Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of computer systems software

EMPIRICAL METHODS OF SOFTWARE ENGINEERING

Working program of the academic discipline (Syllabus)

Details of the academic discipline	
Level of Higher Education	***First (bachelor's)***
Field of Study	*12 Information Technologies*
Specialty	*121 Software Engineering*
Education Program	*Software Engineering of Multimedia and Information Retrieval Systems*
Type of Course	*Selective*
Mode of Studies	*full-time*
Year of studies, semester	*4th year, spring semester*
ECTS workload	*Lectures: 36 hours, computer workshop: 18 hours, independent work: 66 hours.*
Testing and assessment	*Assessment, modular control work, calendar control*
Course Schedule	*According to the schedule for the spring semester of the current academic year (rozklad.kpi.ua)*
Language of Instruction	*English*
Course Instructors	Lecturer: *Ph.D., Associate Professor, Onai Mykola* Practical training: *Ph.D., Associate Professor, Onai Mykola*

Outline of the Course

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

The study of the discipline "Empirical Methods of Software Engineering" allows students of higher education to develop the competencies necessary for solving complex problems of professional activity related to the development of software systems that use generators of pseudo-random numbers.

***The aim of** the study of the discipline "Empirical Methods of Software Engineering" is the formation of students' ability to conduct scientific and innovative activities related to the development of software systems that use generators of pseudo-random numbers.*

***The subject of** the discipline "Empirical Methods of Software Engineering" is the methods of generating pseudo-random numbers with a given distribution law.*

*The study of the discipline "Empirical Methods of Software Engineering" strengthens the formation of students **of professional competences (FC)** necessary for solving practical tasks of professional activity:*
***PC08** Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.*
***PC15** Ability to apply fundamental and interdisciplinary knowledge to build advanced retrieval algorithms.*

*The study of the discipline "Empirical Methods of Software Engineering" contributes to the formation in students of the following **program learning outcomes** (PRL) according to the educational program:*
***PLO01** To analyze, purposefully search for and select for the information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology.*

***PLO25** To know and to be able to use fundamental mathematical tools to build algorithms and develop modern software.*

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

The successful study of the discipline "Empirical Methods of Software Engineering" is preceded by the study of the discipline "Algorithmic support of multimedia and information-search systems" of the bachelor's training plan in the specialty 121 Software engineering.

The theoretical knowledge and practical skills obtained as a result of mastering the discipline "Empirical Methods of Software Engineering" can be useful for conducting scientific research and completing bachelor's qualification work.

3. Content of the academic discipline

The discipline "Empirical Methods of Software Engineering" involves the study of topics:

Topic 1. Methods of generating uniformly distributed pseudorandom numbers

Topic 2. Methods of generating pseudorandom numbers of a given distribution

Topic 3. Empirical criteria for assessing the degree of randomness of a sequence of numbers

Modular control work

Test

4. Educational materials and resources

Basic literature:

1. *Michael Felderer, Guilherme Horta Travassos Contemporary Empirical Methods in Software Engineering, 2020.*
2. *Ronald L. Graham, Donald E. Knuth and Oren Patashnik Concrete Mathematics: A Foundation for Computer Science Second Edition*

Additional literature:

1. *Donald E.Knuth The art of computer programming. – 2007.*
2. *Lehmer, D.H. (1949). "Mathematical methods in large-scale computing units".* Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery*: 141–146. (journal version: Annals of the Computation Laboratory of Harvard University, Vol. 26 (1951)).*
3. *Henry S. Warren, Jr. Hacker's Delight, Second Edition – Addison-Wesley, 2012. – 512 p.*
4. *George* Marsaglia *Computer science and statistics : proceedings of the Sixteenth Symposium on the Interface, Atlanta, Georgia, March 1984 / edited by L. Billard.*
5. *Eichenauer, Jiirgen, and Jiirgen* Lehn *(*1986*), A non-linear congruential pseudo random number generator,* Statistische Hefte 27*, 315-326*
6. *M.D.* MacLaren *AND G.* Marsaglia*, Uniform* random number generators*, J. Assoc. Comput. Mach.*
7. *Rudolf Lidl, Harald Niederreiter Finite Fields (Encyclopedia of Mathematics and its Applications, Series Number 20) 2nd Edition – Cambridge University Press; 2nd edition (June 19, 2008) – 772 p.*

