



## Critères d'évaluation du travail pratique #1

Pondération : 17%  
Travail en équipe de 2

Note : ces critères ne sont pas exhaustifs.

Qualité des interfaces graphiques
Menu et items dotés d'accélérateurs
Icône et nom de l'application en titre de la fenêtre
Positionnement de la fenêtre (au milieu) & sa taille ne peut jamais dépasser celle de l'écran
Boîte à outils <ul style="list-style-type: none"><li>• Au moins 6 couleurs de contours et 6 couleurs de remplissage</li><li>• Boutons formes</li><li>• Positionnée en haut de la fenêtre</li><li>• Facilité à identifier les couleurs de contours de celles de remplissage</li><li>• Infobulles</li><li>• Bien présentée (bonne dimension des boutons)</li></ul>
La boîte « À propos » <ul style="list-style-type: none"><li>• Nom logiciel</li><li>• Auteurs</li><li>• Version</li><li>• Logo</li><li>• Bien présentée</li></ul>
Fonctionnement des options du menu
Menu Nouveau <ul style="list-style-type: none"><li>• Demande d'enregistrer le fichier en cours s'il est modifié depuis son dernier enregistrement ou s'il n'a pas encore de nom</li></ul>
Menu Enregistrer <ul style="list-style-type: none"><li>• Utilisation des boîtes de dialogue avec filtre</li><li>• Si le fichier a un nom, l'enregistrement se fait directement, sans boîte de dialogue.</li></ul>
Menu Enregistrer sous <ul style="list-style-type: none"><li>• Utilisation des boîtes de dialogue avec filtre.</li></ul>
Menu Ouvrir <ul style="list-style-type: none"><li>• Demande d'enregistrer le fichier en cours s'il est modifié depuis son dernier enregistrement ou s'il n'a pas encore de nom.</li><li>• Utilisation des boîtes de dialogue avec filtre.</li></ul>
Menu Quitter <ul style="list-style-type: none"><li>• Demande d'enregistrer le fichier en cours s'il est modifié depuis son dernier enregistrement ou s'il n'a pas encore de nom.</li><li>• Demande de confirmer avant de quitter</li></ul>

### Fonctionnement de l'application en général

- Le nom du fichier, suivi du nom de l'application en cours apparaît sur la barre de titre. Si le fichier n'a pas encore de nom ce sera « Sans-titre ».
- Choix des couleurs de contour
- Choix des couleurs de remplissage
- Solution pour la couleur de remplissage quand un trait est sélectionné
- Choix par défaut, au lancement de l'application, de la forme, de la couleur de remplissage et de la couleur de contour.
- Le dessin **progressif** des formes pendant le drag de la souris
- Le dessin se fait sans « **Alliasing** »
- Le dessin se rafraichit correctement
- L'application est **stable** : pas de messages d'erreur qui apparaissent de temps à autre.

### Code des classes métiers : Forme, Cercle, Rectangle, Trait

- Respect des spécifications
- Respect de l'encapsulation: données membres privées, accesseurs au besoin, méthodes internes de type **private**
- Utilisation de **l'héritage des constructeurs** : appeler les constructeurs de la classe parent (super()) lors de l'élaboration des méthodes constructrices (même si la classe parent est abstraite comme Forme)
- Choix des noms pour les accesseurs : si variable est une donnée membre, ce sera getVariable() et setVariable().
- **Intégration optimale** de la classe Ovale dans les classes existantes
- Utilisation du Tag **@Override** lors des redéfinitions des méthodes
- Ne pas utiliser les mots **super** et **this** que si nécessaire.
- Clarté du code : bien indenté, facile à comprendre, pas de code inutile (exemple des `System.out.println()` écrits pour des tests puis oubliés),...

### Code des classes graphiques

- Respect des spécifications : Classe PanDessin
- Taille des méthodes : pas plus que 20 lignes
- Utilisation des tableaux pour les images, couleurs, infobulles...
- Utilisation des constantes significatives : exemple RECTANGLE au lieu de 2

### Qualité générale du code

- Respect de l'encapsulation: données membres privées, accesseurs au besoin, méthodes internes de type **private**.
- Utilisation du Tag **@Override** lors des redéfinitions des méthodes
- Ne pas utiliser les mots **super** et **this** que si nécessaire.
- Clarté du code : bien indenté, facile à comprendre, pas de code inutile (exemple des `System.out.println()` écrits pour des tests puis oubliés),...

### Gestion des évènements

- Utilisation de classes externes pour la gestion des évènements (contrôleurs)

<b>Documentation de l'application (Javadoc)</b>
<ul style="list-style-type: none"><li>• Qualité et pertinence de la documentation</li></ul>
<b>Structuration du projet</b>
<ul style="list-style-type: none"><li>• Utilisation des packages</li><li>• Classes nommées correctement (nom significatif, débutent par une majuscule, deuxième mot débutant par une majuscule)</li></ul>
<b>Modules complémentaires (amélioration en bonus: 6%)</b>
<ul style="list-style-type: none"><li>• Ajout de l'étoile ou une autre forme</li><li>• Ajout de la possibilité de choisir la grosseur du pinceau.</li><li>• Ajout de la possibilité de choisir la couleur dans une boîte de dialogue de choix de couleurs</li></ul>