de loin le plus utilisé. Il ne se limite d'ailleurs pas aux smartphones et tablettes. Il est possible d'utiliser l'IDE Eclipse pour le développement, mais depuis quelques années, c'est Android Studio qui s'impose. Il faut inclure le SDK Android (ce qui est le cas par défaut, sous Android Studio). Historiquement, le développement se fait en Java ME, mais maintenant il existe une autre alternative avec le langage Kotlin. Il est facilement possible de déployer et tester une application faite pour Android. La publication est très simple (Android Market).

## iPhone

Les téléphones de type iPhone utilisent le système d'exploitation iOS Apple. Cette fois, le développement d'applications se fait dans des conditions un peu différentes. Le langage utilisé est Objective-C qui est une surcouche du langage C. L'IDE adapté est Xcode. Plus récemment, le langage Swift a été adopté. Le déploiement se passe de façon un peu différente pour les iPhones : il est possible de prendre une licence pour un déploiement de petite ampleur, ou choisir l'option de proposer l'application sur l'Apple Store pour une diffusion générale : l'application est alors contrôlée et, si elle est validée, elle est alors diffusée.

## 1B. PDA

Les PDA (Personal Digital Assistant), appelés aussi "Assistants Numériques Personnels", sont des appareils numériques portables qui offrent classiquement les fonctionnalités suivantes : agenda, carnet d'adresses, bloc-notes mais aussi éventuellement calendrier, calculatrice... D'autres possibilités viennent souvent s'ajouter, comme des applications bureautiques, un dictaphone, un navigateur, un lecteur MP3, un gestionnaire de compte bancaire, des jeux...

Le PDA est en fait un ordinateur de poche, qui peut utiliser une connexion (Wi-Fi, infrarouge, bluetooth). Il existe différents systèmes d'exploitation adaptés aux PDA : Microsoft Windows Mobile, Palm OS mais aussi, plus récemment, Linux, Android et aussi iOS.

Suivant les PDA, différents langages sont préconisés. A titre d'exemple on retrouve bien sûr Java (J2ME).

## 1C. Tablettes

La programmation des tablettes est très proche de la programmation des smartphones. Les langages et IDE utilisés sont identiques. La différence provient essentiellement de la taille de l'écran : il faut donc adapter l'application. ou plutôt la concevoir directement pour qu'elle s'adapte à tout type d'écran, donc en responsive.

## 1D. Consoles de jeux

Dans la même logique que les smartphones, il existe des SDK pour les différentes consoles : Xbox se programme en C# avec le kit de développement XNA, PSP se programme en C++, Nintendo DS se programme en C...

## 1E. Et aussi...

Les exemples précédents d'informatique embarquée vous paraissent peut-être évidents. Sachez que la notion d'informatique embarquée va beaucoup plus loin que les téléphones, PDA, tablettes et consoles de jeux.

Il y a bien longtemps que l'informatique s'est imposée dans beaucoup de domaines en intégrant des composants électroniques et des programmes souvent enregistrés de façon permanente en usine. C'est par exemple le cas pour certains appareils électroménagers, pour les voitures (de plus en plus "intelligentes" qui vous disent qu'une portière n'est pas fermée, qu'une ceinture n'est pas bouclée, qui vous aide à vous garer et qui commencent même à conduire à votre place...), et d'une manière plus générale dans le domaine de la domotique (régulation du chauffage, des lumières...).

## 2. Contraintes

En réalité, les bases du développement de l'informatique embarquée sont les mêmes que pour l'informatique classique. La logique "algorithmique" reste la même. Les différences proviennent essentiellement des caractéristiques du support : taille d'écran (ou éventuellement même, absence d'écran), autonomie, rapidité... Par exemple, dans le cadre d'une voiture, suivant le type de programme, la rapidité de réaction est primordiale.

Le but ici n'est pas de parler de tous les cas possibles, mais de s'attacher plus particulièrement à ce qui nous intéresse le plus : le développement sur smartphone. Voici donc les contraintes à prendre en compte.

### 2A. Dispositif de sortie

Lisibilité : prendre en compte la taille réduite de l'écran

Adaptabilité : variations de taille d'un support à l'autre

Il est question ici de l'affichage. Les smartphones possèdent des écrans de petite taille par rapport à un écran normal d'ordinateur. Quelle que soit la résolution, l'information doit rester suffisamment "grande" pour être lisible. Il faut donc penser autrement la disposition et l'affichage de l'information.

