|  |
| --- |
| Al-Farabi Kazakh National University Mechanic-mathematical faculty **Educational program on a specialty «6D060200»**Elective courses - **«Software development on the basis of templates and models»**SYLLABUSAutumn semester 2017-2018 an academic year |
| **Code of discipline** | **Name of the discipline** | Type | Number of hours per week | The volume of the credits | ECTS |
| Lectures | Seminar | Labor.work |
| FMRPO 8305 | «Software development on the basis of templates and models» | ОК | 2 | 1 | 0 | 3 |  |
| **List of prerequisite** | "Discrete mathematics", "Programming on algorithmic languages", "Algorithms and Data Structures", Software engineering, Algorithms theory, Programming, Fundamentals of Applied Math statistics. |
| Lector's name | Shakenov K.K.,Doct. phys.-math. sci), professor (part-time) | Office hours |  |
| **e-mail** | shakenov@mail.ru |
| **Telephone:** | 221-15-77 | auditory | 224 |
| **Course description**  | Assignment of this discipline is training of the highly qualified specialists owning skills of application of the modern information technologies in the sphere of professional area. |
| **The purpose of the course** | The purpose of discipline is to develop doctoral students in understanding of the principles and concepts that underpin the development of algorithms, arming them with the knowledge and skills sufficient to design and implement the tasks using fast and efficient modern algorithms. |
| **The learning outcomes** | Doctoral students will be able to have an idea about the different ways of construction and analysis of efficient algorithms, evaluate complexity of algorithms and its dependence on the selected data structure. |
| **References** | **Basic:**1. Royce W. W. Managing the Development of Large Software Systems // Proceedings of the 9th International Software Engineering Conference. Computer Society Press, 2013. Ñ. 328–338.
2. Model Driven Architecture (MDA). ormsc/2011-07-01: OMG, 2011.
3. Kruchten P. The Rational Unified Process: An Introduction. Addison-Wesley Longman Publishing Co., Inc. 2013.
4. Pages B. BoUML user manual. 2010. http:/bouml.free.fr.
5. Holzmann G. Spin Model Checker, The Primer and Reference Manual. Addison Wesley, 2003.
6. OMG Unified Modeling Language (OMG UML), Superstructure, v2.1.2. Object Management Group (OMG), 2007. http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF.
7. Крэг Ларман. Применение UML и шаблонов проектирования: Введение в объектно-ориентированный анализ и проектирование. М. - Санкт-Петербург- Киев. 2011. -Издательский дом "Вильямс. - 496 с.
8. А.М.Вендров. CASE - технологии. Современные методы и средства проектирования информационных систем // http://www.interface.ru/logworks/caset/glava4/glava4\_1.htm.
9. Крылов Е. Rational Rose98- новое CASE - средство для Oracle8 // <http://www.interface.ru>.
10. Швецов А.Н. Агентно-ориентированные системы: от формальных моделей к промышленным приложениям // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы», 2008. – 101 с. – <http://window.edu.ru/window/library?p_rid=56179>.
11. Швецов А.Н. Концептуальные основы метаметодологии построения мультиагентных интеллектуальных систем // Информатизация процессов формирования открытых систем на основе СУБД, САПР, АСНИ и систем искусственного интеллекта: Материалы 5-й межд. научно-техн. конф. – Вологда: ВоГТУ, 2009. – С. 342 – 345.
12. Швецов А.Н., Сергушичева М.А. Проектирование прикладных мультиагентных систем с использованием пакета DISIT / Информационные технологии. – 2009, №8. – С. 54-60.
13. J.-R. Abrial. The B-Book. Assigning programs to meanings. Cambridge, University Press. 1996.
14. M.Atkinson, et al. The Object-Oriented Database System Manifesto. Proceedings of DOOD’89.
15. Berg K., Kalinichenko L.A. Modeling facilities for the component-based software development method. Proceedings of the Third International Workshop ADBIS’96, Moscow, September 1996.
16. Brioukhov D., et al. Interoperable information systems: architectures and technologies. DBMS, N4, 1995 (in Russian).
17. The Common Object Request Broker Architecture, OMG Document 91.12.1.
18. CORBA 2.0 http://www.omg.org/corba/corbiiop.htm.
19. Kalinichenko L.A. Emerging semantic-based interoperable information system technology. Proceedings of the International Conference {Computers as our better partners, Tokyo, March 1994. World Scientific.
20. Kalinichenko L.A., Kogalovsky M.R. OMG standards: interface definition language IDL in the CORBA arcitecture. DBMS, N 2, 1996 (in Russian).
21. Kalinichenko L.A., Kogalovsky M.R. Interoperability of brokers in the CORBA 2.0 standard.DBMS, N 3, 1996 (in Russian).
22. K. Lano, H. Haughton (Eds) Object-oriented specification (case studies) Prentice Hall, 1994
23. K.Lano. The B language and method. Springer Verlag. 1996.
24. Methodological Support of RFBR projects for the development of open, interoperable information and computing resources for fundamental science: http://www.ipi.ac.ru/synthesis/IISTech/ObjTech/.
25. C. Morgan, T. Vickers. On the refinement calculus. Springer Verlag, 1992.
26. Caroll Morgan. Programming from Specifications. Prentice Hall, 1994.
27. Ontolingua. http://ontolingua.stanford.edu.
28. J.M.Spivey. The Z Notation (A reference manual). Prentice Hall, 1989.
29. S.Stepney, R.Barden, D.Cooper (Eds), Object Orientation in Z. Springer Verlag, WiC series, 1992.
30. M.Stonebraker, et. al. Third generation database system manifesto. The committee for advanced DBMS function. April 1990, Memorandum No. UCB/ERL M90/28, College of Engineering, University of California, Berkely.
31. Linda Wills, Ph. Newcomb (Eds). Reverse Engineering. Kluwer Academic Publishers, 1996.
32. J.B. Wordsworth. Software engineering with B. Addison Wesley. 1996.

