

Conception Objet en Java avec BlueJ

Une approche pratique

David Barnes et Michael Kölling

Table des matières détaillée

Avant-Propos par James Gosling, Sun Microsystems

Préface pour les enseignants

Java

BlueJ

De vrais objets d'abord

Une approche itérative

Une étude non exhaustive du langage

Une approche par projet

Une progression par concept plutôt que par éléments du langage

Contenu des chapitres

Groupe de discussion

Compléments

Note des traducteurs

Liste des projets

Fondements et orientation objet

Objets et classes

Objets et classes

Création d'objets

Appel des méthodes

Paramètres

Types de données

Instances multiples

État

Définition d'un objet

Interaction entre les objets

Code source

Un autre exemple

Valeurs de retour

Objets agissant comme paramètres

Résumé

Résumé des concepts

Comprendre les définitions de classes

Une billetterie automatique

Étudier le comportement d'une billetterie automatique

Étude d'une définition de classe
 Champs, constructeurs et méthodes
 Champs
 Constructeurs
 Transfert de données par le biais des paramètres
 Affectation
 Méthodes d'accès
 Méthodes de modification
 Impression à partir de méthodes
 Résumé à propos de la billetterie élémentaire
 Réflexions sur la conception de la billetterie
 Les choix : l'instruction conditionnelle
 Autre exemple d'instruction conditionnelle
 Variables locales
 Champs, paramètres et variables locales
 Résumé à propos de la billetterie améliorée
 Un exemple familial
 Résumé
 Résumé des concepts

Interactions entre objets

L'exemple de l'horloge
 Abstraction et modularité
 L'abstraction dans les logiciels
 La modularité dans l'exemple de l'horloge
 Implantation de l'affichage de l'horloge
 Diagrammes de classes et diagrammes d'objets
 Types primitifs et types d'objets
 Le code source de ClockDisplay
 La classe NumberDisplay
 La concaténation
 L'opérateur modulo
 La classe ClockDisplay
 Objets créateurs d'autres objets
 Constructeurs multiples
 Appels de méthodes
 Appels de méthodes internes
 Appels de méthodes externes
 Résumé à propos de l'affichage de l'horloge
 Autre exemple d'interaction entre les objets
 Exemple du système de messagerie
 Le mot clé this
 Le débogueur
 Définition de points d'arrêt
 Pas à pas
 Avancer dans les méthodes
 Nouvelle expérience de l'appel de méthodes
 Résumé
 Résumé des concepts

Groupement d'objets

- Groupement d'objets dans des collections de taille variable
- Un agenda électronique
- Un premier aperçu des classes de bibliothèques
- Exemple d'utilisation d'une bibliothèque
- Structures d'objets et collections
- Numérotation au sein des collections
- Suppression d'un élément d'une collection
- Traitement de l'ensemble d'une collection
- La boucle while
- Parcourir une collection
- Accès à l'index ou itérateurs
- Résumé à propos de l'exemple de l'agenda
- Autre exemple : un système d'enchères
- La classe Lot
- La classe Auction
- Le transtypage
- Collections de taille fixe
- L'analyseur de fichier journal
- Déclarer des variables de tableau
- Créer des objets tableau
- Utiliser des objets tableau
- Analyser le fichier journal
- La boucle for
- Résumé
- Résumé des concepts

Comportements plus complexes

- Documentation pour les classes de bibliothèques
- Le système TechSupport
- Explorer le système TechSupport
- Lire le code
- Lecture de la documentation des classes
- Interfaces ou implantation
- Utiliser des méthodes de classe de bibliothèque
- Vérifier l'égalité des chaînes
- Ajout d'un comportement aléatoire
- La classe Random
- Nombres aléatoires dans un intervalle donné
- Générer des réponses aléatoires
- Paquetages et import
- Des tableaux associatifs pour les associations
- Le concept d'association
- Utiliser un HashMap
- Utiliser une association pour le système TechSupport
- Les ensembles
- Transformation de chaînes en suite de mots
- Fin du système TechSupport
- Rédaction d'une documentation de classe
- Javadoc dans BlueJ

- Éléments de la documentation de classe
- Public ou privé
- Masquer les informations
- Méthodes privées et champs publics
- Découvrir les classes à partir de leurs interfaces
- Variables de classes et constantes
- Le mot clé static
- Constantes
- Résumé
- Résumé des concepts

Des objets bien conçus

- Tester et déboguer
- Tests unitaires dans BlueJ
- Utilisation des inspecteurs
- Test positif versus test négatif
- Automatisation des tests
- Tests de non-régression
- Vérification automatique des résultats de tests
- Modularité et interfaces
- Un scénario de débogage
- Commentaires et style de programmation
- Traces d'exécution manuelles
- Une trace d'exécution de haut niveau
- Vérification de l'état à l'aide d'une trace d'exécution
- Traces d'exécution orales
- Instructions d'affichage
- Activation des informations de débogage
- Débogueurs
- Mise en pratique des techniques
- Résumé
- Résumé des concepts

Conception des classes

- Introduction
- Exemple : le jeu du monde de Zuul
- Introduction au couplage et à la cohésion
- Duplication du code
- Création d'extensions
- La tâche
- Découverte du code source pertinent
- Couplage
- Utilisation de l'encapsulation pour réduire le couplage
- Conception dirigée par les responsabilités
- Responsabilités et couplage
- Localisation des modifications
- Couplage implicite
- Prévoir les évolutions futures
- Cohésion
- Cohésion des méthodes