Educational content

5. Methods of mastering an educational discipline (educational component)

No.	*Type of training session*	*Description of the training session*
Topic 1. Methods of generating uniformly distributed pseudorandom numbers		

1	*Lecture 1. Basic provisions and iterative formulas of the linear congruent method*	*Uniform distribution law. Method of John von Neumann. Knuth's method. Lemaire scheme. Mersenne numbers as a modulus. Module selection. Varieties of the linear congruent method.* *Literature: additional [1] - p. 150, 214 – 241.* *Task on SRS: to analyze the iterative formulas of the linear congruent method.*
2	*Computer workshop 1*	*Task: To develop a program for generating pseudorandom numbers using the linear congruent method.*
3	*Lecture 2. The length of the period of the linear congruent method, its cryptoresistance and the quadratic congruent method*	*Theorem on the maximum length of the period. Theorem on the period of a sequence with a composite module. The concept of the potential of a linear congruent sequence. A linear congruent sequence from the point of view of a cryptanalyst. Theorem on the period of the sequence generated by the quadratic congruent method. Theorem on the optimal choice of parameters.* *Literature: additional [1] - p. 231 - 242; [2] - p. 141 - 143; [3] - p. 197 - 217.* *Task on SRS: to propose your own method of finding the length of the period.*
4	*Lecture 3. Polynomial method and Eichenauer-Lehn method*	*Basic elements of the theory of finite fields. A necessary and sufficient condition for the primitiveness of a polynomial. Iterative formula of the polynomial method and selection of optimal parameters. The concept of inverse congruent sequence. Fermat's method of finding a multiplicative inverse element. Euclid's method of finding a multiplicative inverse element. Iterative formula of the Eichenauer-Lehn method.* *Literature: additional [1] - p. 341-372, 431-472; [3] - p. 145 - 149; [7] - p. 3 – 11.* *Task on SRS: consider the features of calculating the multiplicative inverse element in the Eichenauer-Lehn method.*
5	*Computer workshop 2*	*Task: Develop a program for generating pseudorandom numbers using the polynomial method.*
6	*Lecture 4. Other combined methods*	*John Mauchly's method. Mitchell-Moore method. Fibonacci method. Method of McLaren and Marsaglia. Bayes-Durham method. Lüscher's method.* *Literature: additional [1] - p. 511-552; [2] - p. 145 - 146.* *Task on SRS: to consider examples of the combined methods of generating random numbers.*

7	*Computer workshop 3*	*Task: To develop a program for generating pseudorandom numbers using one of the combined methods.*
Topic 2. Methods of generating pseudorandom numbers of a given distribution		
8	*Lecture 5. Methods of generating a sequence of numbers with the normal distribution law, with the exponential law, the Poisson distribution law and the Erlang distribution*	*Distribution function and probability density of normal distribution and standard normal distribution. The central limit theorem and the method are based on this theorem. The Marsaglia-Bray method. The distribution and probability density function of the exponential distribution law, Poisson's law, and Erlang's law. The physical content of these laws of distribution. The inverse function method and its application for generating a sequence of numbers distributed according to these laws.* *Literature: additional [1] - p. 234-312, 311-354; [9] - p. 39-54.* *Task on SRS: consider the features of generating random numbers subject to Erlang's law.*
9	*Lecture 6. Methods of generating a sequence of numbers with a gamma distribution*	*Complete and incomplete gamma function. Methods of calculating the gamma function. The distribution and probability density function of the gamma distribution. Connection of gamma distribution with other types of distribution. Generation of a pseudo-random number subject to the gamma distribution with an integer parameter α and a fractional parameter α.* *Literature: additional [1] - p. 711 - 725; [7] - p. 47-55.* *Task on SRS: consider examples of the method of generating random numbers with a gamma distribution.*
10	*Computer workshop 4*	*Task: Develop a program for generating pseudo-random numbers forming a gamma distribution.*
11	*Lecture 7. Methods of generating a sequence of numbers with a χ2 distribution*	*Distribution function and probability density of χ2-distribution. The relationship between the χ2-distribution and the gamma-distribution. The relationship between the χ2 distribution and the standard normal distribution law. Methods of generating a pseudo-random number subject to the χ2 distribution.* *Literature: additional [1] - p. 309 – 343.* *Task on SRS: to consider examples of the method of generating random numbers with a χ2 distribution.*
12	*Computer workshop 5*	*Task: Develop a program for generating pseudorandom numbers forming a χ2 distribution.*
13	*Lecture 8*	*Modular control work. Part 1*

