

**Federal State Autonomous Educational Institution of Higher Education "Moscow
Institute of Physics and Technology
(National Research University)"**

APPROVED
Vice Rector for Academic Affairs

A.A. Voronov

Work program of the course (training module)

course: Python Programming/Программирование на языке Python
major: Biotechnology
specialization: Biomedical Engineering/Биомедицинская инженерия
Phystech School of Biological and Medical Physics
Chair of Informatics and Computational Mathematics
term: 2
qualification: Bachelor

Semester, form of interim assessment: 3 (fall) - Grading test

Academic hours: 30 AH in total, including:

lectures: 0 AH.

seminars: 0 AH.

laboratory practical: 30 AH.

Independent work: 60 AH.

In total: 90 AH, credits in total: 2

Number of course papers, tasks: 4

Author of the program: T.F. Khiryaynov, senior professor

The program was discussed at the Chair of Informatics and Computational Mathematics 27.04.2022

Annotation

Advantages and disadvantages of Python 3. The Zen Of Python.

The concept of assignment in Python. Variables, values and their types. The concept of dynamic typing. Exchange of two variables values. Tuples of variables.

Arithmetic operations. Exponentiation, integer division is.

The while loop. Cycle control instructions. The nested while loop.

One-pass algorithms: counting, sum, product. The arithmetic mean.

If statement. Cascade conditional elif construction. Logical operations and, or, not in Python.

Single-pass algorithms: search for a number in a stream, maximum.

Number simplicity test. Decomposition of a number into digits. Euclidean algorithm

The str type. Immutability of a string. The length of the string len(s). Naive search for a substring in a string.

String methods find, rfind, count, replace.

Type list. Mutability of the list. Reference data model in Python. == And is operators. Standard functions len, max, min, sum. Slices of lists. Assignment to slice. Methods of a list. Copying list objects.

List comprehensions: generate lists. Array reversal algorithm. Algorithm of cyclic shift in an array.

Function definition. Named function parameters.

The for loop and its features in Python. The range () function.

Sort by inserts. Sorting by choice. Sorting by bubble method.

1. Study objective

Purpose of the course

Teaching students to program in Python at a level sufficient for use in scientific research and in professional activities.

Tasks of the course

- Provide students with a clear understanding of the basics of computer science, including some areas of mathematics (number systems, logic, discrete mathematics);
- to teach students the basic algorithms for processing of numeric and text information;
- to form students ' skills of using Python 3 programming language for solving specific application problems.

2. List of the planned results of the course (training module), correlated with the planned results of the mastering the educational program

Mastering the discipline is aimed at the formation of the following competencies:

Code and the name of the competence	Competency indicators
Gen.Pro.C-2 Use modern IT and software tools to perform professional tasks in compliance with information security requirements	Gen.Pro.C-2.1 Apply modern computing tools and Internet services in professional settings
	Gen.Pro.C-2.2 Apply numerical mathematical methods and use software applications for scientific problem-solving in professional settings
	Gen.Pro.C-2.3 Fulfill basic information security requirements

3. List of the planned results of the course (training module)

As a result of studying the course the student should:

know:

- Fundamentals of the Python programming language;
- software development techniques.

be able to:

- To choose optimal algorithm for the particular problem;
- to develop complete complete Python programs using modern tools for writing and debugging programs;
- to use math packages of Python language to automate the solution of applied and fundamental problems.

master:

- Skill of research of theoretical and applied problems;
- skill of coding of algorithmic problem's solution on Python programming language;
- ways of presenting the results;
- skills of independent work and use of information from knowledge bases in the Internet.

4. Content of the course (training module), structured by topics (sections), indicating the number of allocated academic hours and types of training sessions

4.1. The sections of the course (training module) and the complexity of the types of training sessions

№	Topic (section) of the course	Types of training sessions, including independent work			
		Lectures	Seminars	Laboratory practical	Independent work
1	Basics of Python syntax			4	14
2	Working with collections			6	14
3	Work with files			4	4
4	Functions and working with them			4	6
5	Modules and packages			2	4
6	Single-pass algorithms			4	8
7	Sorting algorithm			6	10
AH in total				30	60
Exam preparation		0 AH.			
Total complexity		90 AH., credits in total 2			

4.2. Content of the course (training module), structured by topics (sections)

Semester: 3 (Fall)

1. Basics of Python syntax

Variables. Expressions. Functions. Conditional statements and loops.

2. Working with collections

Collection methods. Search collections. Conditional expressions and collections. Compare collections.

3. Work with files

Open and close methods. Construction with as. Read and write data in different encodings.

4. Functions and working with them

The default values of the function arguments. Mandatory and non-mandatory arguments. Positional and named arguments.

5. Modules and packages

Connect modules with the import instruction. Various import syntaxes.

Executing the module as a script. Search path. dir()

"Compilation" of modules.

Packages. Import inside the module.

6. Single-pass algorithms

The problem of finding the largest element of the sequence.

7. Sorting algorithm

Sort. The concept of recursion and its application to simple problems.

5. Description of the material and technical facilities that are necessary for the implementation of the educational process of the course (training module)

Computer class with a Board, projector or TV, connected to the network, installed the necessary software, access to educational materials.

