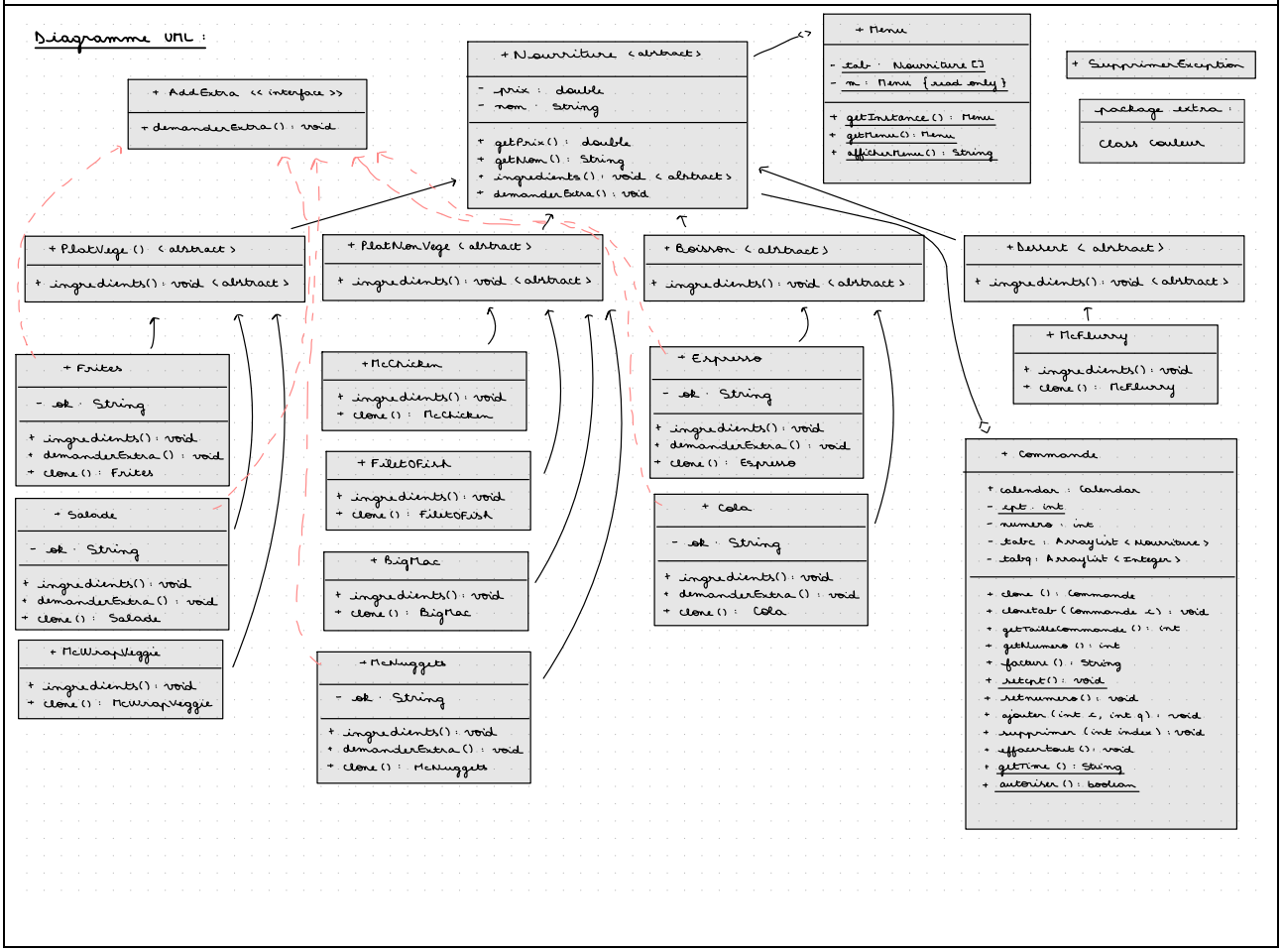
Projet LU2IN002 - 2021-2022

Numéro du groupe de TD/TME : Groupe 7

Nom : KETCHEK	*Nom :* DANQUIGNY
Prénom : Filip	*Prénom :* Julie
N° étudiant : 28613143	*N° étudiant :* 28706902

Thème choisi (en 2 lignes max.)