Ce point est essentiel aussi lorsqu'il est question de créer une version smartphone d'un site web.

### 2B. Dispositif d'entrée

Clavier : fonctionnement classique

Ecran tactile : capture d'événements tactiles (géré par le SDK)

Il est question ici de la saisie qui peut prendre différentes formes. Soit l'écran est tactile, soit le smartphone possède un clavier, soit les deux dispositifs sont intégrés. Il faut donc soit gérer des événements provenant d'une saisie clavier (ce qui reste classique) soit des événements provenant de l'utilisation de l'écran tactile : ces événements sont gérés en fonction de l'objet graphique sélectionné. C'est le SDK qui gère la capture de ces événements : il ne vous reste plus qu'à programmer le contenu de la procédure événementielle. Evidemment, il faut aussi tenir compte de la taille de l'écran et du doigt, pour adapter la taille des zones de sélection.

### 2C. Autonomie

Minimiser le déchargement de la batterie

Autre aspect nouveau par rapport à un ordinateur classique, c'est l'aspect autonomie. Les smartphones fonctionnent avec une batterie qui va se décharger plus ou moins rapidement, suivant le type d'application exécutée. L'application doit donc être optimisée pour être la moins gourmande possible en ressources et ainsi consommer le moins de batterie possible.

### 2D. Rapidité

Temps de réponse : minimiser les accès internet

Accès à l'information : minimiser les manipulations

Les conditions d'utilisation d'une application sur smartphones sont très différentes d'une application sur un ordinateur classique. L'utilisateur est généralement en mouvement ou dans une situation où l'obtention de l'information doit être instantanée. L'application doit donc être non seulement optimisée en temps de réponse, mais aussi permettre d'accéder à l'information en minimisant le nombre de manipulations.

## 2E. Récupération de données

Enfin, il ne faut pas oublier que le smartphone possède des fonctionnalités supplémentaires par rapport à un ordinateur classique à travers les capteurs qui sont intégrés et des données enregistrées. Ces fonctionnalités peuvent être très utiles suivant les applications.

### Gyroscope

Le gyroscope permet de connaître la position angulaire du smartphone : il permet par exemple de repérer si le smartphone passe de la position verticale à la position horizontale. Cette fonctionnalité est utilisée dans quasiment toutes les applications qui, du coup, permettent de gérer un affichage en portrait ou en paysage. Elle est aussi utilisée dans certains jeux : par exemple les courses de voitures pour simuler les tournants.

On peut imaginer des tas d'applications exploitant le gyroscope. Je me rappelle par exemple d'une application pour les réparateurs d'ascenseurs, qui faisait sonner le smartphone si celui-ci passait à la position horizontale, et en l'absence de réaction de l'utilisateur (s'il n'arrêtait pas la sonnerie) envoyait un SMS d'alerte à l'entreprise pour avertir qu'il y avait peut-être un accident.

### Accéléromètre

Dans la même logique, il existe une autre fonctionnalité qui peut s'avérer bien sympathique : l'accéléromètre. Il permet de mesurer l'accélération linéaire du smartphone. Elle est très utilisée dans les applications liées au sport.

### Géolocalisation

La géolocalisation donne la position du smartphone au niveau géographique. Cette option n'est accessible que si elle a été activée sur le smartphone. Les applications de type GPS utilisent cette fonctionnalité. Il existe aussi des applications qui permettent de repérer où se trouvent vos amis (en fonction des numéros de téléphone qui sont dans votre carnet d'adresses).

### Et aussi au niveau des capteurs

Thermomètre, capteur de lumière, appareil photo...

### Les données enregistrées

Les smartphones permettent d'enregistrer plusieurs sortes de données (contacts, photos, sons, fichiers divers) qui du coup peuvent être gérés par une application, à condition que l'utilisateur en donne l'autorisation d'accès.

