Les diagrammes

1. **Le diagramme de packages** permet de décomposer le système en catégories ou parties plus facilement observables, appelés « packages ». Cela permet également d’indiquer les acteurs qui interviennent dans chacun des packages.
2. **Le diagramme de cas d’utilisation** représente les fonctionnalités (ou dit cas d’utilisation) nécessaires aux utilisateurs. On peut faire un diagramme de cas d’utilisation pour le logiciel entier ou pour chaque package.
3. **Le diagramme de classes**

Dans la phase d’analyse, ce diagramme représente les entités (des informations) manipulées par les utilisateurs.  
Dans la phase de conception, il représente la structure objet d’un développement orienté objet.

1. **Le diagramme d’objets** sert à illustrer les classes complexes en utilisant des exemples d’instances.

Une instance est un exemple concret de contenu d’une classe. En illustrant une partie des classes avec des exemples (grâce à un diagramme d’objets), on arrive à voir un peu plus clairement les liens nécessaires

1. **Le diagramme de séquence** permet de décrire les différents scénarios d’utilisation du système.
2. **Le diagramme d’activité** représente le déroulement des actions, sans utiliser les objets. En phase d’analyse, il est utilisé pour consolider les spécifications d’un cas d’utilisation.
3. **Le diagramme de collaboration**(appelé également diagramme de communication) permet de mettre en évidence les échanges de messages entre objets. Cela nous aide à voir clair dans les actions qui sont nécessaires pour produire ces échanges de messages. Et donc de compléter, si besoin, les diagrammes de séquence et de classes.
4. **Le diagramme d’état-transition** permet de décrire le cycle de vie des objets d’une classe.
5. **Le diagramme global d’interaction** permet de donner une vue d’ensemble des interactions du système. Il est réalisé avec le même graphisme que le diagramme d’activité. Chaque élément du diagramme peut ensuite être détaillé à l’aide d’un diagramme de séquence ou d’un diagramme d’activité.
6. **Le diagramme de temps** est destiné à l’analyse et la conception de systèmes ayant des contraintes temps-réel. Il s’agit là de décrire les interactions entre objets avec des contraintes temporelles fortes.
7. **Le diagramme de structure composite** décrit un objet complexe lors de son exécution. Ce diagramme ne sera pas étudié dans ce cours
8. **Le diagramme de composants** décrit tous les composants utiles à l’exécution du système (applications, librairies, instances de base de données, exécutables, etc.).
9. **Le diagramme de déploiement** correspond à la description de l’environnement d’exécution du système (matériel, réseau…) et de la façon dont les composants y sont installés. Ce diagramme ne sera pas étudié dans ce cours.