Simulation d'un guichet McDonald's, avec prise de commande, choix du menu, et affichage de la facture (ticket) à la fin.

Schéma UML des classes vision fournisseur (dessin "à la main" scanné ou photo acceptés)



Checklist des contraintes prises en compte:	*Nom(s) des classe(s) correspondante(s)*
Classe contenant un tableau ou une ArrayList	Commande Menu
Classe avec membres et méthodes statiques	Commande Menu

Classe abstraite et méthode abstraite	Nourriture PlatVege PlatNonVege Boisson Dessert
Interface	AddExtra
Classe avec un constructeur par copie ou clone()	Commande
Définition de classe étendant Exception	SupprimerException
Gestion des exceptions	McDo Commande
Utilisation du pattern singleton	Menu

Présentation brève de votre projet (max. 10 lignes) : texte libre expliquant en quoi consiste votre projet.

L'utilisateur va pouvoir choisir lors de l'exécution du programme, son menu en le composant avec les aliments qu'il souhaite. Il peut ajouter ou retirer les plats (végétarien et non végétarien), boissons, desserts. L'utilisateur doit respecter les heures d'ouvertures du restaurant : s'il est trop tôt ou trop tard le programme va afficher un message pour dire qu'il ne peut plus commander. Pour commander, l'utilisateur entre les chiffres correspondant au numéro du plat ainsi que la quantité souhaitée. Dès que la commande est finie, l'utilisateur peut choisir l'option payer, et il obtiendra sont ticket avec le montant total. Il peut également demander plus d'informations sur la composition de chaque plat (ex : ingrédients du BigMac). L'utilisateur peut également dupliquer une commande qui a déjà été réalisé juste avant. Un fichier .txt est généré avec tous les tickets générés depuis le début du lancement du programme. Pour mettre fin au programme, il peut quitter en appuyant sur un bouton.

Copier / coller vos classes et interfaces à partir d'ici :

```
import extra.Couleur;
public class McFlurry extends Dessert{
  public McFlurry(double p){
    super(p, "McFlurry");
  }

  public String toString(){
    return super.toString();
  }
  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+
    "Glace a base de lait\nEclats de friandises(Oreo, Lotus, Kit Kat Ball, Daim ou
M&M's)\nNappage saveur caramel, chocolat ou coulis de cassis"
    +Couleur.ANSI_RESET);
```

```
  }

  public McFlurry clone(){return new McFlurry(4);}
}
```

```
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.ArrayList;
import extra.Couleur;

public class Commande{
  public static Calendar calendar = new GregorianCalendar();
  public static int heure = calendar.get(Calendar.HOUR_OF_DAY);
  public static int minute = calendar.get(Calendar.MINUTE);
  public static int jour = calendar.get(Calendar.DATE);
  public static int mois = calendar.get(Calendar.MONTH);
  public static int annee = calendar.get(Calendar.YEAR);
  private static int cpt;
  private int numero;
  private ArrayList<Nourriture> tabc;
  private ArrayList<Integer> tabq;

  public Commande(){
    cpt++;
    System.out.println(cpt);
    try {
      Thread.sleep(2000);
    }catch(InterruptedException ex){
      Thread.currentThread().interrupt();
    }
    numero = cpt;
    tabc= new ArrayList<Nourriture>();
    tabq = new ArrayList<Integer>();

  }

  public Commande clone(){
    return new Commande();
  }

  public void clonetab(Commande c){
    for(Nourriture n: c.tabc){
      tabc.add(n.clone());
    }
    for(Integer i: c.tabq){
      tabq.add(i.clone());
    }
  }

  public String toString(){
    String
s=String.format("%sCommande %03d\n%s\n",Couleur.ANSI_YELLOW,numero,Couleur.ANSI_RESET);
    for(int i=0; i<tabc.size();i++){
```

```
      s+=Couleur.ANSI_CYAN+(i+1)+". "+(tabc.get(i)).getNom() +" x"+