## 3. Choix de développement

Lorsqu'il est nécessaire de créer une application pour smartphone ou tablette, il faut réfléchir à la technologie la plus adaptée. Il existe trois possibilités :

### 3A. Développement web

Les smartphones et tablettes accèdent à internet. Une application web responsive permet de s'adapter à tout écran. Une application responsive ne consiste pas juste à adapter le contenu à la taille : elle peut permettre de basculer sur une autre présentation suivant la taille, de sorte que l'information reste lisible.

**Avantages** : développement simple et unique (une application pour le web et tous les mobiles).

**Inconvénients** : pas forcément pratique (obligé d'aller sur internet et sur le site concerné), pas d'utilisation possible sans accès internet (dans le cas d'une application qui ne nécessite pas internet) et utilisation plus complexe des capteurs et données enregistrées dans le smartphone.

### 3B. Application native

Application spécifique à chaque type de mobile.

**Avantages** : rapide et efficace, qui exploite toutes les fonctionnalités du mobile, peut éventuellement fonctionner sans internet et dont l'accès est immédiat

**Inconvénients** : un développement différent par type de support. Vous devez donc développer autant d'applications différentes que de supports que vous voulez exploiter.

### 3C. Technologies multiplateformes

Outils permettant de générer automatiquement du code natif pour tous les supports (Ionic, React Native, Flutter, unity, Xamarin...).

**Avantages** : une seule application pour l'ensemble des supports en obtenant au final des applications natives

**Inconvénients** : code généré assez lourd, solution généralement payante

## 4. Développement natif : choix du support

Si vous décidez de faire un développement natif, vous pouvez soit réaliser plusieurs applications (une par type de support), soit faire des choix stratégiques en fonction de l'utilisation des supports. Ce tableau comparatif peut vous donner une idée et vous aider dans vos choix (ce tableau a été créé avec les données de 2019 ( marché et nb smartphones d'après le site zdnet.fr)) :

	Android	iOS (iPhone)
Marché	86,7%	13,3%
Nb smartphones	1 268 millions	220 millions
Langage	Java ME ou Kotlin	Objective-C et Swift
IDE	Android Studio (ou Eclipse)	Xcode
Plugin	SDK Android	iPhone SDK
Déploiement	oui (.apk)	oui
Publication	Android Market	Apple store
Coût	25 €	80 €/an
Applications	> 1 000 000	> 1 000 000

Il existe d'autres types de smartphones, nettement moins répandus mais globalement, on remarque que les supports Android et iOS couvrent la quasi totalité du marché. Stratégiquement, réaliser le développement juste sur ces deux supports paraît suffisant.

## 5. Environnement de développement natif

Pour développer dans le cadre de l'informatique embarquée, il est nécessaire d'utiliser des outils logiciels, parfois matériels, spécifiques au domaine abordé.

Dans le cadre de cette fiche, nous n'allons aborder que le développement sur smartphone. Voici les éléments à prendre en compte.

## 5A. Choix du langage

Le choix du langage dépend du type de smartphone pour lequel vous désirez développer une application. Le choix n'est donc pas libre. Il faudra alors installer les outils nécessaires suivant le langage utilisé. Par exemple, pour le langage Java : il faut installer le JDK (Java Development Kit).

## 5B. Modules complémentaires

Le langage représente la base mais nécessite des modules complémentaires pour intégrer les fonctionnalités spécifiques au dispositif. Il est généralement question d'un SDK (Software Development Kit) : Android SDK, Appel iPhone SDK...

Le SDK se récupère sur le site correspondant et s'installe pour ensuite être exploité dans l'IDE choisi. Certains IDE intègrent directement le SDK : c'est le cas d'Android Studio.

## 5C. Choix et configuration de l'IDE

Le choix de l'IDE dépend du langage mais normalement, est plus "libre" que le langage lui-même. En effet, il existe souvent plusieurs IDE pour un même langage. Cependant chaque IDE se configure d'une façon précise, donc vous serez assez naturellement tenté d'utiliser l'IDE préconisé par le tutoriel que vous risquez de suivre pour l'apprentissage d'une des technologies choisies.

Vous l'avez compris, une fois l'IDE choisi, il faut normalement le configurer pour y intégrer le SDK correspondant au smartphone à programmer. L'IDE va alors se charger de permettre la création de projets spécifiques à ce SDK et va permettre généralement aussi l'accès à l'émulateur.