**Additional**1. Дж. Макконел. Основы современных алгоритмов - М., Техносфера, 2004
2. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы - М., Вильямс, 2001
 |
| **Organization**the course | This is an introductory course that will be carried out general knowledge with a large amount of theoretical material, so in preparation for the discipline of a significant role for the textbook and laboratory assignments.  |
| Requirements the course | The study of doctoral discipline should• gain an understanding of how to design and analysis of efficient algorithms,• Explore methods of software development and utilization strategy,• learn to assess the complexity of algorithms and its dependence on the selected data structure. |
| **Policy on assessment** | **Description of independent work**  | **The weight** | The learning outcomes |
| LectureLaboratory workIWST | 25%50%25%100% | 1,2,34,5,62,3,44,5,61,2,3,4,5,6 |
|

|  |  |  |  |
| --- | --- | --- | --- |
| Alphabetic estimation | Digital estimation (GPA) | Points (%) | Traditional estimation |
| А | 4 | 95-100 | «Excellent» |
| А- | 3,67 | 90-94 |
| В+ | 3,33 | 85-89 | «Good» |
| В | 3 | 80-84 |
| В- | 2,67 | 75-79 |
| С+ | 2,33 | 70-74 | «Satisfactorily» |
| С | 2 | 65-69 |
| С- | 1,67 | 60-64 |
| D+ | 1,33 | 55-59 |
| D | 1 | 50-54 |
| F | - | 0-49 | «Unsatisfactorily» (chanceless estimation) |
| I(Incomplete) | - | - | «The discipline isn't complete»(it is not considered at GPA calculation) |
| P(Pass) | - | - | «Pass»(it is not considered at GPA calculation) |
| NP (No Pass) | - | - | «No Pass»(it is not considered at GPA calculation) |
| W(Withdrawal) | - | - | «Refusal of discipline»(it is not considered at GPA calculation) |
| AW(Academic Withdrawal) |  | - | Removal from discipline for the academic reasons(it is not considered at GPA calculation) |
| AU (Audit) | - | - | «The discipline is heard»(it is not considered at GPA calculation) |
| Get a credit |  | 30-60 50-100 | Get a credit |
| Not get a credit |  | 0-29 0-49 | Not get a credit |
| R(Retake) | - | - | Repeated studying of discipline |

 |
| **Academic policy course** | All work must be performed and defend within a specified time. Students who do not pass a regular job or received for its implementation less than 50% of points, have the opportunity to work on additional specified job schedule. Students who miss laboratory classes for a good reason, and spend their extra time in the presence of a laboratory, after approval of the teacher. Students who fail to meet all kinds of work, for the exam are not allowed. In addition, when evaluating based on activity and attendance of students during class. Be tolerant and respect the opinions of others. Objection was formulated in the correct form. Plagiarism and other forms of unfair work unacceptable. Unacceptable prompting and cheating while putting CDS intermediate and final control exam, copying solved problems by others, passing the exam for another student. Student convicted of falsification of any information rate, unauthorized access to the Intranet using cribs will receive a final evaluation of «F». For advice on the implementation of independent works (CPC), their delivery and protection, as well as additional information on the material covered and all the other issues that arose in Reading courses, contact the instructor during his office hours. |