tabq.get(i)+Couleur.ANSI_RESET+" | ";
    }
    return s;
  }

  public int getTailleCommande(){
    return tabc.size();
  }

  public int GetNumero(){
    return numero;
  }

  public String Facture(){
    double total =0;
    String s=String.format("Commande %03d\n%s\n",numero,getTime());
    for(int i=0; i<tabc.size();i++){
      total+=((tabc.get(i)).getPrix())*tabq.get(i);
      s+=String.format("%d. %s x%d\n",(i+1),(tabc.get(i)).getNom(),tabq.get(i));
    }
    s+=String.format("\nFacture: %.2f Euros\n\n", total);
    return s;
  }


  public void ajouter(int c, int q){
    if(tabc.size() > 10 ){
      System.out.println(Couleur.ANSI_RED+"Limite de la commande
atteinte.\n"+Couleur.ANSI_RESET);
      try {
        Thread.sleep(1000);
      }catch(InterruptedException ex){
        Thread.currentThread().interrupt();
      }
      return;
    }
    for(int i=0;i<tabc.size();i++){
      if((Menu.getMenu()[c-1]).getClass() == (tabc.get(i)).getClass()){
        int val = tabq.get(i);
        tabq.set(i, val+q);
        if ((Menu.getMenu())[c-1] instanceof AddExtra) (Menu.getMenu()[c-
1]).demanderExtra();
        return;
        }
    }
    tabc.add((Menu.getMenu())[c-1]);
    tabq.add(q);
    if ((Menu.getMenu())[c-1] instanceof AddExtra) (Menu.getMenu()[c-
1]).demanderExtra();

  }
```

```
  public void supprimer(int index) throws SupprimerException{
    if(tabc.size()==0){
      throw new SupprimerException();
    }
    if(index>=tabc.size()){
      System.out.println(Couleur.ANSI_RED+"Index invalide.\n"+Couleur.ANSI_RESET);
      try {
        Thread.sleep(2000);//pauser l'execution du programme
      }catch(InterruptedException ex){
        Thread.currentThread().interrupt();
      }
      return;
    }
    tabc.remove(index);
    tabq.remove(index);
  }

  public void effacertout(){
    tabc.clear();
    tabq.clear();
  }

  public static String getTime(){
    return String.format("%d/%d/%d | %d:%d",jour,mois+1,annee,heure,minute);
  }

  public static boolean autoriser(){
    return (heure >10 && heure < 23);
  }

}
```

```
import java.util.Scanner;
import extra.Couleur;
public class Salade extends PlatVege implements AddExtra{
  private String ok;
  Scanner sc = new Scanner(System.in);
  public Salade(double p){
    super(p, "Salade");
  }

  public String toString(){
    return super.toString();
  }
  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+
    "Feuilles de salades fraiches, francaises et de saison\nLe tout rehausse d'une
delicieuse vinaigrette a l'huile de noisette"
    +Couleur.ANSI_RESET);
  }

  public void demanderExtra(){
```

```
    System.out.println(Couleur.ANSI_BLUE+"ajouter sauce vinegrette(v) | ajouter sauce
crudite(c) | rien(x)"+Couleur.ANSI_RESET);
    ok = sc.next();
        while(!ok.equals("v") && !ok.equals("c") && !ok.equals("x")){
        System.out.println(Couleur.ANSI_RED+"choix non valide"+Couleur.ANSI_RESET);
        ok = sc.next();
      }
  }
  public Salade clone(){return new Salade(4);}
}
```

```
import java.util.Scanner;
import extra.Couleur;

public class Espresso extends Boisson implements AddExtra{
  private String ok;
  Scanner sc = new Scanner(System.in);
  public Espresso(double p){
    super(p, "Espresso");
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+"Grains de cafe finement moulus 100%
Arabica"+Couleur.ANSI_RESET);
  }

  public String toString(){
    return super.toString();
  }
  public void demanderExtra(){
    System.out.println(Couleur.ANSI_BLUE+"sucre(s) | bien sucre(bs) | sans
sucre(x)"+Couleur.ANSI_RESET);
    ok = sc.next();
        while(!ok.equals("s") && !ok.equals("bs") && !ok.equals("x")){