Pour l'application que vous allez créer dans les séances, vous pourriez ainsi utiliser Eclipse après l'avoir configuré, mais vu qu'il existe un IDE déjà configuré avec le SDK (Android Studio), on va l'utiliser. Les fonctionnalités sont proches d'Eclipse.

## 5D. Émulateur

L'émulateur permet de visualiser sur l'ordinateur l'affichage du smartphone. C'est un bon moyen de tester l'application en particulier si vous n'avez pas de smartphone. Si vous avez un smartphone, vous pouvez aussi l'utiliser directement comme émulateur pour faire les tests. Ce n'est pas du déploiement (expliqué juste en dessous) mais, juste en branchant le smartphone, il apparaît dans la liste des émulateurs.

Attention, le smartphone utilisé comme émulateur ne fonctionne pas pour des tests lorsque l'application doit accéder à un serveur distant. Autre information à connaître : suivant les émulateurs, il y a parfois quelques surprises entre l'affichage dans l'émulateur et l'affichage sur le smartphone.

## 5E. Déploiement

Le déploiement consiste à installer l'application sur le smartphone. Le transfert pour ensuite gérer l'installation peut se faire de diverses façons : par câble, en bluetooth, par mail... Attention, le déploiement direct n'est pas possible sur les iPhones excepté après avoir pris une licence (pour une utilisation limitée) ou en passant directement par la publication.

## 5F. Publication

Contrairement au déploiement qui ne concerne qu'un seul dispositif, la publication consiste à proposer l'application directement au constructeur qui décide de la publier ou non sur son site de vente d'applications (Apple Store, Android Market...). Vous pouvez même par ce biais, gagner de l'argent (application payante, publicité)!

## 6. Exercices récapitulatifs

Voici quelques petits exercices pour contrôler vos connaissances théoriques. Essayez de les faire sans aide, pour voir si vous avez retenu les notions abordées.

Il y a très peu d'exercices sur ce cours qui est une simple introduction à la notion d'informatique embarquée. Le plus important va être la mise en pratique avec le codage de l'application..

Les exercices 1 et 2 sont juste un contrôle de connaissances. L'exercice 3 est plus dans l'esprit de l'épreuve écrite, avec une demande de réflexion de votre part en fonction d'un contexte. L'application que vous allez coder va vous permettre de passer à la pratique : ce sera l'occasion de voir concrètement le développement d'une application native sous Android.

### ■ Exercice 1

Répondez aux questions suivantes en argumentant.

1. Quels sont les principaux supports utilisés en informatique embarquée ? Citez au moins 3 supports différents.
2. Peut-on parler d'informatique embarquée pour une voiture ?
3. Tous les smartphones sont-ils programmables ?
4. Dans un smartphone, qu'est-ce que le gyroscope ? L'accéléromètre ?
5. Pourquoi est-il important de tenir compte de la taille du dispositif de sortie d'un smartphone, pour le développement ?
6. De quoi a-t-on besoin pour réaliser une application pour un smartphone ?
7. Pourquoi une application sur smartphone doit-elle être optimisée et peu gourmande en ressources ?

## ■ Exercice 2

Choisissez la ou les bonnes réponses en justifiant vos choix :

1. Le développement natif sous Android se fait en :

- a. Objective-C
- b. Java
- c. JavaScriptd.
- d. PHP

2. PDA signifie :

- a. Personal Digital Assistant
- b. Personal Data Access
- c. Personal Data Assistant

3. Le dispositif d'entrée d'un smartphone peut être :

- a. le clavier
- b. l'écran
- c. la souris

4. Les smartphones possèdent généralement des fonctionnalités supplémentaires comme :

- a. la géolocalisation
- b. l'autolocalisation
- c. le gyromètre
- d. le gyroscope
- e. le ralentiromètre
- f. l'accéléromètre

5. Dans un développement pour smartphone, que représente le SDK (que ce soit Android SDK, Appel iPhone SDK...) ?

- a. C'est le langage de développement spécifique au smartphone concerné.
- b. C'est l'ensemble des modules complémentaires pour développer sur le smartphone concerné.
- c. C'est le programme qui permet de publier l'application.

