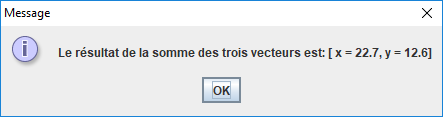
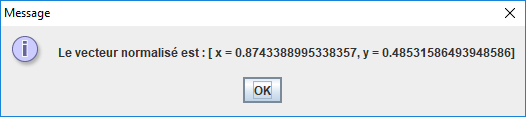
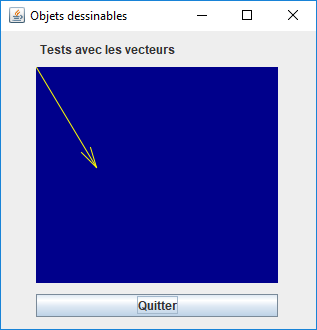
|  |  |
| --- | --- |
| Exercice formatif | **Les objets "dessinables"**  **La classe Vecteur** |

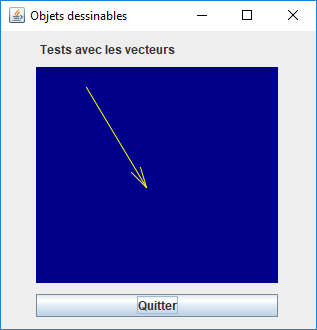
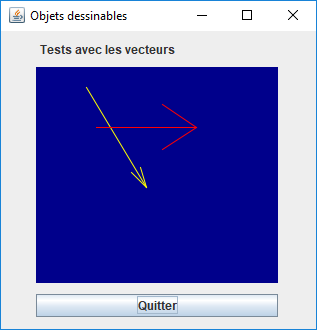
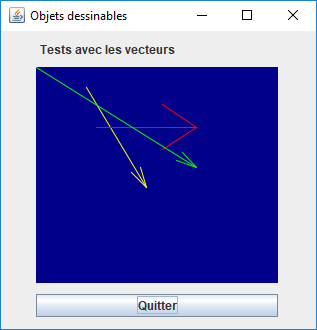
1. Chargez le projet *Vecteurs*. Il contient entre autres la classe *Vecteur* qui permet de représenter un vecteur Euclidien et d'appliquer diverses opérations telles que: addition, soustraction, produit scalaire, calcul de module, etc.

* Notez la double signature qu’offrent plusieurs fonctions.
  + Le projet contient une application *AppDeCalcul.java*. Vous y trouverez du code qui crée trois vecteurs. Complétez l'application ainsi:
    - Placez dans *vec4* la somme suivante : vec1 + vec2 + vec3. Affichez le résultat, en utilisant la méthode *toString* de la classe Vecteur.

* + - Ajoutez du code qui affiche le vecteur résultant de la normalisation du vecteur résultant. Dans votre try/catch, affichez le message "Vecteur: erreur de normalisation" dans le cas où celle-ci surviendrait.
  + On voudrait maintenant pouvoir dessiner un vecteur. Une première approche serait d'ajouter une méthode *dessiner* directement dans la classe Vecteur (ce qui serait plausible). Dans le but de laisser cette classe mathématique intacte, nous choisirons plutôt de créer une seconde classe, qui dérivera de la première, mais qui sera aussi dessinable.

Examinez la classe *VecteurGraphique*.

Dans le composant *DessinDeVecteurs*, créer et dessinez un objet de type *VecteurGraphique* pour représenter un vecteur à l'origine dont les composantes sont (60, 100).

* + Ajoutez le code nécessaire à la méthode *dessiner* de la classe *VecteurGraphique* pour compléter l'autre partie de la tête de flèche. Approche: ajoutez une rotation, puis faites un autre *draw* avec le trait existant.
  + Utilisez la méthode setOrigineXY pour déplacer l’origine du vecteur à *(50,20)*.
  + Dans *DessinDeVecteurs*, dupliquez l'instruction d'appel à *dessiner*. La deuxième instruction devrait redessiner le vecteur au même endroit (donc aucun changement visuel). Mais ce n'est pas le cas … Pourquoi?
* Utilisez *getTransform/setTransform* dans la méthode *dessiner* pour corriger ce problème.
  + L'angle pour la tête de flèche est présentement fixe. On vous demande d'ajouter un *setter* (et pourquoi pas un *getter*) pour permettre de le modifier. Utilisez ce petit truc:
    - Sélectionnez le champ *angleTete* + clic-droit + *Source / Generate*
    -  *getters and setters*. Dans le centre de la fenêtre de dialogue, dans la section *Insertion*, choisir *Last member.*
    - Regardez les méthodes auto-générées à la fin de la classe.
    - Testez: créez un deuxième vecteur (100, 0) dont l'origine est à 60, 60). Ce vecteur sera rouge et l'angle pour la tête sera de 1.2 radians.
    - On vous demande aussi de changer la longueur des segments de tête pour 40 pixels.
  + A l’aide d’une méthode héritée de *Vecteur*, calculez la somme du vecteur jaune et du vecteur rouge. Créer un vecteur graphique à partir du résultat et affichez-le en vert à l’origine.
  + ❓Vous aimeriez pouvoir associer à chaque vecteur un libellé, qui serait dessiné au besoin. Par exemple, le texte "N" apparaitrait près de la tête d'un vecteur de normale, et le texte "g" près d'un vecteur de force gravitationnelle. Quelle serait votre approche?
  + ✪ Pour les pros : Implantez l'option d'ajouter des libellés (voir question précédente). ✪✪ Pour les super-pros : Assurez-vous que jamais le libellé ne sera affiché sur le vecteur lui-même, peu importe son orientation.