

Misinformation about the coronavirus vaccine is still spreading on Facebook and Instagram, more than a month after Facebook said it would take them down.

Under pressure of containing an avalanche of falsehoods, Facebook announced on 3 December that it would ban debunked claims about the safety and efficacy of vaccines now being distributed worldwide. The company said it removed more than 12m pieces of content from Facebook and Instagram between March and October, and that it worked with factcheckers to place labels on 167 million more pieces of content over the same period.

But researchers say that big Facebook accounts, some with more than half a million followers and long histories of promoting falsehoods, are still openly circulating new posts questioning the vaccine. Meanwhile, prominent anti-vaxxers who have been banned from Facebook are continuing to spread misinformation to hundreds of thousands of people on Instagram, which Facebook owns.

In a December report, the Center for Countering Digital Hate (CCDH), which has tracked the rapid growth of the anti-vaccine movement during the pandemic, argued that "Anything less than the dismantling (taking away/removal) of these individuals' profiles, pages and groups and permanent denial of service is the thing to be done. Experts say the anti-vaccine had used private Facebook groups to convince mothers not to vaccinate their children. They use real data and anecdotes that foster doubt, rather than sharing explicitly false claims. Major anti-vaccine accounts on social media platforms have gained more than 10 million new followers since 2019, according to an analysis by the CCDH.

The Pfizer and Moderna vaccines authorized for distribution in the United States this December each went through a series of rigorous clinical trials. More than 15,000 people received each vaccine in the final phase of these tests, and both vaccines were found to be more than 90% effective at preventing coronavirus. False posts indicate that many died because of the vaccines.

Other activists identified by researchers as "super-spreaders" of coronavirus vaccine misinformation also had their Facebook accounts removed this year, but continue to operate Instagram accounts with hundreds of thousands of followers.

Facebook said in early December that it's using a new policy that was born out of concern that Covid vaccine misinformation could lead to "imminent physical harm", and promised to remove claims from Facebook and Instagram that experts had identified as false. But a month after Facebook launched its aggressive new policy, researchers who study anti-vaccine activism say that false claims are still easy to find, and that many posts with misinformation do not have any additional warning labels.

NewsGuard's health editor also flagged continuing coronavirus false claims on GreenMedInfo, a Facebook page with more than 500,000 followers that has repeatedly been linked to health misinformation. In early December, the page was the first to publish a story falsely claiming that "Pfizer's vaccine had killed two people in the vaccine trial", said John Gregory, NewGuard's deputy editor for health news. In fact, those two deaths had not been linked to the vaccine. When posts on both Worldtruth.tv and GreenMedInfo have received relatively little engagement Facebook said that its efforts to limit distribution are working. But researchers say it is frustrating to see a continuing falsehoods from the same "bad actors" who have been at work since the pandemic began. "These are not new actors in the misinformation space," says Gregory.

"They didn't pop up yesterday." Instead of playing "whack-a-mole" with each new false claim, Gregory says Facebook should take proactive action against accounts based on their history of pushing lies.

First name last name.....
2nd year IT department ID.....

Comprehension questions

1. Read the following and tick the ideas mentioned in the text. (2mk)

- The two new vaccines have gone through clinical trials.
- Social media are not really the space to post false information.
- Fake news represents a problematic issue for Facebook.
- People are glad to hear news about the new vaccines.
- Spreading false information about covid is a serious problem.
- Anti-vaccine people are telling mothers not to vaccinate their children.

2. Focus on paragraphs 2 and 3. Give each one a heading(title) indicating the main idea. (2mks)

Paragraph2.....

Paragraph3.....

3. Say whether the following statements are true or false and justify your answer with details from the text. (3mks)

- a. Misinformation continues to spread, although Facebook promised to stop it. (.....)

- b. The number of Facebook users following the major anti-vaccine accounts is decreasing. (.....)

- c. False posts indicate that many people died because of the virus rather than the vaccine. (.....)

4. Why did Facebook announce that it was going to ban the false claims and posts concerning the new vaccines on December 3. At that time what did the company do to stop the false information?(2mks)

.....

.....