14	*Lecture 9. Methods of generating a sequence of numbers with a beta distribution*	*Full and partial beta feature. Methods of calculating the gamma function. The distribution and probability density function of the beta distribution. The relationship of the beta distribution with other types of distribution. Methods of generating a pseudo-random number subject to the beta distribution.* *Literature: additional [1] - p. 414-510.* *Task on the SRS: to consider examples of the method of generating random numbers with a beta distribution.*
15	*Computer workshop 6*	*Task: Develop a program for generating pseudo-random numbers forming a beta distribution.*
16	*Lecture 10. Methods of generating a sequence of numbers with F-distribution and Student's distribution*	*Fisher-Snedecor and Student distribution and probability density function. Connection of the Fisher-Snedecor distribution and the Student distribution with other types of distribution. Methods of generating a pseudo-random number subject to the law of Fisher-Snedekor distribution and Student's distribution. The Marsaglia method.* *Literature: additional [1] - p. 341 - 402.* *The task at the SRS: to analyze the features of using the F-distribution in engineering tasks.*
17	*Lecture 11. Methods of generating a sequence of numbers with a binomial distribution law*	*Distribution function and probability density of binomial distribution. Connection of binomial distribution with other types of distribution. The physical meaning of this law of distribution. Methods of generating a pseudorandom number subject to the binomial distribution law.* *Literature: additional [1] - p. 314 - 390.* *Task on SRS: to consider examples of the method of generating random numbers with a binomial distribution.*
18	*Computer workshop 7*	*Task: Develop a program for generating pseudorandom numbers forming a binomial distribution.*
19	*Lecture 12. Methods of generating a sequence of numbers with a geometric distribution law*	*Distribution function and probability density of geometric distribution. Connection of hemetric distribution with other types of distribution. Methods of generating a pseudo-random number subject to the hemetric law of distribution.* *Literature: additional [1] - p. 69-74; [2] - p. 287-290; [4] - p. 132 - 135.* *Task on SRS: to consider examples of the method of generating random numbers with a geometric distribution.*
Topic 3. Empirical criteria for assessing the degree of randomness of a sequence of numbers		

20	*Lecture 13. Elements of the theory of statistical hypothesis testing, the Pearson test and the Kolmogorov-Smirnov test*	*Simple and complex hypothesis. Errors of the first and second kind. Concept of critical area. Euclidean distance and Pearson distance. Number of degrees of freedom. Determination of the critical point Pearson statistics. Two statistics of the Kolmogorov-Smirnov criterion. The difference between the Kolmogorov-Smirnov criterion and the χ2-criterion. Determination of theoretical and experimental statistics.* *Literature: additional [1] - p. 217-359; [2] - p. 134 - 141.* *Task on SRS: consider examples of errors of the first and second kind.*
21	*Lecture 14. Criterion of uniformity, intervals, series and division*	*Transition from a sequence of real numbers to a sequence of integers. Two schemes of application of the frequency uniformity criterion. The concept of an interval in the context of a sequence of numbers. Two interval criteria. Terms of use of the series criterion. Classical series criterion and its generalization. Marsalli's modification of the series criterion. Stirling numbers of the first and second kind, methods of their calculation. Two types of breakdown criteria and their generalization.* *Literature: additional [1] - p. 435 - 550.* *Task on SRS: consider examples of application of the uniformity criterion.*
22	*Lecture 15. Criteria for collecting coupons and permutations*	*Algorithm for using the coupon collection criterion. Application of Stirling numbers to the coupon collection criterion. Definition of the concept of permutation. Algorithm for finding the permutation number. Algorithm for applying the permutation criterion.* *Literature: additional [1] - p. 214-351.* *Task on SRS: to consider examples of application of the criterion of permutations.*
2 3	*Lecture 16. Monotonicity criterion and maximum criterion*	*The concept of monotonicity in the context of a sequence of numbers. Peculiarities of applying the monotonicity criterion. Connection of the maximum criterion with the Kolmogorov-Smirnov criterion.* *Literature: additional [1] - p. 514 - 611.* *Task on SRS: consider examples of application of the monotonicity criterion.*
24	*Computer workshop 8*	*Task: Develop a program for testing a sequence of numbers for pseudo-randomness by the method specified by the option.*
25	*Lecture 17. Criterion of conflicts, Marsalli and autocorrelation criterion*	*Definition of the concept of confit. Algorithm for counting the number of conflicts. Derivation of the formula for calculating the theoretical number of*