        System.out.println(Couleur.ANSI_RED+"choix non valide"+Couleur.ANSI_RESET);
        ok = sc.next();
      }
  }

  public Espresso clone(){return new Espresso(2);}
}
```

```
import java.util.Scanner;
import extra.Couleur;

public class Frites extends PlatVege implements AddExtra{
  private String ok;
  Scanner sc = new Scanner(System.in);
  public Frites(double p){
    super(p, "Frites");
  }
```

```
  public void demanderExtra(){
    System.out.println(Couleur.ANSI_BLUE+"ajouter Ketchup(k) | ajouter Mayo(m) |
rien(x)"+Couleur.ANSI_RESET);
    ok = sc.next();
        while(!ok.equals("k") && !ok.equals("m") && !ok.equals("x")){
        System.out.println(Couleur.ANSI_RED+"choix non valide"+Couleur.ANSI_RESET);
        ok = sc.next();
      }
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+"Pommes de terre\nHuile
vegetale\nSel"+Couleur.ANSI_RESET);
  }

  public String toString(){
    return super.toString();
  }
  public Frites clone(){return new Frites(3);}
}
```

```
public abstract class Dessert extends Nourriture{

  public Dessert(double p, String n){
    super(p,n);
  }

 public String toString() {
   return super.toString();
 }
 public abstract void ingredients();
}
```

```
import extra.Couleur;
public abstract class Nourriture{
  private double prix;
  private String nom;

  public Nourriture(double p, String n){
    prix = p;
    nom = n;
  }

  public String toString(){
    return String.format("%s%s%s | %s%.1f Euros%s",Couleur.ANSI_PURPLE, nom,
Couleur.ANSI_RESET,
    Couleur.ANSI_BLUE, prix, Couleur.ANSI_RESET);
  }

  public double getPrix(){
```

```
    return prix;
  }

  public String getNom(){
    return nom;
  }

  public abstract void ingredients();

  public void demanderExtra(){};
}
```

```
import extra.Couleur;
public class FiletOFish extends PlatNonVege{
  public FiletOFish(double p){
    super(p, "Filet-O-Fish");
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+
    "Poisson pane croustillant\nUne sauce legerement vinaigree aux oignons et aux
capres\nLe tout dans un pain cuit vapeur"+Couleur.ANSI_RESET);
  }

  public String toString(){
    return super.toString();
  }
  public FiletOFish clone(){return new FiletOFish(4);}
}
```

```
public class SupprimerException extends Exception{
  public SupprimerException(){
    super("La Commande est vide!");
  }
}
```

```
import java.util.Scanner;
import java.util.InputMismatchException;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import extra.Couleur;//notre package couleur

public class Mcdo{
  public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    FileWriter fwriter = null;
    int choix=-1, quantite=-1, index=-1, cpt=0;
    String ok;
    Menu m= Menu.getInstance();//singleton
```

```
    Commande c = new Commande();
    Commande c2 = null;

    if (Commande.autoriser()){
      System.out.print(Couleur.CLEAR);
      System.out.flush();
      System.out.println(Couleur.ANSI_PURPLE+Commande.getTime()
      +"\n\nBienvenu(e) chez
Mcdonald's\n\n\n"+Couleur.ANSI_RESET+Couleur.ANSI_BLUE+"Appuyer pour
continuer"+Couleur.ANSI_RESET);
      while(!sc.hasNextLine()){ok = sc.nextLine();}
    }
    else { System.out.println(Commande.getTime()+"\nLe restaurant est ferme.");
System.exit(0);}

    sc.next();
    do{
      choix=-1; quantite=-1; index=-1;
      System.out.print(Couleur.CLEAR);
      System.out.flush();
      System.out.println(Couleur.ANSI_PURPLE+Commande.getTime()+Couleur.ANSI_RESET);
      if(c2!=null){System.out.println(c2.toString()+"\n");}
      else{System.out.println(c.toString()+"\n");}
      if (c.getTailleCommande() >= 1) System.out.println(Couleur.ANSI_YELLOW+"Desirez
vous autre chose?"