DO NOT WRITE HERE

5. What is the outcome (result) of the measures taken by Facebook in December, 3. Did the company succeed to stop the falsehood?(2mks)

.....

6. According to the CCDH, what should be done to stop false information about the vaccines from circulating on Facebook?(2mks)

.....

7. Why did Facebook decide to use a new policy?(2mks)

.....

8. Are the efforts made by Facebook solving the problem? What does Facebook think and what do researchers think?(2mks)

.....

9. There are 2 expressions in the text which refer to the people who are actively circulating and posting anti-vaccine ideas on social media. What are the two expressions? (1mk)

-
-

Language

1. Turn the following sentences to passive voice. (4mks)

- a. Nowadays, many technologies like smartphones recognize the user's face.
→
- b. Summarizebot recapitulates articles of different types like e-mails and books.
→
- c. Rolling robots carry medicine, food and linens to patients' rooms.
→
- d. Engineers program machines perform complex human tasks
→

2. Put the verbs between parentheses in the right tense. (4mks)

a. You know! It's been tough for me recently. I would love to take some rest and do something different.

b. That's a good idea! Do you have anything in mind?

a. Yes! I (visit) Clarissa next week. I'm hoping we could go out and see movie together.

b. And when are going?

a. Friday afternoon. The train to New York (leave) at 9:30.

b. Aren't you worried about catching the coronavirus?

Oh! Yes! The scientific committee in my town (meet) this week. I think the situation is becoming a bit serious. I'm trying to be very careful. They say a group of engineers (launch) a new application soon and this believe this will help.

3. Combine the pairs of sentences below using the relative pronouns between parentheses. (2mks)

a. Mary Louise designed an amazing application. She is the owner of start-up specialized in AI.

→

b. Face recognition is becoming very popular these days. It can be used to identify people in photos.

→

4. Match the sentence beginnings in column A with the correct endings in column B. (2mks)

Column A	Column B
The CPU is the principal part of any digital computer system:	that can be thought of as the "backbone" of the PC: it holds all the pieces together.
The motherboard is the piece of computer hardware	it is generally composed of the main memory, control unit, and arithmetic-logic unit.
An expansion card is an electronic card/board that	which is capable of performing calculations and logical operations.
Inside a computer, there is an Arithmetic Logic Unit (ALU),	is used to add extra functionality to a computer.

Writing

Write an 8-line paragraph to explain how homputer systems/technologies are helping in the medical field. Use specific examples. (8mks)

.....

.....

.....

.....

.....

.....

Good luck

The text

```
public boolean correspond(Candidat c);
}
```

3. Classes CritereLangue et CritereAge

- Donnez le code d'une classe **CritereLangue** qui est un critère satisfait par tous les candidats qui maîtrisent une langue précisée à la construction du critère.
- Donnez le code d'une classe **CritereAge** qui est un critère satisfait par tous les candidats ayant un âge inférieur à un âge fixé à la construction du critère.

Remarque : Les deux classes implémentent l'interface Critère et dispose chacune d'un constructeur paramétré.

4. Les classes CandidaturePoste et AjoutInterditException

On propose une classe **CandidaturePoste** définie ainsi :

CandidaturePoste	
- LesCandidats : tableau de Candidats	
- String référence	
- int tailleMax	
- int nbrCandidat	
- Critere critereAjout ;	
+CandidaturePoste (String référence, Critere critereAjout, int max)	
+void selectionne(Critere c)	
+void ajout(Candidat c)	

- La méthode *selectionne()* permet d'afficher parmi tous les candidats (contenues dans l'attribut **LesCandidats**) celles qui satisfont le critère donné.

- La méthode *ajout()* permet d'ajouter une candidat dans le tableau **LesCandidats**, si le candidat à ajouter ne respecte pas le critère donné la méthode lève une exception *AjoutInterditException*.

- Ecrire le code de la classe **AjoutInterditException**.
- Ecrire le code de la classe **CandidaturePoste**

EXAMEN

Matière: Programmation OO
Enseignantes: Daghfous N., Mnif S.
Classes : MDW 21 et MDW22

Calculatrice: Non autorisée

Documents: Non autorisés
Durée: 1h30'
Date: Janvier 2021

Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.