		conflicts. The relationship between the χ2-criterion and the conflict criterion. Application of Stirling numbers in conflict criteria. The connection between the Marsaglia criterion and the placement problem. Algorithm for applying the Marsaglia criterion. Spearman's correlation coefficient. Application of the Spearman correlation coefficient to a shifted sequence of numbers. Theoretical and practical values of the autocorrelation criterion. *Literature: additional [1] - p. 314-415, 514-610.* *Task on SRS: to consider examples of the application of the Marsaglia criterion.*
26	*Computer workshop 9*	*Results*
27	*Lecture 1 8*	*Modular control work. Part 2*

6. Independent work of a student/graduate student

The discipline "Empirical Methods of Software Engineering" is based on independent preparation for classroom classes on theoretical and practical topics.

No. z/p	*The name of the topic submitted for independent processing*	*Number of hours*	*literature*
1	*Preparation for lectures*	*16*	*1-9*
2	*Preparation for a computer workshop*	*27*	*1-9*
3	*Preparation for modular control work. Part 1*	*9*	*1-9*
4	*Preparation for modular control work. Part 2*	*9*	*1-9*
5	*Preparation for the test*	*5*	*1-9*

Policy and Assessment

7. Policy of academic discipline (educational component)

Attending classes. *Absence from a classroom session does not involve the calculation of penalty points, since the student's final rating score is formed solely on the basis of the evaluation of study results. At the same time, discussion of the results of the thematic tasks, as well as presentation / public speaking and participation in discussions and additions at seminars will be evaluated during classroom classes. In order to actively participate in the work of the seminar, the student prepares for a specific seminar class in literature as recommended by the teacher. Participation in the work of the seminar also involves the preparation of reports and co-reports within all classes.*

Missed evaluation control measures. *Every student has the right to make up lessons missed for a valid reason (hospital, mobility, etc.) at the expense of independent work. More details at the link: https://kpi.ua/files/n3277.pdf .*

The procedure for contesting the results of assessment control measures. *A student may raise any issue relating to the assessment procedure and expect it to be dealt with in accordance with pre-defined procedures. Students have the right to challenge the results of control measures with arguments, explaining which criteria they disagree with according to the evaluation. Calendar control is carried out in order to improve the quality of students' education and monitor the student's fulfillment of the syllabus requirements.*

Academic integrity. *The policy and principles of academic integrity are defined in Chapter 3 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". More details: https://kpi.ua/code .*

Norms of ethical behavior. *Standards of ethical behavior of students and employees are defined in Chapter 2 of the Code of Honor of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". More details: https://kpi.ua/code .*

Inclusive education. *The acquisition of knowledge and skills in the course of studying the discipline "Research activity in computer engineering" can be accessible to most people with special educational needs, except for students with serious visual impairments that do not allow them to perform tasks with the help of personal computers, laptops and/or other technical means.*

Studying in a foreign language. *In the course of the tasks, students may be recommended to refer to English-language sources. Assigning incentive and penalty points According to the Regulation on the system of evaluation of learning results, the sum of all incentive points cannot exceed 10% of the rating scale.*

All students must attend lectures and practical classes, where you need to actively work on learning the learning material. For objective reasons (for example - illness, international internship), training can take place in an online form individually upon agreement with the head of the course.