      +Couleur.ANSI_RESET);
      System.out.println(Couleur.ANSI_BLUE+"ajouter un article(+) | supprimer un
article(-) | payer(=) | quitter(x) | plus d'infos(?)"
      +Couleur.ANSI_RESET);


      ok = sc.next();
      while(!ok.equals("=") && !ok.equals("x") && !ok.equals("+") && !ok.equals("-")
&& !ok.equals("?")){
        System.out.print(Couleur.CLEAR);
        System.out.flush();
        System.out.println(c.toString()+"\n");
        System.out.println(Couleur.ANSI_RED+"choix non valide"+
        Couleur.ANSI_RESET+Couleur.ANSI_BLUE+"\najouter un article(+) | supprimer un
article(-) | payer(=) | quitter(x) | plus d'infos(?)"
        +Couleur.ANSI_RESET);
        ok = sc.next();
      }

      if(ok.equals("-")) { //CONDITION -
        System.out.print(Couleur.CLEAR);
        System.out.flush();
        System.out.println(c.toString()+"\n");
        System.out.println(Couleur.ANSI_RED+"Selectionner l'article qui doit etre
supprime"+Couleur.ANSI_RESET);
        while(!sc.hasNextInt() || (index < 1 || index > 10)){
          try{
            index = sc.nextInt();
```

```
            if (index < 1 || index > 10) System.out.println(Couleur.ANSI_RED+"index non
valide, ressayer"+Couleur.ANSI_RESET);
            else break;
          }
          catch(InputMismatchException e){
            System.out.println(Couleur.ANSI_RED+"non entier,
reessayer"+Couleur.ANSI_RESET); }
            sc.next();
          }
          try{c.supprimer(index-1);} // exception personalisee
          catch(SupprimerException e){
            System.out.println(Couleur.ANSI_RED+e.getMessage()+Couleur.ANSI_RESET);
            try {
              Thread.sleep(2000);//pauser l'execution du programme
            }catch(InterruptedException ex){
              Thread.currentThread().interrupt();
            }
          }
        continue;
      }

      if(ok.equals("+")) { //CONDITION +
        System.out.print(Couleur.CLEAR);
        System.out.flush();
        System.out.println(Couleur.ANSI_RED+"Selectionner"+Couleur.ANSI_RESET);
        System.out.println(Menu.afficherMenu());

        while(!sc.hasNextInt() || (choix < 1 || choix > 10)){
          try{
            choix = sc.nextInt();
            if (choix < 1 || choix > 10) System.out.println(Couleur.ANSI_RED+"choix non
valide, ressayer"+Couleur.ANSI_RESET);
            else break;
          }
          catch(InputMismatchException e){
            System.out.println(Couleur.ANSI_RED+"non entier,
ressayer"+Couleur.ANSI_RESET);
            sc.next();
          }
        }
        System.out.println(Couleur.ANSI_BLUE+"combien de pieces?"+Couleur.ANSI_RESET);