EXERCICE 1:

5 PTS

1) Donnez la ou les réponses justes en exécutant la méthode *ArrangerFleurs()*

```
public class Fleur {  
    public void Parfum() {  
        System.out.println("Parfum Fleur");  
    }  
}  
public class Géranium extends Fleur{  
    public void fragrance() {  
        System.out.println("Parfum Géranium");  
    }  
}  
public class Bouquet {  
    public void ArrangerFleurs() {  
        Fleur f1 = new Géranium();  
        f1.Parfum();  
    }  
}
```

- L'affichage produit est : Parfum Fleur
- L'affichage produit est : Parfum Géranium
- Il n'y a pas d'affichage c'est une erreur de compilation.
- Une exception est générée lors de l'exécution de la méthode.

2) En considérant la signature de méthode **public boolean Incrire(int identifiant, String nom)**. Indiquer la(s) définition(s) qui permet(tent) de surcharger correctement cette méthode.

- public boolean Incrire(int volume, String nom)
- private boolean Incrire(int identifiant, String nom)
- public boolean Désinscrire(int identifiant, String nom)
- public boolean Incrire(int identifiant, String nom, Date d)
- public int Incrire(int identifiant, String nom)
- public boolean Incrire(String nom, int identifiant)
- public boolean Incrire (String nom, int identifiant, Date d)

3) En considérant les définitions de classes suivantes, indiquer quelles sont les affectations d'une variable possible.

```
interface Animal {}  
class Vivipare {}  
class Lion extends Vivipare implements Animal {}
```

EXERCICE 2 :

15PTS

- a. Animal var1 = new Animal();
- b. Vivipare var2 = new Vivipare();
- c. Lion var3 = new Vivipare();
- d. Animal var4 = new Vivipare();
- e. Animal var5 = new Lion();

4) En considérant le code suivant, indiquer quelle instruction insérée à la place du marqueur /* Compléter code */ permet d'afficher la valeur de la variable pagesParMin

```
public class Imprimante {
    int niveauEncre;
}

class ImprimanteLaser extends Imprimante {
    int pagesParMin;
    public static void main(String args[]) {
        Imprimante imp1 = new ImprimanteLaser();
        System.out.println(/* Compléter code */);
    }
}
```

- a. (ImprimanteLaser) imp1.pagesParMin
- b. ((ImprimanteLaser) imp1).pagesParMin
- c. ImprimanteLaser.imp1.pagesPerMin
- d. ImprimanteLaser.pagesParMin
- e. imp1.pagesParMin

5) Indiquer le(s) affichage(s) produit(s) par l'exécution du code suivant

```
public class Couleur {
    public static int numero = 20;
    static int getNumero() {
        return numero;
    }
}
class Artiste {
    public static void main(String args[]) {
        Couleur cl = new Couleur();
        System.out.println(cl.getNumero());
        System.out.println(cl.numero);
    }
}
```

- f. Aucun affichage.
- g. L'affichage produit est : 20
- h. Erreur à la compilation.
- i. Génère un warning

1. La classe Candidat

Ecrire et compléter le code de la classe *Candidat*

```
class Candidat
{
    private String nom ;
    private String diplome ;
    private int annnee;
    private String [] langues;
    private int nbreEffectif = 0; //nombre effectif de langues
    private int nbMax; //nombre maximum de langues
    public Candidat (String nom , String dip , int a, int nbreMax)
    { /* à compléter */}
    public String getNom(){ return nom ;}
    public String getDiplome() { return diplome;}
    public int getAnnee() { return annnee;}
    public int getNombreEffectif() { return nbreEffectif;}
    public String[] getLangues () { /* à compléter */}
    public boolean ajoutLangue(String langue) { /* à compléter */}
    public boolean maitriseLangue(String langue) {/* à compléter */}
    public String toString() { /* à compléter */}
}
```

- La méthode *ajoutLangue()* permet d'ajouter une langue, si ceci est possible, dans le tableau langues tout en s'assurant que la langue n'existe pas dans le tableau. Si l'ajout est effectué avec succès elle retourne **true** sinon elle retourne **false**.

