

MD-2045, CHISINAU, STR. Studentilor, 9/7, TÉL: 022 50-99-08 www.utm.md
D.A.013 TEST SOFTWARE
1. Données sur la discipline/module

Faculté	Ordinateurs, Informatique et Microélectronique				
Département	Ingénierie Logicielle et Automatique				
Degré académique	Licence, 1er cycle				
Programme d'études	0612.3 Science des données				
Année d'études	Semestre	Type d'évaluation	Catégorie formative	Catégorie d'option	Crédits ECTS
III (régime d'études à temps plein);	6 ;	E	D-discipline du domaine professionnel	A - unité de cours optionnelle	4

2. Durée totale estimée

Nombre total d'heures dans le programme	D'où				
	Heures auditives		Travail individuel		
	Cours	Td/tp	Projet de l'année	L'étude du matériel théorique	Préparation des candidatures
120	30	30	-	30	30

3. Prérequis pour accéder à l'unité de cours/module

Selon le plan d'éducation	Structures de données et algorithmes, mathématiques discrètes, programmation C++, analyse et conception d'algorithmes, langages formels et conception de compilateurs.
Selon les compétences	Acquisition par les étudiants des principes de développement d'algorithmes, de langages de programmation, de méthodes et de techniques de formulation de modèles informatiques et de modélisation de divers phénomènes et processus techniques.

4. Conditions de réalisation du processus éducatif pour

Cours	Un projecteur et un ordinateur sont nécessaires pour présenter le matériel théorique en classe. Les retards des étudiants et les appels téléphoniques pendant les cours ne seront pas tolérés.
Td/tp	Les étudiants compléteront des rapports selon les conditions imposées par les indications méthodiques. Le délai de remise des travaux de laboratoire – une semaine après leur achèvement. Pour la soumission tardive du travail, ceci est déduit de 1 pct./semaine de retard.

5. Compétences spécifiques accumulées

Compétences professionnelles	C2. Examiner les aspects organisationnels et informationnels de la sécurité. <ul style="list-style-type: none"> ✓ Identifier et définir les concepts, théories, méthodes et principes de développement de la documentation de test sur la base des entretiens avec les clients et de la spécification des exigences du client. ✓ Expliquer les concepts, les théories et les méthodes de planification et de conception de scénarios de test.
------------------------------	--

	<ul style="list-style-type: none"> ✓ Appliquer des concepts, des théories et des méthodes pour tester la planification et la conception de scénarios. ✓ Maîtrise des concepts, principes et méthodes de conception de tests basés sur la spécification des exigences logicielles.
Compétences professionnelles	<p>C4. Examiner les méthodes et technologies de développement de solutions en matière d'assurance et de contrôle qualité des produits logiciels.</p> <ul style="list-style-type: none"> ✓ Comprendre les étapes du cycle de vie d'un produit logiciel et le rôle de l'assurance et du contrôle qualité au cours de ces étapes ; ✓ Former les capacités pour pouvoir appliquer les méthodes et techniques de conception de tests basées sur la spécification des exigences logicielles ; ✓ Connaissance et compréhension des étapes du cycle de vie des bogues logiciels ; ✓ Comprendre les étapes du processus de test ; ✓ Familiarisation avec les éléments de gestion des tests logiciels.
Compétences transversales	<p>CT2. Identifier, décrire et réaliser les activités organisées en équipe avec le développement des capacités de communication et de collaboration, mais aussi avec la prise de différents rôles en réalisant le projet annuel avec l'utilisation correcte des sources bibliographiques et des méthodes spécifiques, ainsi que leurs soutien.</p>

6. Objectifs de l'unité de cours/du module

L'objectif global	Étudier les méthodes de test et de vérification des programmes, les principes de base des techniques de vérification de l'exactitude des programmes, tout en maintenant un équilibre entre considérations théoriques et pratiques
Les objectifs spécifiques	Avoir des connaissances sur les méthodes de développement de systèmes de tests automatiques, sur les erreurs qui apparaissent dans le processus de développement de produits logiciels ; sur la vérification formalisée des programmes ; les notions de logique dans le processus de vérification de l'exactitude des programmes

7. Contenu de l'unité de cours/du module

Le thème des activités didactiques	Nombre d'heures	
	éducation régulière	éducation à temps partiel
Le sujet des conférences		
T1. ASPECTS INCLUS DANS L'ÉTUDE DES TESTS DE PROGRAMMES. Définition d'une erreur logicielle. Causes des erreurs. Le coût des erreurs. Principes de base des tests logiciels. Le rôle et les objectifs du testeur dans le processus de développement logiciel. Axiomes des tests. Termes et définitions des tests.	6	2
T2. TECHNIQUES DE CONCEPTION DE TESTS EN BOÎTE NOIRE. Tests fonctionnels. La méthode de la boîte noire. Test des classes d'équivalence. Tester les valeurs à partir des limites. Test des tables de décision. Test des états et des transitions. Tester le graphique de cause à effet	6	2
T3. TECHNIQUES DE CONCEPTION DE TESTS EN BOÎTE BLANCHE. Tests structurels. Aspects du test de structures linéaires, alternatives et itératives. Tests de flux de contrôle. Testez les niveaux de couverture. Graphique de flux de contrôle. Test de chemin de base. Numéro cyclomatique. Tests de flux de données. Tests de flux de données statiques. Tests de flux de données dynamiques	4	2