6. Qu'est-ce que l'émulateur ?

- a. Il permet de transférer le programme sur le smartphone
- b. Il permet de visualiser l'exécution sur l'ordinateur
- c. Il permet de réaliser une exécution pas-à-pas du programme

7. Que signifie déploiement ?

- a. C'est le transfert du programme sur un smartphone
- b. C'est le transfert du programme sur plusieurs smartphones
- c. C'est le transfert du programme sur un serveur de déploiement

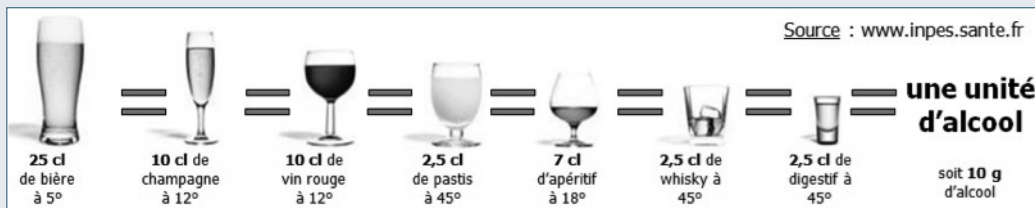
8. Que signifie publication ?

- a. C'est la mise à disposition du programme sur le site officiel de vente
- b. C'est l'édition d'une aide officielle du programme
- c. C'est la vente d'un exemplaire du programme



### ■ Exercice 3

Vous avez décidé de réaliser une petite application Android pour que les personnes puissent tester leur taux d'alcoolémie avant de prendre le volant. Le but de l'application est de préciser si on est un homme ou une femme, de saisir son poids et le nombre de verres bus, sachant qu'un verre représente environ 10g d'alcool, d'après cette représentation :



Vous avez déjà téléchargé et installé tout l'environnement nécessaire pour créer l'application.

Lors de la création de l'IHM, vous avez remarqué que celle-ci est enregistrée au format XML. Des layouts (conteneurs) permettent d'organiser la position des différentes informations. Voici un extrait du code XML de l'IHM :

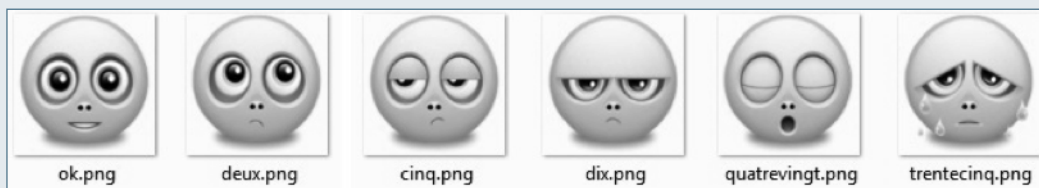
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="center_vertical|center_horizontal" >

    <ImageView
        android:id="@+id/imgSmiley"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_action_search" />

    <TextView
        android:id="@+id/lblTxAlcool"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

Cet extrait permet d'afficher une image parmi les images suivantes :



Et permet d'afficher, en accompagnement de l'image, une zone de texte pour un message précisant le taux de risque.

Vous avez commencé à écrire le code et vous avez une erreur sur chacune des lignes suivantesx:

```
lblTxAlcool.setText("Bonne route !");
imgSmiley.setImageResource(R.drawable.Ok);
```

Vous êtes débutant en programmation sous Android cependant vous savez que c'est du Java et vous avez des bases de connaissances dans ce langage.

**Question 1 : D'après vous quelles sont les erreurs qui ont été faites ?**

En ce qui concerne l'affichage du message, celui-ci doit être sous la forme :

"taux d'alcool : X.Xg !"

Vous savez que la méthode `setText` permet de valoriser la propriété `texte` d'une zone de texte. Vous savez aussi que le `taux` se trouve dans une variable locale appelée `"taux"`. Enfin, en faisant des recherches sur internet, vous avez trouvé un moyen de limiter à un chiffre après la virgule l'affichage d'une valeur de type `float` (ce qui est le cas de la variable `taux`). Il faut pour cela utiliser la méthode statique `format` de la classe `String` qui attend deux paramètres : le premier représente le format attendu, le second représente la variable à formater. Le format attendu est une chaîne sous la forme : `"%.XXY"` où `XX` représente le nombre de chiffres attendus (par exemple `02` pour 2 chiffres) et `Y` représente le format (`f` pour `float`).

