

Compte Rendu des Réunions

Semaine du 05 – 11 Janvier 2026

Équipe : **43G8** | Rédacteur : **TOUATI Ayoub**

Récapitulatif

N°	Date	Horaires	Durée	Objet
3	Dim. 11/01	14h00–17h30	3h 30	Point d’avancement et finalisation

Total (Volume Horaire de toutes les réunions) : 10 h 58

Réunion 3 – Dim 11/01 (14h00–17h30)

Lieu : Visioconférence (Meet)

Présents : 5/5

Objectifs

Présentation de l’état d’avancement global du projet, validation des tests et finalisation de la génération de code avec prise en charge de l’orienté objet.

Ordre du jour

1. Tour de table et présentation de l’avancement individuel
2. Automatisation et intégration continue des tests
3. Avancement sur les tests (lexicaux, syntaxiques, contextuels, génération de code)
4. État d’avancement de l’Étape C (sans objet) : Génération de code (finalisation)
5. Implémentation de la partie orientée objet
6. Discussion des prochaines étapes

Infrastructure de Tests et Intégration Continue

Travaux réalisés :

- Création de scripts bash automatisant l’exécution des tests de manière responsable, en ne lançant que les tests nécessaires par rapport aux modifications effectuées depuis la dernière exécution :
 - `test_lex_energy.sh` – Tests d’analyse lexicale
 - `test_syntax_energy.sh` – Tests d’analyse syntaxique
 - `test_context_energy.sh` – Tests d’analyse contextuelle
 - `test_gencode_energy.sh` – Tests de génération de code
 - `test_all_energy.sh` – Exécution globale de tous les tests
 - Scripts auxiliaires `not_basic-*.sh` associés
- Mise en place de l’intégration continue sur GitLab :
 - Création de pipelines CI/CD déclenchés à chaque `push`

- Affichage des résultats sous forme `n passed / m failed`
- Ajout d'un README avec lien vers la page de résultats des tests
- Création d'un script automatisant les tests différentiels (*Differential Testing*)

Statut :

Complété

Tests et Validation

Travaux réalisés :

- **Tests syntaxiques Deca** : Début de l'implémentation des tests syntaxiques ; une partie est déjà en place
- **Tests contextuels Deca** : Avancement significatif sur les tests contextuels, notamment :
 - Cas invalides
 - Erreurs de typage
- **Tests unitaires / Couverture** : Écriture de tests unitaires ciblés pour augmenter la couverture de code
- **Couverture JaCoCo actuelle** : environ **28%**
- Définition d'une stratégie de tests pour le compilateur Deca sans objet basée sur deux approches :
 - *Metamorphic Testing*
 - *Differential Testing*
- Rédaction des tests pour la génération de code selon la hiérarchie de répertoires Git

Éléments en cours :

- Complétion des tests syntaxiques

Statut :

En cours

État d'avancement – Étape C : Génération de code (Finalisation)

Travaux réalisés :

- Réimplémentation de `codeGenPrint` pour couvrir le cas d'affichage des `float` en hexadécimal
- Terminaison du gestionnaire d'erreurs runtime

Statut :

Complété (partie sans objet)

État d'avancement – Partie Orientée Objet

Travaux réalisés :

- **Table des méthodes** : Implémentation complète de la table des méthodes

- Construction, pour chaque classe, de sa table d'étiquettes
- Utilisation de la structure `TreeSet` de Java
- Indexation des méthodes via l'index obtenu à l'étape B
- Gestion des méthodes héritées et redéfinies (évitements des doublons grâce à l'index unique)
- **Initialisation des classes** : Implémentation des `init` pour les classes selon la logique du polycopié :
 - Initialisation des attributs de la sous-classe à leurs valeurs par défaut
 - Appel du `init` de la super-classe
 - Initialisation des attributs avec leurs valeurs spécifiées
- **Expression new** : Implémentation de la génération de code pour l'expression `new`
 - Utilisation de l'instruction assembleur `NEW`
 - Appel de la méthode `init` de la classe instanciée

Statut :

En cours

Synthèse et décisions

Étape	Avancement	Statut
Infrastructure CI/CD	Pipelines et scripts automatisés	Complété
Tests & Validation	Tests contextuels avancés, couverture 28%	En cours
Étape C (sans objet)	Génération de code finalisée	Complété
Partie Objet	Table des méthodes, <code>init</code> , <code>new</code>	En cours

Points d'attention :

- Les extensions `case` et `instanceof` n'ont pas encore été traitées
- La couverture de code doit être augmentée (objectif > 50%)
- Finalisation des tests syntaxiques en cours

Prochaines étapes :

- Finaliser l'implémentation de la partie orientée objet (appels de méthodes, héritage)
- Compléter les tests syntaxiques et continuer l'augmentation de la couverture JaCoCo
- Préparer la documentation pour le rendu intermédiaire

Fait le 11 janvier 2026
Équipe 43G8