        while(!sc.hasNextInt() || quantite < 1){
          try{
            quantite = sc.nextInt();
            if (quantite < 1) System.out.println(Couleur.ANSI_RED+"quantite non valide,
reessayer"+Couleur.ANSI_RESET);
            else break;
          }

          catch(InputMismatchException e){
            System.out.println(Couleur.ANSI_RED+"non entier,
reessayer"+Couleur.ANSI_RESET);
            sc.next();
```

```
          }
        }
        c.ajouter(choix, quantite);
        continue;
      }

      if(ok.equals("=")) { //CONDITION =
        System.out.print(Couleur.CLEAR);
        System.out.flush();
        if(c.getTailleCommande()==0){
          System.out.println(Couleur.ANSI_PURPLE+c.toString()+Couleur.ANSI_RESET+"\n");
          System.out.println(Couleur.ANSI_RED+"La commande est
vide.\n"+Couleur.ANSI_RESET);
          continue;
        }

if(c2!=null){System.out.println(Couleur.ANSI_YELLOW+c2.Facture()+Couleur.ANSI_RESET);}
        else {System.out.println(Couleur.ANSI_YELLOW+c.Facture()+Couleur.ANSI_RESET);}
        try{
          if(cpt==0) fwriter = new FileWriter("historique.txt");
          else fwriter = new FileWriter("historique.txt", true); //si true on ecrase
pas le fichier
          if(c2!=null){fwriter.write(c2.Facture());}
          else{fwriter.write(c.Facture());}
          fwriter.close();
        }
        catch(IOException e){
          System.out.println("Erreur fichier!");
        }

        System.out.println(Couleur.ANSI_BLUE+"Nouvelle commande(+) | Quitter(x) |
Dupliquer la commande precendante(*)"+Couleur.ANSI_RESET);
        ok = sc.next();
        while(!ok.equals("+") && !ok.equals("x") && !ok.equals("*")){
        System.out.println(Couleur.ANSI_RED+"choix non valide"+Couleur.ANSI_RESET);
        ok = sc.next();
      }
        if(ok.equals("+")){
          cpt++;
          c.effacertout();
          c = new Commande();
          continue;
        }
        if(ok.equals("*")){
          cpt++;
          c2 = c.clone();
          c2.clonetab(c);
          continue;
        }
        if(ok.equals("x")){
          c.effacertout();
          System.out.print(Couleur.CLEAR);
          System.out.flush();
```

```
          System.out.println(Couleur.ANSI_PURPLE+"Merci de votre
visite!"+Couleur.ANSI_RESET);
          break;
        }
        if(c2!=null){c2.effacertout();c2 = null;}
      }

      if(ok.equals("x")){ //CONDITION x
        c.effacertout();
        System.out.print(Couleur.CLEAR);
        System.out.flush();
        System.out.println(Couleur.ANSI_PURPLE+"Merci de votre
visite!\n"+Couleur.ANSI_RESET);
        break;
      }
      if(ok.equals("?")) { //CONDITION ?