Indication : Pour comparer deux chaînes de caractères en ignorant la casse, il est possible d'utiliser la méthode *equalsIgnoreCase* qui existe dans la classe *String* dont la signature est la suivante *boolean equalsIgnoreCase(String anotherString)* où *anotherString* est une variable de type *String*.

- La méthode *maitriseLangue()* permet de retourner true si la langue est dans le tableau langues, false sinon.

- la méthode *toString()* qui permet de retourner une chaîne décrivant un candidat .

2. L'interface Critère

Ecrire l'interface Critere définie ainsi :

```
public interface Critere {
    /* retourne true si et seulement si l'objet o est conforme au critère (on dit que o satisfait le critère) */
}
```

EXAMEN DE:

*****NB : N'écrivez pas votre nom ici *****

== Feuille à rendre ==

```
public class Amis {
```

.....

.....

```
public Amis(int Nmax) {  
    liste = new Abonne[Nmax];  
}
```

```
public void rejoindreAmis(Abonne a) {
```

```
if (nbAbonnes <liste.length) {  
    liste[nbAbonnes]=a;  
    nbAbonnes++;
```

}

```
System.out.println("Liste pleine");
```

```
public String quiAppelle(Abonne a, int d) {
```

```
String nomAmi="" ;
```

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

{

if

omAmi;

EXAMEN DE :

*****NB : N'écrivez pas votre nom ici *****

```
public String quiEnvoiSMS(Abonne a) ..... {  
    String nomAmi="";  
    for (int i=0; i<nbAbonnes; i++) {  
        if (.....)  
            .....  
    }  
    if (nomAmi=="")  
        .....  
    return nomAmi;  
}
```

```
public void rechargerTous(double m) {  
    .....  
}  
public void identifierTous() {  
    .....  
}
```

```
public static void main(String[] args) {  
    Amis friends = new Amis(5);  
    Abonne a = .....  
    friends.rejoindreAmis(a);
```

EXAMEN DE :

*****NB : N'écrivez pas votre nom ici *****

a =

friends.rejoindreAmis(a);

a =

friends.rejoindreAmis(a);

System.out.print("L'encadrant à contacter :");

a =

a.identifie();

try {

} catch (.....) {

}
finally {

}

}

}



EXAMEN

Matière : PROGRAMMATION ORIENTEE OBJET

Enseignantes : Mejdoub S., Guerbouj S., Hedhli J., Mhafdh H.

Classes : DSI2, RSI2

Documents : Non autorisés

Durée : 1h30

Date: 18/01/2021

*Le devoir comporte 7 pages**Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.*

EXERCICE 1

(4 POINTS)

1°/ Donner la trace d'exécution de cette classe.

```
public class test {  
    public static void main (String[] args) {  
        try {  
            System.out.println("just trying");  
            int value = Integer.parseInt ("140bb");  
            System.out.println("end of try clause");  
        } catch (NumberFormatException exception) {  
            System.out.println("catching");  
            System.out.println("End of catching");  
        } finally {  
            System.out.print ("finally clause"); }  
            System.out.print ("end of the main");  
    }}  
}
```

2°/ Soient les classes Mere et Fille suivante :

```
1  public class Mere{  
2      public String nom;  
3      Mere(){ }  
4      Mere(String a){ nom=a; }  
5      public void saluer(String n){  
6          System.out.println("Bonjour "+n); }  
7      .  
8  class Fille extends Mere {  
9      public void f(){  
10         System.out.println("bonsoir "+nom); }  
11      public static void main(String[] args){  
12          Fille f=new Fille();  
13          f.saluer("java") ; } }  
}
```

Ce bout de code :

- S'exécute correctement et affiche bonjour java
- S'exécute correctement et affiche bonsoir
- Génère une erreur de compilation à cause de la ligne 9
- Génère une erreur de compilation à cause de la ligne 11

EXERCICE 2**(16 POINTS)**

Des étudiants n'ont pas beaucoup de solde dans leurs téléphones alors qu'ils ont besoin de contacter leurs encadrant. Ils ont développé une petite application en Java pour déterminer lequel d'eux peut appeler cette personne pour une durée spécifiée.