**Question 2 : Écrire la ligne de code qui permet de valoriser le bon objet graphique afin d'obtenir le message attendu.**

### ■ Exercice 1

Voici les réponses aux questions :

**1. Quels sont les principaux supports utilisés en informatique embarquée ? Citez au moins 3 supports différents.**

Le premier support qui vient à l'esprit est le smartphone. Mais il est loin d'être le seul. Dans la même catégorie, on trouve les PDA (qui sont finalement des smartphones sans l'option téléphone !) et les tablettes.

Il y a aussi les consoles de jeux.

L'informatique embarquée utilise de nombreux autres supports auxquels on ne pense pas spontanément : les appareils électroménagers, les voitures, et tout ce qui touche à la domotique.

**2. Peut-on parler d'informatique embarquée pour une voiture ?**

Oui. Les voitures sont de plus en plus "intelligentes". Elles possèdent des programmes intégrés, qui réagissent à des événements précis : par exemple des capteurs sont capables de détecter qu'une ceinture n'est pas attachée et vont faire réagir le programme qui va le signifier par une alarme.

**3. Tous les smartphones sont-ils programmables ?**

Oui. Le principe même du smartphone, par rapport à un téléphone simple, est sa capacité à être programmé. Il offre ainsi de nombreuses fonctionnalités qui varient suivant les programmes installés.

**4. Dans un smartphone, qu'est-ce que le gyroscope ? l'accéléromètre ?**

Gyroscope : cette fonctionnalité est basée sur des capteurs qui permettent de repérer les changements de positions angulaires du smartphone (par exemple quand le smartphone passe de la position verticale à la position horizontale). Le gyroscope est très utilisé dans les jeux, mais pas seulement.

L'accéléromètre : cette fonctionnalité est basée sur des capteurs qui permettent de repérer le changement de vitesse de déplacement linéaire du smartphone.

**5. Pourquoi est-il important de tenir compte de la taille du dispositif de sortie d'un smartphone, pour le développement ?**

Le dispositif de sortie représente l'écran. L'écran d'un smartphone est petit : il faut donc, pour que l'information reste lisible, gérer un affichage de taille suffisante et donc réduire au minimum le nombre d'informations à afficher à l'écran. De plus, le dispositif de sortie (l'écran) représente souvent aussi le dispositif d'entrée, donc il faut prévoir des boutons ou autres éléments pouvant être sélectionnés, de taille suffisante par rapport au doigt.

**6. De quoi a-t-on besoin pour réaliser une application pour un smartphone ?**

Il faut développer dans le langage concerné. Le plus simple est d'utiliser un IDE qui va permettre aussi d'intégrer le SDK spécifique au smartphone, l'émulateur pour la visualisation de l'exécution sur l'ordinateur, et même la gestion du déploiement. La configuration de l'IDE est donc importante pour le préparer au développement pour le smartphone. Il est aussi possible de développer une application web responsive pour s'adapter à tout type d'écran, dont les smartphones, mais les possibilités sont plus limitées et la connexion Internet devient obligatoire.

**7. Pourquoi une application sur smartphone doit-elle être optimisée et peu gourmande en ressources ?**

Il faut tenir compte de l'autonomie : une application gourmande va être consommatrice de batterie et va considérablement diminuer l'autonomie du smartphone. De plus, le manque d'optimisation peut avoir aussi pour conséquence des lenteurs d'exécutions, dans un environnement où l'on attend, au contraire, une grande réactivité.

## Exercice 2

Choisissez la ou les bonnes réponses en justifiant vos choix :

1. Le développement natif sous Android se fait en :

- a. Objective-C
- b. Java**
- c. JavaScriptd.
- d. PHP

Objective-C est utilisé pour le développement sous Apple. JavaScript et PHP n'ont rien à voir : ce sont des langages pour le développement de sites. Cependant, en passant par un navigateur, vous pouvez accéder à des sites sur le smartphone. Mais cela ne représente pas une application native.

2. PDA signifie :

- a. Personal Digital Assistant**
- b. Personal Data Access
- c. Personal Data Assistant

En français : Assistant Numérique Personnel.