        System.out.print(Couleur.CLEAR);
        System.out.flush();
        System.out.println(Couleur.ANSI_RED+"Selectionner"+Couleur.ANSI_RESET);
        System.out.println(Menu.afficherMenu());

        while(!sc.hasNextInt() || (choix < 1 || choix > 10)){
          try{
            choix = sc.nextInt();
            if (choix < 1 || choix > 10) System.out.println(Couleur.ANSI_RED+"choix non
valide, ressayer"+Couleur.ANSI_RESET);
            else break;
          }
          catch(InputMismatchException e){
            System.out.println(Couleur.ANSI_RED+"non entier,
ressayer"+Couleur.ANSI_RESET);
            sc.next();
          }
        }
        System.out.print(Couleur.CLEAR);
        System.out.flush();
        System.out.println(Couleur.ANSI_YELLOW+(Menu.getMenu()[choix-
1]).getNom()+"\n"+Couleur.ANSI_RESET);
        (Menu.getMenu()[choix-1]).ingredients();
        System.out.println(Couleur.ANSI_BLUE+"\n\nAppuyer pour
continuer"+Couleur.ANSI_RESET);
        sc.next();
        while(!sc.hasNextLine()){ok = sc.nextLine();}
        continue;
      }

    } while (true);
    sc.close();
  }
}
```

```
import extra.Couleur;
```

```java
public class McWrapVeggie extends PlatVege{
  public McWrapVeggie(double p){
    super(p, "McWrap Veggie");
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+
    "Deux palets panes aux legumes et a l'emmental\nUne sauce onctueuse\nDes oignons frits\nUne tranche d'emmental fondue\nDes rondelles de tomates\nDe la batavia\nLe tout enveloppe dans une galette de ble\n"
    +Couleur.ANSI_RESET);
  }

  public String toString(){
    return super.toString();
  }
  public McWrapVeggie clone(){return new McWrapVeggie(4);}
}
```

```java
public interface AddExtra{

  public void demanderExtra();
}
```

```java
import java.util.Scanner;
import extra.Couleur;

public class McNuggets extends PlatNonVege implements AddExtra{
  private String ok;
  Scanner sc = new Scanner(System.in);
  public McNuggets(double p){
    super(p, "McNuggets");
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+
    "Filets de poulet origine France\nFinement haches et marines\nEnrobes d'une panure\nServis dores et croustillants"
    +Couleur.ANSI_RESET);
  }

  public String toString(){
    return super.toString();
  }

  public void demanderExtra(){
    System.out.println(Couleur.ANSI_BLUE+"ajouter sauce BBQ(bbq) | ajouter sauce moutarde au miel(mm) | rien(x)"+Couleur.ANSI_RESET);
    ok = sc.next();
        while(!ok.equals("bbq") && !ok.equals("mm") && !ok.equals("x")){
        System.out.println(Couleur.ANSI_RED+"choix non valide"+Couleur.ANSI_RESET);
```

```
        ok = sc.next();
      }
  }
  public McNuggets clone(){return new McNuggets(3);}
}
```

```
public abstract class PlatNonVege extends Nourriture{

  public PlatNonVege(double p, String n){
    super(p,n);
  }

  public abstract void ingredients();

  public String toString() {
   return super.toString();
  }
}
```

```
import extra.Couleur;
public class BigMac extends PlatNonVege{
  public BigMac(double p){
    super(p, "BigMac");
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE
    +"Deux steaks haches\nCheddar fondu\nOignons\nCornichons\nLit de salade\nUne sauce
inimitable"+Couleur.ANSI_RESET);
  }

  public String toString(){
    return super.toString();
  }

  public BigMac clone(){return new BigMac(4);}
}
```

```
public abstract class Boisson extends Nourriture{

  public Boisson(double p, String n){
    super(p,n);
  }

 public String toString() {
   return super.toString();
 }
 public abstract void ingredients();

}
```

```java
public abstract class PlatVege extends Nourriture{

  public PlatVege(double p, String n){
    super(p,n);
  }

  public abstract void ingredients();

 public String toString() {
   return super.toString();
 }
}
```

```java
import extra.Couleur;
public class McChicken extends PlatNonVege{
  public McChicken(double p){
    super(p, "McChicken");
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+"Pain special\n"+
    "Specialite panee au poulet\nSalade\nSauce\n"+Couleur.ANSI_RESET);