6. List of the main and additional literature, that is necessary for the course (training module) mastering

Main literature

1. Программирование на языке высокого уровня Python , учебное пособие для вузов/Д. Ю. Федоров,-Москва, Юрайт, 2020

2. Программирование на языке высокого уровня Python, учебное пособие для вузов / Д. Ю. Федоров. — Москва, Юрайт, 2020.— URL: <https://urait.ru/bcode/454100> (дата обращения: 15.12.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Additional literature

1. Алгоритмы и программы на языках С и PYTHON. Сортировка. Поиск. Строки / В. В. Прут, Москва, МФТИ, 2020

7. List of web resources that are necessary for the course (training module) mastering

<http://scipy.org/>

<http://www.numpy.org/>

<http://www.sympy.org/en/index.html>

<http://scipy.org/scipylib/index.html>

<http://matplotlib.org/>

<http://pandas.pydata.org/>

8. List of information technologies used for implementation of the educational process, including a list of software and information reference systems (if necessary)

manuals from the site <http://acm.mipt.ru>

manuals from the site <http://judge.mipt.ru>

Python documentation <http://python.org>

Package documentation:<http://www.numpy.org/>

<http://matplotlib.org/>

9. Guidelines for students to master the course

Work on the subject includes visiting laboratory classes and independent problems solving.

Independent work includes: reading and taking notes of the recommended literature, viewing Internet resources on the subject of the course, problem solving, preparation for answers to control questions.

Assessment funds for course (training module)

major: Biotechnology
specialization: Biomedical Engineering/Биомедицинская инженерия
Phystech School of Biological and Medical Physics
Chair of Informatics and Computational Mathematics
term: 2
qualification: Bachelor

Semester, form of interim assessment: 3 (fall) - Grading test

Author: T.F. Khiryakov, senior professor

1. Competencies formed during the process of studying the course

Code and the name of the competence	Competency indicators
Gen.Pro.C-2 Use modern IT and software tools to perform professional tasks in compliance with information security requirements	Gen.Pro.C-2.1 Apply modern computing tools and Internet services in professional settings
	Gen.Pro.C-2.2 Apply numerical mathematical methods and use software applications for scientific problem-solving in professional settings
	Gen.Pro.C-2.3 Fulfill basic information security requirements

2. Competency assessment indicators

As a result of studying the course the student should:

know:

- Fundamentals of the Python programming language;
- software development techniques.

be able to:

- To choose optimal algorithm for the particular problem;
- to develop complete complete Python programs using modern tools for writing and debugging programs;
- to use math packages of Python language to automate the solution of applied and fundamental problems.

master:

- Skill of research of theoretical and applied problems;
- skill of coding of algorithmic problem's solution on Python programming language;
- ways of presenting the results;
- skills of independent work and use of information from knowledge bases in the Internet.

3. List of typical control tasks used to evaluate knowledge and skills

Short tests at the begin of each seminars.

4. Evaluation criteria

A sample checklist:

1. The concept of assignment in Python
2. Functions
3. The while loop. Cycle control instructions.
4. If statement. Cascade conditional elif construction.
5. Logical operations in Python.
6. The str type. The length of the string len(s). Immutability of a string.
7. Slices of strings.
8. String methods find, count, replace, startswith, endswith.
9. Reference data model in Python. == And is operators. Copying objects.
10. Slices of lists. Assignment to slice. Methods of a list.
11. List of strings. Split and join methods for a string.
12. The for loop and its features in Python.
13. List comprehensions: generate lists.
14. Two-dimensional arrays (lists of lists). Sub-generation.
15. Type set. Sets and working with them.
16. Tuples and their use.
17. Type dict. Dictionary (associative array) and operations with it.
18. Dict comprehensions: generating sets and dictionaries.

19. Work with files. Methods open, close. Advantages of using with as construction.

20. Modules in Python. Various options for import. Advantages and disadvantages.

Excellent (10) - Exhibited to the student who has shown comprehensive, systematic, deep knowledge of the curriculum of the discipline, showing interest in the subject area, demonstrated the ability to confidently and creatively apply them in practice when solving specific problems, free and correct justification of decisions.

Excellent (9) - is Exhibited to the student who has shown comprehensive, systematic, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, free and correct justification of the decisions made.

Excellent (8) - is Exhibited to the student who has shown comprehensive, systematic, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, the correct justification of the decisions made, with some shortcomings.

Good (7) - is Exposed to the student, if he firmly knows the material, competently and essentially States it, is able to apply the knowledge in practice, but not competently substantiates the results.

Good (6) - is Exposed to the student, if he firmly knows the material, competently and essentially presents it, is able to apply the knowledge in practice, but admits in the answer or in solving problems some inaccuracies.

Good (5) - Exhibited to the student, if he basically knows the material, competently and essentially sets it out, is able to apply the knowledge in practice, but admits in the answer or in solving problems quite a large number of inaccuracies.

Satisfactory (4) - is Exposed to the student who has shown fragmentary, scattered character of knowledge, insufficiently correct formulations of basic concepts, violations of logical sequence in the presentation of the program material, but at the same time he has mastered the main sections of the curriculum necessary for further training, and can apply the acquired knowledge on the model in a standard situation.

Satisfactory (3) - is Exposed to the student who has shown fragmentary, scattered character of knowledge, admitting mistakes in formulations of basic concepts, violations of logical sequence in the statement of the program material, poorly owns the main sections of the curriculum necessary for further training and hardly applies the received knowledge even in a standard situation.

Unsatisfactory (2) - is Exposed to a student who does not know most of the main content of the curriculum of the discipline, makes gross errors in the wording of the basic principles and is not able to use the knowledge gained in solving typical problems.

Unsatisfactory (1) - is Exposed to a student who does not know the main content of the curriculum of the discipline, makes gross errors in the wording of the basic concepts of the discipline and does not have the skills to solve typical practical problems.

5. Methodological materials defining the procedures for the assessment of knowledge, skills, abilities and/or experience

During the test, students can use the program of the discipline, as well as reference literature, computers, abstracts.