T4. GESTION DES ERREURS ET DÉFAUTS LOGICIELS. Liste des erreurs les plus courantes dans le processus de test de logiciels. Cycle de vie des défauts logiciels. Rapports de défauts efficaces.	4	1
T5. LES ÉTAPES DU PROCESSUS DE TEST. Cycle de tests logiciels (SDLC). Planification et contrôle. Analyse et conception des tests. Mise en place et exécution des tests. Évaluation et signalement des défauts. Activités pour conclure le processus de test.	4	1
T6. MÉTHODOLOGIES ET MODÈLES DE DÉVELOPPEMENT DE PRODUITS LOGICIELS. Méthodologie itérative. Méthodologie cyclique. La méthodologie des écluses. Modèles : Cascade, V, Spirale, Agile, Prototypage. Ingénierie inverse.	2	1
T7. NIVEAUX DE TESTS. Tests unitaires. Tests de modules. Tests d'intégration. Test du système. Tests d'acceptation.	4	1
Total des conférences :	30	10
Le thème des activités didactiques	Nombre d'heures	
	éducation régulière	éducation à temps partiel
Le thème des séminaires		
S1. Spécification des exigences	4	1
S2. Techniques de test en boîte noire	4	1
S3. Techniques de test en boîte blanche	2	1
S4. Rapport de défauts logiciels.	2	1
S5. Tests d'interface graphique	2	
S6. Tests non fonctionnels	1	
Semis total :	15	4
Le thème des activités didactiques	Nombre d'heures	
	éducation régulière	éducation à temps partiel
Le thème des travaux de laboratoire		
LL1. Analyse et spécification des exigences logicielles	6	2
LL2. Conception de tests basée sur des techniques de boîte noire	6	2
LL3. Conception de tests basée sur des techniques de boîte blanche	4	2
LL4. Description et déclaration des défauts	4	2
LL5. Outils de tests	4	
LL6. Outils de tests	2	
LL7. Automatisation des tests	4	
Travail total en laboratoire :	30	8

8. Références bibliographiques

Principal	1. Balanescu T. ; "Correction des algorithmes", Maison d'édition technique, Bucarest, 1995.
-----------	---

	<ol style="list-style-type: none"> 2. Marnie L. Hutcheson, <i>Fondamentaux des tests logiciels : méthodes et mesures</i>, John Wiley & Fils, 2003, 408 p. 3. Lee Copeland, <i>Un guide du praticien sur la conception de tests logiciels</i>, Maison Artech, 2004, 294 p. 4. Ron Patton, <i>Tests de logiciels</i>, Éditions Sams, 2005, 408 p. 5. Glenford J. Myers, <i>L'art du test de logiciels</i>, John Wiley et fils, 2004. 6. Myers G. Fiabilité du logiciel. – M. : Mir, 1980. 7. Gris D. La science de la programmation. – M. : Mir, 1984 8. Tursky V. Méthodologie de programmation. M. : Mir, 1981. 9. Joukov D.Yu. Méthodes de vérification du programme. 2001. 10. Joukov D.Yu. Méthodes de vérification du programme. 2001. 11. Orlov S. Technologies de développement de logiciels. Manuel - Saint-Pétersbourg : Peter, 2002.- 464 p. 12. Pocatilu P. Coûts des tests logiciels. Maison d'édition ASE, Bucarest, 2004.
plus	<ol style="list-style-type: none"> 1. Hazfield R. Kirby L. et al. L'art de la programmation en C. Algorithmes fondamentaux, structures de données et exemples d'application. Encyclopédie du programmeur. K. : Maison d'édition « DiaSoft », 2001. – 736 p. 2. Andrew Hunt, David Thomas. <i>Tests unitaires pragmatiques en C#</i>. 3. Jeff Tian. <i>Ingénierie de la qualité logicielle</i>. John Wiley et fils, 2005 4. Hung Q.Nguyen, <i>Tester des applications sur le Web</i>, John Wiley et fils, 2001.

9. ÉVALUATION

flux		Projet de l'année	examen final
Attestation 1	Attestation 2		
30%	30%		40%
Norme de performance minimale			
Présence et activité aux cours et aux travaux de laboratoire ; Obtenir la note minimale de « 5 » dans chacun des certificats et travaux de laboratoire ; Obtention de la note minimale de « 5 » dans le projet de l'année ; Démonstration dans l'épreuve finale de la connaissance des conditions d'application des procédures de modélisation constructive.			

10. Utilisation de l'IA générative

Permission d'utilisation	<p>Utilisation de l'IA générative dans le cadre des travaux et projets</p> <p>L'utilisation de l'IA générative est autorisée, sous réserve que les étudiants respectent les règles suivantes :</p> <ul style="list-style-type: none"> - L'IA générative peut être utilisée pour générer des idées, des structures de texte ou du code, mais tous les matériaux générés doivent être révisés et ajustés par l'étudiant afin de garantir qu'ils répondent aux exigences académiques. - • Toute utilisation de l'IA générative doit être déclarée dans la section des annexes de chaque travail, en utilisant la phrase suivante : "Lors de la préparation de ce travail, l'auteur a utilisé [NOM DE L'OUTIL / SERVICE] dans le but de [MOTIF]. Après avoir utilisé cet outil/service, l'auteur a révisé et édité le contenu comme nécessaire et assume l'entière responsabilité du contenu du travail."
Restrictions d'utilisation	<p>Règles concernant l'utilisation de l'IA générative</p> <ul style="list-style-type: none"> - Les étudiants ne doivent pas considérer l'IA générative comme une source fiable d'informations, car elle ne fournit pas de références claires ni de sources documentées.

- | | |
|--|---|
| | <ul style="list-style-type: none">- Il est interdit de citer directement le contenu généré par l'IA dans les travaux académiques comme s'il s'agissait d'une source primaire.-<ul style="list-style-type: none">• Les activités dans lesquelles l'utilisation de l'IA générative est interdite sont spécifiées par l'enseignant et concernent généralement les évaluations intermédiaires et finales, ou celles qui ne nécessitent pas le développement de compétences professionnelles. |
|--|---|