- Cohésion des classes
- Cohésion et lisibilité
- Cohésion et réutilisation
- Réingénierie
- Réingénierie et test
- Un exemple de réingénierie
- Conception : mode d'emploi
- Exécution sans BlueJ
- Méthodes de classe
- La méthode principale
- Restrictions des méthodes de classe
- Résumé
- Résumé des concepts

Structure des applications

Améliorer la structuration avec l'héritage

- L'exemple DoME
- Classes et objets DoME
- Code source DoME
- Étude de l'application DoME
- Utilisation de l'héritage
- Hierarchies d'héritage
- Héritage en Java
- Héritage et droits d'accès
- Héritage et initialisation
- DoMe : ajout d'autres types d'éléments
- Avantages de l'héritage (pour l'instant)
- Sous-typage
- Sous-typage et sous-classes
- Sous-typage et affectation
- Sous-typage et passage de paramètre
- Variables polymorphes
- La classe Object
- Collections polymorphes
- Types d'éléments
- Le transtypage revisité
- Classes enveloppes
- Hierarchie de collection
- Résumé
- Résumé des concepts

Héritage – Approfondissement

- Le problème : la méthode print de DoME
- Type statique et type dynamique
- Appeler print depuis Database
- Surcharge
- Recherche de méthode dynamique
- Appel super dans les méthodes
- Polymorphisme de méthode

Méthodes de la superclasse Object : toString
 Accès protégé (protected)
 Autre exemple d'héritage avec la surcharge
 Résumé
 Résumé des concepts

De nouvelles techniques d'abstraction

Simulations
 La simulation renards-lapins
 Le projet foxes-and-rabbits (renards et lapins)
 La classe Rabbit
 La classe Fox
 La classe Simulator: initialisation
 La classe Simulator : un pas de simulation
 Améliorer la simulation
 Classes abstraites
 La superclasse Animal
 Méthodes abstraites
 Classes abstraites
 Des méthodes plus abstraites
 Héritage multiple
 Une classe Actor
 Flexibilité par abstraction
 Dessin sélectif
 Acteurs que l'on peut dessiner : héritage multiple
 Interfaces
 Une interface Actor
 Héritage multiple d'interfaces
 Interfaces comme types
 Interfaces comme spécifications
 Un nouvel exemple d'interface
 Classe abstraite ou interface ?
 Résumé sur l'héritage
 Résumé
 Résumé des concepts

Gestion des erreurs

Le projet address-book (carnet d'adresses)
 Programmation défensive
 Interactions client-serveur
 Vérifier les arguments
 Signaler les erreurs de serveur
 Informer l'utilisateur
 Informer l'objet client
 Principes du déclenchement d'exception
 Déclencher une exception
 Classes d'exception
 Effet du déclenchement d'une exception
 Exceptions hors contrôle
 Empêcher la création d'objet

- Gestion des exceptions
- Exceptions sous contrôle : la clause throws
- Capturer une exception : le bloc try
- Déclencher et capturer plusieurs exceptions
- Propager une exception
- La clause finally
- Définir de nouvelles classes d'exception
- Correction et prévention des erreurs
- Corriger les erreurs
- Éviter les erreurs
- Étude de cas : entrées/sorties de texte
- Lecteurs (readers), écrivains (writers) et flux (streams)
- Le projet address-book-io (E/S et carnet d'adresses)
- Sortie de texte sur fichier avec FileWriter
- Entrée de texte à partir de fichier avec FileReader
- Sérialisation d'objet
- Résumé
- Résumé des concepts

Conception des applications

- Analyse et conception
- La méthode noms/verbes
- Le système de réservation de places de cinéma
- Découvrir les classes
- Utiliser les fiches CRC
- Scénarios
- Conception des classes
- Concevoir les interfaces de classes
- Concevoir l'interface utilisateur
- Documentation
- Coopération
- Prototypage
- Croissance du logiciel
- Le modèle de la chute d'eau
- Développement itératif
- Utiliser les patrons de conception
- Structure d'un patron
- Le décorateur
- Le singleton
- La méthode fabrique
- L'observateur
- Résumé sur les patrons
- Résumé
- Résumé des concepts

Une étude de cas

- Étude de cas
- Description du problème
- Analyse et conception
- Découvrir les classes

- Utiliser les fiches CRC
- Scénarios
- Conception des classes
- Conception des interfaces de classes
- Collaborations
- L'implantation résumé
- Tests
- Encore quelques remarques
- Développement itératif
- Étapes de développement
- Une première étape
- Tester la première étape
- Une étape ultérieure du développement
- D'autres idées de développement
- Réutilisation
- Un autre exemple
- Pour aller plus loin

Travailler avec un projet BlueJ

- Installer BlueJ
- Ouvrir un projet
- Le débogueur BlueJ

Types de données Java

- Types primitifs
- Types objet
- Classes enveloppes

Structures de contrôle Java

- Instructions de sélection
- if-else
- switch
- Boucles
- while
- do-while
- for
- Exceptions

Opérateurs Java

- Expressions arithmétiques
- Expressions booléennes

Utiliser Java sans BlueJ

- Exécution de programme Java sans BlueJ
- Créer des fichiers jar exécutables
- Développer sans BlueJ

Configurer BlueJ

- Généralités
- Version française

java bluej. share|improve this question. Browse other questions tagged java bluej or ask your own question. asked. 3 years, 3 months ago. BlueJ is a Java development environment that runs on top of the Sun Microsystems Java Development Kit making use of the standard compiler and virtual machine. It allows readers to create objects of any class and interact with their methods. For the first time, the traditionally difficult concepts of objects and classes are brought alive in an easily manipulable visual form. This truly "objects first" approach within the customized BlueJ environment will revolutionize the way programming is learned. The book includes a copy of BlueJ. Takes a project driven approach to problem solving-