Deadlines and Rescheduling Policy:

Works that are submitted late without good reason will be assigned a lower grade. Rearranging modules takes place with the permission of the dean's office if there are good reasons (for example, sick leave).

Policy on academic integrity *:*

All written works are checked for plagiarism and accepted for defense with correct textual borrowings of no more than 20%. Write-offs during control work are prohibited (including using mobile devices).

8. Types of control and rating system of assessment of learning outcomes

During the semester, students perform 8 computer workshops. The maximum number of points for each computer workshop: 6 points.

Points are awarded for:
- quality of performance of the computer workshop: 0-2 points;
- answer to theoretical questions during the defense of the computer workshop: 0-2 points;
- timely presentation of work for defense: 0-2 points.

Performance evaluation criteria:
2 points – the work is done qualitatively, in full;
1 point - the work is completed in full, but contains minor errors;
0 points – the work is incomplete or contains significant errors.

Answer evaluation criteria:
2 points – the answer is complete, well-argued;
1 point – the answer is generally correct, but has flaws or minor errors;
0 points - there is no answer or the answer is incorrect.

Criteria for evaluating the timeliness of work submission for defense:
2 points – the work is presented for defense no later than the specified deadline;
0 points – the work is submitted for defense later than the specified deadline.

The maximum number of points for performing and defending computer practicals:
6 points × 8 comp. practice = 48 points.

*The assignment for **the modular test** consists of 3 questions - 1 theoretical and 2 practical. The answer to a theoretical question is worth 6 points, and the answer to a practical question is worth 10 points.*

Evaluation criteria for each theoretical test question:
6 points – the answer is correct, complete, well-argued;
5 points – the answer is correct, detailed, but not very well argued;
4 points - in general, the answer is correct, but has shortcomings;
3 points – there are minor errors in the answer;
1-2 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

Evaluation criteria for the practical test question:
9-10 points – the answer is correct, the calculations are completed in full;
7-8 points - the answer is correct, but not very well supported by calculations;
5-6 points - in general, the answer is correct, but has flaws;
3-4 points – there are minor errors in the answer;
1-2 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

The maximum number of points for a modular control work:
*2 papers * (6 points × 1 theoretical question + 10 points × 2 practical questions) = 52 points.*

The rating scale for the discipline is equal to:
$R_c = R_{com.practice} + R_{MKR} = 48$ *points +* 52 *points =* 100 *points.*

Calendar control: is carried out twice a semester as a monitoring of the current state of fulfillment of the syllabus requirements.
At the first certification (7th week), the student receives "passed" if his current rating is at least 50% of the maximum number of points (20 points) that the student can receive before the first certification.
At the second certification (13th week), the student receives "passed" if his current rating is at least 50% of the maximum number of points (35 points) that the student can receive before the second certification.

*Semester control: **assessment***

Conditions for admission to semester control:
With a semester rating (R_C) of at least 60 points and the enrollment of all computer practical work, the graduate student receives credit "automatically" according to the table (Table of correspondence of rating points to grades on the university scale). Otherwise, he has to complete the credit control work.

Completion and protection of a computer workshop is a necessary condition for admission to the performance of credit control work.

A graduate student can try to improve his grade by writing a graded test, and his semester marks will be canceled ("hard" grading system).

The composition and evaluation criteria of the assessment test:

***The test task** consists of 4 questions - 2 theoretical and 2 practical. The answer to each theoretical and practical question is evaluated by 25 points.*

Evaluation criteria for each theoretical test question:
24-25 points – the answer is correct, complete, well-argued;
21-23 points – the answer is correct, detailed, but not very well argued;
17-20 points - in general, the answer is correct, but has flaws;
12-16 points – there are minor errors in the answer;
1-11 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

Evaluation criteria for the practical test question:
24-25 points – the answer is correct, the calculations are completed in full;

21-23 points - the answer is correct, but not very well supported by calculations;
17-20 points - in general, the answer is correct, but has flaws;
12-16 points – there are minor errors in the answer;
1-11 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

The maximum number of points for a modular control work:
25 points × 2 theoretical questions + 25 points × 2 practical questions = 100 points.