STRUCTURE, VOLUME AND CONTENT of THE DISCIPLINE

|  |  |  |
| --- | --- | --- |
| **Week** | **Subject: «6D060200» «Software development on the basis of templates and models» - 3 credits** |  |
| **Title of topic** | **Hours.** | **Task for Self- preparing** | **Max. mark** |
| **1** | **Lect.1.Introduction to**  Formal methods in the process**Sem 1.**  Formal languages and techniques | **2****1** | Background and development of formal methods for the selection criteria.The specifics of the formal approach to the implementation of the main phase of the technological cycle.Cleanroom Software EngineeringPreparing to Quiz #1. | **2****4****4** |
| **2** | **Lect.2.**  Mathematical foundations of formal methods**.** **Sem.2.**  Formal methods in the process of development of software systems **. Quis#1.** | **2****1** | **2****4****4+10** |
| **3** | **Lect.3**  The semantic network and ontology**Sem.3.**  Predicate calculus, set theory, model theory, category theory | **2****1** | Computational models and frames.OWL - structured ontology specification language | **2****4****4** |
| **4** | **Lect.4.** Algebraic specification**Sem.4**  Algebraic specification of abstract data types | **2****1** | Polysort algebraic systems  | **2****4****4+10** |
| **5** | **Lect.5.**  denotational modeling**Sem.5**  Denotational simulation of dynamics systems/ **Quiz #2.** | **2****1** | Denotational semantics of information objects | **2****4****4** |
| **6** | **Lect.6.**  process Algebra**Sem.6.** Calculus of communicating sequential processes (Communicating Sequential Processes, CSP). | **2****1** | Universal properties of CSP-specifications Preparing to Quiz#2, and Mid-Term exam | **2****4****4+10** |
| **7** | **Lect.7.**  Deductive description**Sem.7.**  Deductive description and a logical approach to programming | **2****1** | The analogy between programs and structural evidence. resolutions and Prolog language method | **2****4****4+10** |
|  | **MID\_TERM EXAM (100) + Evaluation (RK1).**  |  |  | **100** |
| **8** | **Lect.8.**  Kripke structures**Sem.8.**  Modal and dynamic logic | **2****1** | Dependency graph, the operator schemes and Petri nets | **2****4****4** |
| **9** | **Lect.9.**  Mutates (evolutionary) algebra**Sem.9**  Machine of abstract state (MAC)./ **Quiz #3.** | **2****1** | Language AsmL and verification during execution. | **2****4****4** |
| **10** | **Lect.10.**  Structural analysis**Sem.10.** Structural analysis and decomposition of complex systems | **2****1** | Graphical presentation of the results of the structural analysis.Quiz #3 | **2****4****4+10** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **11** | **Lect.11.**  structural analysis**Sem.11.** Structural analysis and decomposition of complex systems | **2****1** | Examples of different encoding techniques. | **2****4****4** |
| **12** | **Lect.12.**  Object-oriented approach**Sem.12.**  Object-oriented approach to analysis and design / **Quiz #4.** | **2****1** | F-logic - the device logical formalization of the process of object-oriented design. | **2****4****4** |
| **13** | **Lect.13.**  Component design **Sem.13.**  Component design and architecture description languages. | **2****1** | The principles of the component approach. Examples of architecture description languages: Wright, Rapide. | **2****4****4** |
| **14** | **Lect.14.**  Formal methods and quality assurance**Sem.14.**  Formal methods and quality assurance of software systems | **2****1** | The formalization of the effectiveness of non-functional requirements | **2****4****4** |
| **15** | **Lect.15.**  Optimization of labor**Sem.15.** Optimize labor costs associated with the use of formal methods | **2****1** | Models and algorithms design patterns, Quiz #4 andpreparing to final exam. | **2****4****4+10** |
|  | **Evaluation (RK2)** |  |  **See explanations below:** | **100** |
|  | **FINAL EXAM**  |  |  | **100** |

**Dean of the Faculty Bektemesov M.A.**

**Chairman of the the**

**methodical bureau Gusmanova F.R.**

Department's chief Urmashev B.A.

Lector Shakenov K.K.