3. Le dispositif d'entrée d'un smartphone peut être :

- a. le clavier**
- b. l'écran**
- c. la souris

Il n'y a pas de souris pour les smartphones. Certains possèdent un clavier mais la plupart fonctionnent avec un écran tactile.

4. Les smartphones possèdent généralement des fonctionnalités supplémentaires comme :

- a. la géolocalisation**
- b. l'autolocalisation
- c. le gyromètre
- d. le gyroscope**
- e. le ralentiromètre
- f. l'accéléromètre**

Le gyromètre est en fait un synonyme de gyroscope. Les deux autres termes sont inventés.

5. Dans un développement pour smartphone, que représente le SDK (que ce soit Android SDK, Appel iPhone SDK...) ?

- a. C'est le langage de développement spécifique au smartphone concerné.
- b. C'est l'ensemble des modules complémentaires pour développer sur le smartphone concerné.**
- c. C'est le programme qui permet de publier l'application.

Le langage sera par exemple Java, Objective-C... le SDK (Software Development Kit) contient les modules complémentaires qui doivent être intégrés. Le SDK est d'ailleurs plus qu'un simple ensemble de modules : c'est un framework (notion que vous allez découvrir dans la séquence 4 de ce fascicule).

6. Qu'est-ce que l'émulateur ?

- a. Il permet de transférer le programme sur le smartphone
- b. Il permet de visualiser l'exécution sur l'ordinateur**
- c. Il permet de réaliser une exécution pas-à-pas du programme

Sans avoir de smartphone, vous pouvez tester l'application directement sur l'ordinateur en utilisant l'émulateur qui affiche un écran similaire à celui du smartphone.

7. Que signifie déploiement ?

- a. C'est le transfert du programme sur un smartphone**
- b. C'est le transfert du programme sur plusieurs smartphones
- c. C'est le transfert du programme sur un serveur de déploiement

Le fait de transférer le programme sur un smartphone représente le déploiement. Cela permet ensuite d'utiliser l'application directement sur le smartphone.

### 8. Que signifie publication ?

- a. C'est la mise à disposition du programme sur le site officiel de vente
- b. C'est l'édition d'une aide officielle du programme
- c. C'est la vente d'un exemplaire du programme

Cette publication se fait en passant par le site officiel et en demandant une autorisation.

### ■ Exercice 3

Un certain nombre d'informations sont données dans cet exercice pour vous donner un aperçu du principe des questions dans une étude de cas. Mais en réalité, les questions sont très simples. Il faut cependant avoir bien tout lu et analysé pour répondre.

#### Question 1 : D'après vous quelles sont les erreurs qui ont été faites ?

En observant le code par rapport aux informations données, on s'aperçoit qu'à la première ligne de code, il y a une erreur dans le nom du label :

il se nomme lblTxtAlcool dans le code et lblTxAlcool dans le fichier XML (donc un t en moins).

Dans la seconde ligne de code, c'est cette fois une différence de casse :

Ok est écrit avec une majuscule alors que le nom du fichier n'en porte pas. Même sans avoir des connaissances précises en programmation sous Android, on comprend que cette ligne sert à donner le nom de l'image à afficher. Le fichier se nomme "ok.png".

#### Question 2 : Écrire la ligne de code qui permet de valoriser le bon objet graphique afin d'obtenir le message attendu.

Il fallait bien lire les consignes, retrouver le nom de l'objet concerné, appeler correctement la méthode statique et réaliser la concaténation :

```
lblTxAlcool.setText("taux d'alcool : "+String.format("%.01f", taux)+" !");
```



Les cours du CNED sont strictement réservés à l'usage privé de leurs destinataires et ne sont pas destinés à une utilisation collective. Les personnes qui s'en serviraient pour d'autres usages, qui en feraient une reproduction intégrale ou partielle, une traduction sans le consentement du CNED, s'exposeraient à des poursuites judiciaires et aux sanctions pénales prévues par le Code de la propriété intellectuelle. Les reproductions par reprographie de livres et de périodiques protégés contenues dans cet ouvrage sont effectuées par le CNED avec l'autorisation du Centre français d'exploitation du droit de copie (20, rue des Grands-Augustins, 75006 Paris).

**CNED, BP 60200, 86980 Futuroscope Chasseneuil Cedex, France**

© CNED 2021

87D22TEWB1321