Table of correspondence of rating points to grades on the university scale :

Scores	Grade
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
Less than 60	Fail
Admission conditions not met	Not Graded

9. Additional information on the discipline (educational component)

The list of questions submitted for module and semester control is in Appendix 1

Work program of the academic discipline (syllabus):

Is designed by Ph.D., Assoc. Prof., Onai M.V.

Adopted by Computer Systems Software Department (protocol № 12, 11 February 2026)

Approved by the Methodical commission of the Faculty of Applied Mathematics (protocol № 7, 23 February 2026)

Appendix 1. List of questions submitted for module and semester control.

Modular control work No. 1 includes the following questions:

Mochli's method and John von Neumann's method for generating a sequence of random numbers.

A linear congruent method for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

A polynomial method for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

The Eichenauer-Lehn method for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

MacLaren-Marsaglia and Bayes-Durham shuffling methods for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

Pearson's agreement criterion.

Kolmogorov-Smirnov agreement criterion.

Frequency uniformity criterion and series criterion.

Interval criterion.

Breakdown criterion.

Criteria for collecting coupons.

Criterion of permutations.

Monotonicity criterion and maximum criterion.

Criterion of conflicts.

Marsaglia criterion and autocorrelation criterion.

Modular control work No. 2 includes the following questions:

Generation of a sequence of real random numbers having a standard normal distribution on the interval [a; b].

Generation of a sequence of real random numbers having a normal distribution on the interval [a; b].

The inverse function method and generation of a sequence of real random numbers having an exponential distribution on the segment [a; b].

Generation of a sequence of real random numbers having a gamma distribution on the interval [a; b] with given real parameters α and β.

Generation of a sequence of real random numbers having a χ2 distribution on the interval [a; b].

Generation of a sequence of real random numbers having a beta distribution on the interval [a; b] with given real parameters α and β.

Generation of a sequence of real random numbers having an F-distribution on the interval [a; b].

Generation of a sequence of real random numbers having a t-distribution on the interval [a; b].

Generation of a sequence of real random numbers having a geometric distribution on the segment [a; b].

Generation of a sequence of real random numbers having a binomial distribution on the interval [a; b] with the specified parameters.

Generation of a sequence of real random numbers having a Poisson distribution on the interval [a; b].

Generation of a sequence of real random numbers having an Erlang distribution on the interval [a; b].

The list of questions submitted for module and semester control.

Modular control work No. 1 includes the following questions:

Mochli's method and John von Neumann's method for generating a sequence of random numbers.

A linear congruent method for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

A polynomial method for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

The Eichenauer-Lehn method for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

MacLaren-Marsaglia and Bayes-Durham shuffling methods for generating a sequence of real numbers having a uniform distribution on the interval [a; b].

Pearson's agreement criterion.

Kolmogorov-Smirnov agreement criterion.

Frequency uniformity criterion and series criterion.

Interval criterion.

Breakdown criterion.

Criteria for collecting coupons.

Criterion of permutations.

Monotonicity criterion and maximum criterion.

Criterion of conflicts.

Marsaglia criterion and autocorrelation criterion.

Modular control work No. 2 includes the following questions:

Generation of a sequence of real random numbers having a standard normal distribution on the interval [a; b].

Generation of a sequence of real random numbers having a normal distribution on the interval [a; b].

The inverse function method and generation of a sequence of real random numbers having an exponential distribution on the segment [a; b].

Generation of a sequence of real random numbers having a gamma distribution on the interval [a; b] with given real parameters α and β.

Generation of a sequence of real random numbers having a χ2 distribution on the interval [a; b].

Generation of a sequence of real random numbers having a beta distribution on the interval [a; b] with given real parameters α and β.

Generation of a sequence of real random numbers having an F-distribution on the interval [a; b].

Generation of a sequence of real random numbers having a t-distribution on the interval [a; b].

Generation of a sequence of real random numbers having a geometric distribution on the segment [a; b].

Generation of a sequence of real random numbers having a binomial distribution on the interval [a; b] with the specified parameters.

Generation of a sequence of real random numbers having a Poisson distribution on the interval [a; b].

Generation of a sequence of real random numbers having an Erlang distribution on the interval [a; b].