1°/ Soit la classe abstraite **Abonne** qui possède les caractéristiques suivantes :

- **num**: un entier qui représente le numéro de téléphone de l'abonné.
- **solde** : un réel qui représente le solde en dinar.
- **nom** : une chaîne contenant le nom de l'abonné.
- **getOperateur()** : une méthode abstraite qui retourne une chaîne contenant le nom de l'opérateur de l'abonné.
- **getNom()** : un accesseur qui retourne le nom de l'abonné.
- **getSolde()** : un accesseur qui retourne le solde d'un abonné.
- **rechargerSolde(double m)** : une méthode qui ajoute au solde le montant **m**.
- **identifier()** : une méthode qui affiche tous les attributs d'un abonné.

La classe **Abonne** implémente l'interface **Communication** suivante :

```
public interface Communication {  
    public boolean envoyerSMS();  
    public boolean appeler (Abonne contact, int duree);  
}
```

L'interface contient ces méthodes :

- **envoyerSMS()** : retourne faux si le solde ne permet pas d'envoyer un SMS (sachant que le cout d'envoi d'un SMS est 60 millimes). Si le solde est suffisant, alors, le solde est diminué du cout d'un SMS puis la méthode retourne vrai.
- **appeler (Abonne contact, int duree)** : calcule d'abord le cout d'appel à *contact* pendant la *durée* donnée, puis, retourne Faux si le solde est insuffisant pour passer l'appel. Si le solde est suffisant, alors, elle diminue le solde d'un montant égale au cout de l'appel et retourne Vrai.

NB : Le cout de l'appel est calculé à partir de la formule suivante :

$$Cout = tarif \ minute \times \ durée$$

Sachant que le tarif d'une minute vaut 90 millimes si le contact appelé est du même opérateur sinon le tarif vaut le double (180 millimes).

Donnez le code de la classe **Abonne**.

2°/ Soit la classe **AbonneOrange** qui hérite de la classe **Abonne** et définit les méthodes suivantes :

- un constructeur : initialise tous les attributs hérités de la classe.
- **getOperateur()** : retourne « Orange » si le numéro de l'abonné commence par 5, sinon, elle retourne « Inconnu ». (Astuce : Utilisez la division)

Donnez le code de la classe **AbonneOoredoo**.

On suppose qu'on vous donne aussi une classe **AbonneOoredoo** qui hérite de la classe **Abonne** et définit les méthodes suivantes :

- un constructeur : initialise tous les attributs hérités de la classe.
- **getOperateur()** : retourne « Ooredoo » si le numéro de l'abonné commence par 2, sinon, elle retourne « Inconnu ».

N'écrivez pas la classe AbonneOoredoo.

3°/ Soit une classe d'exception intitulée **SoldeInsuffisantException** qui possède un constructeur avec un paramètre **choix** de type entier.

- Si le choix vaut 1, le constructeur affiche "Soldes insuffisants pour passer cet appel ! Veuillez recharger vos soldes".
- Si le choix vaut 2, le constructeur affiche "Soldes insuffisants pour envoyer un message! Veuillez recharger vos soldes".

Donnez le code de la classe **SoldeInsuffisantException**.

4°/ Complétez le code de la classe **Amis** (voir pages 5, 6, 7) sachant qu'elle possède :

- Un attribut **liste** : qui représente un tableau d'**Abonne**.
- Un attribut **nbAbonnes** : un entier initialisé à 0 et qui représente le nombre effectif d'abonnés dans le tableau **liste**.
- Un constructeur : qui crée le tableau **liste** avec une taille maximale donnée en paramètre.
- Une méthode **identifierTous()** : qui affiche tous les abonnés du tableau d'objets « **liste** ».
- Une méthode **rechargerTous(double m)** : recharge les soldes de tous les abonnés dans le tableau **liste** d'un montant **m**.
- Une méthode **rejoindreAmis(Abonne a)** : qui ajoute l'abonne « **a** » dans le tableau des abonnés « **liste** » si la taille maximale n'est pas atteinte.