  }

  public String toString(){
    return super.toString();
  }
  public McChicken clone(){return new McChicken(4);}
}
```

```java
package extra;

public class Couleur{
  public static final String CLEAR = "\033[H\033[2J";
  public static final String ANSI_RESET = "\u001B[0m";
  public static final String ANSI_BLACK = "\u001B[30m";
  public static final String ANSI_RED = "\u001B[31m";
  public static final String ANSI_GREEN = "\u001B[32m";
  public static final String ANSI_YELLOW = "\u001B[33m";
  public static final String ANSI_BLUE = "\u001B[34m";
  public static final String ANSI_PURPLE = "\u001B[35m";
  public static final String ANSI_CYAN = "\u001B[36m";
  public static final String ANSI_WHITE = "\u001B[37m";
  public static final String ANSI_BLACK_BACKGROUND = "\u001B[40m";
  public static final String ANSI_RED_BACKGROUND = "\u001B[41m";
  public static final String ANSI_GREEN_BACKGROUND = "\u001B[42m";
  public static final String ANSI_YELLOW_BACKGROUND = "\u001B[43m";
  public static final String ANSI_BLUE_BACKGROUND = "\u001B[44m";
  public static final String ANSI_PURPLE_BACKGROUND = "\u001B[45m";
  public static final String ANSI_CYAN_BACKGROUND = "\u001B[46m";
  public static final String ANSI_WHITE_BACKGROUND = "\u001B[47m";
```

```
}
```

```
import extra.Couleur;

public class Menu{
  private static Nourriture[] tab;
  private static final Menu m = new Menu(); //instance singleton

  private Menu(){
    tab = new Nourriture[]{new BigMac(4), new McChicken(4), new FiletOFish(4),
      new McNuggets(3), new Frites(3), new McWrapVeggie(4), new Salade(4),
      new McFlurry(4), new Cola(2), new Espresso(2) };
  }

  public static Menu getInstance(){//methode Singleton
    return m;
  }

  public static Nourriture[] getMenu(){
    return tab;
  }

  public static String afficherMenu(){
    String s="\n-------------------------------------------------\n"
    +Couleur.ANSI_GREEN+"     Classiques non vegetariens\n"+Couleur.ANSI_RESET+
    "-------------------------------------------------\n";
    for(int i=0; i < tab.length; i++){
      if(i==4) {
        s+="\n-------------------------------------------------\n"
        +Couleur.ANSI_GREEN+"     Classiques vegetariens\n"+Couleur.ANSI_RESET+
        "-------------------------------------------------\n";
      }
      if(i==7) {
        s+="\n-------------------------------------------------\n"
        +Couleur.ANSI_GREEN+"     Desserts\n"+Couleur.ANSI_RESET+
        "-------------------------------------------------\n";
      }
      if(i==8) {
        s+="\n-------------------------------------------------\n"
        +Couleur.ANSI_GREEN+"     Boissons froides\n"+Couleur.ANSI_RESET+
        "-------------------------------------------------\n";
      }
      if(i>=9) {
        s+="\n-------------------------------------------------\n"
        +Couleur.ANSI_GREEN+"     Boissons chaudes\n"+Couleur.ANSI_RESET+
        "-------------------------------------------------\n";
        s+=String.format("%s%d%s| ", Couleur.ANSI_RED, i+1, Couleur.ANSI_RESET);
      }
      else{s+=String.format("%s%d%s | ", Couleur.ANSI_RED, i+1, Couleur.ANSI_RESET);}
      s+=String.format("%s\n",tab[i].toString());
```

```
    }
    return s;
  }

}
```

```
import java.util.Scanner;
import extra.Couleur;
public class Cola extends Boisson implements AddExtra{
  private String ok;
  Scanner sc = new Scanner(System.in);

  public Cola(double p){
    super(p, "Coca-Cola");
  }

  public String toString(){
    return super.toString();
  }

  public void ingredients(){
    System.out.println(Couleur.ANSI_PURPLE+"Soda\nSirop de mais\nColorant
caramel\nAromes naturels\nCafeine"+Couleur.ANSI_RESET);
  }

  public void demanderExtra(){
    System.out.println(Couleur.ANSI_BLUE+"ajouter une paille(p) | sans
paille(x)"+Couleur.ANSI_RESET);
    ok = sc.next();
        while(!ok.equals("p") && !ok.equals("x")){
        System.out.println(Couleur.ANSI_RED+"choix non valide"+Couleur.ANSI_RESET);
        ok = sc.next();
      }
  }
  public Cola clone(){return new Cola(2);}
}
```