Ministry of Education and Science of Ukraine
National Aerospace University
“Kharkiv Aviation Institute”

Department of Aircraft Control Systems (Dep. 301)

APPROVED:
Guarantor of Educational Program
_______________ A. S. Kulik
«27» 08 2021

WORK PROGRAM OF THE OPTIONAL DISCIPLINE

Object-Oriented Programming

(name of the discipline)

Field of Study: 17 “Electronics and Telecommunication”

Program Subject Area: 173 “Avionics”

Educational Program: Systems of Autonomous Navigation and Adaptive Control of Aircrafts
(code number and the name of specialization)

Level of Qualification: 1st (bachelor degree)

Kharkiv 2021

Work program of the optional discipline "Object-Oriented Programming" is for English-speaking students of training direction 173 "Avionics".

«27» 08 2021, 10 p.

Developer: O.V. Havrylenko, docent of dep. 301, Candidate of Science (Engineering)

____
(sign)

The work program has been examined at the meeting of dep. 301 "Aircraft Control Systems".

Record of proceeding: No. 1 from «27» 08 2021

Head of the department
PhD (Engineering), Candidate of Science ________________ K. Yu. Dergachov
(sign)

1. Course description

Indices		Branch of education, learning direction, education level	Course specification	
			Full-time study	
ECTS credits – 7		Branch of Education: 17 – Electronics and Telecommunication	Professional training area (by choice)	
Modules – 2			Calendar year	
Semantic modules – 3				
Research into the topic ____________ (topic name)		Training Direction: 173 – Avionics	2021-2022	2021-2022
			Semester	
			3	4
Total hours – 210			Lectures	
Academic hours per day for full-time study			32	0
Semester 3			Practice classes	
classroom 3 hrs.	self-study 6 hrs.		–	16
Semester 4			Lab classes	
			32	0
classroom – 1 hrs.	self-study 2 hrs.	Education level: 1st (bachelor degree)	Self-study (unaided work)	
			86 hrs.	44 hrs.
			Credits authorization	
			Pass	Exam

Note: ratio of classroom working and unaided (self-study) work is:
80/130 (under full-time education).

2. Aims and objectives of academic discipline

Learning Aims – mastering of methods and means of object-oriented software development for the design and implementation of control systems, including using the tools of technical vision.

Learning Objectives – studying methods of object-oriented programs development with a graphical user interface for the designing of the of control systems with technical vision, such as engineering calculations, plotting functions, receiving and processing of photo images and video streams.

Learning Outcomes

According to the requirements of the educational-professional program, students should achieve the following **competencies**:

GC1. Ability to abstract thinking, analysis and synthesis.

GC 2. Ability to apply knowledge in practical situations.

GC 3. Ability to communicate in a foreign language.

GC 5. Ability to learn and master a modern knowledge.

GC 6. Ability to search, process and analyze information from various sources.

PCS2. The ability to use science and technology in the profession, to argue the choice of methods for specialized tasks of analysis and synthesis systems of avionics systems.

PCS 3. The ability to implement and use hardware and algorithmic tools to increase the accuracy and reliability of control systems and other qualities of the aircraft.

PCS 7. The ability to determine the composition of the testing equipment necessary for experiments to determine the characteristics and parameters and control system aircraft.

PCS 9. The ability to introduce achievements of domestic and foreign science and engineering, to use innovative experience in avionics

Program learning outcomes:

PLO1. Use different forms of representation of avionics systems and describe their different methods (verbal, graphic, formally), analyze situations that may occur during their operation

PLO3. Use science and technology in the professional activity, to argue the choice of methods for solving specialized tasks of analysis and synthesis of avionics

PLO4. Apply modern technologies for automation of design and construction of information and control systems in the avionics field, be able to create hardware and software to increase the accuracy, reliability of control systems and other qualities of the aircraft

PLO8. To determine the structure and parameters of the test equipment to conduct experiments to determine the characteristics of the instruments and control systems aircraft, parameters their components and products.

PLO14. Preserve and increase moral, cultural, scientific achievements and values of society by understanding the history and patterns of development of this domain, its place in the overall system knowledge and the development of society, techniques and technologies, use different types and forms of healthy living

Interdisciplinary Relations:

Prerequisites for studying this discipline:

Higher mathematics: calculating the systems of equations, functions research and plotting the graphs construction; vector algebra. Fundamentals of algorithmization and Programming.

The course supports the following courses:
Fundamentals of Databases? Computational methods and Simulation Techniques.

3. Content of the course

Module 1. Using libraries of Python's functions and objects to solve technical control tasks

Semantic module 1. Fundamentals of programming in Python for solving engineering problems in the CS

THEME 1. Introduction to discipline and basic Python syntax

Purpose, subject of study and discipline tasks. Origins of the appearance and development of an object-oriented programming approach. The first program on Python. Interactive programming mode. Programming script mode. Python identifiers. Keywords. Lines and indents, comments. Input user data. Assign variable values. Standard data types. Converting data types. Main operators: arithmetic operators, comparison operators, assignment operators, logical operators. Operator priorities. Built-in and library Python functions: mathematical functions, random numbers functions, trigonometric functions. Mathematical constants.

THEME 2. Branching, loops, functions and sequences in Python

Decision-making. Instruction IF. Instruction IF ... ELSE. ELIF instruction. Embedded IF. Title (header) and body (suits). WHILE loop. Loop FOR. Nested loops. Loop control instructions. UML activity diagrams representing complex algorithms. Definition of function. Call of function Transmission of parameters by reference and by value. Function arguments. Required arguments. Arguments by name. Arguments by default. Returns values from the function. Global and local variables. Python sequences. Lists. Tuples. Built-in sequence processing functions.

Semantic module 2. Use of object-oriented approach and Python libraries for working with images in technical vision CS

THEME 3. Working with class objects, files and developing graphical user interface for visualization of function graphs

Overview of OOP terminology. Defining classes. Creating instances and destroying objects. Access attributes. Hiding data. Built-in attributes of the class. Properties (Getters and Setters). Class inheritance. Methods overload. Basic overloaded methods. Overloading operators. UML class diagrams representing the structure of an object program. File input/output. Open and close files. Read and write files. Programming the Graphical User Interface (Tkinter). Tkinter widgets. Button. Static Text

(Label). Entry field. Checkbox. Canvas for painting. Standard widgets attributes. Geometric widget layout methods: pack (), grid (), place (). Standard dialogs. MATPLOTLIB library for plotting.

THEME 4. Libraries for working with photo and video images

Using Pillow library for image processing. A list of the main modules of the Pillow package. Features of Pillow package. Functions for opening, copying and saving image files. Creating a new image. Getting information about the image. Image Conversion with Pillow methods. OpenCV algorithms for image processing in Python. Downloading and visualization of images and video data. Image conversion using OpenCV methods. Image filtering algorithms in the OpenCV library. Histogram of the distribution of image brightness. Image binarization with different threshold of brightness. Affine and project image transformation. Image corners detection in technical vision systems.

Module 2. Coursework

Semantic module 3. Coursework

THEME 5. Image processing in control systems with technical vision

Implementation of the course project tasks (see Topics of practice classes)

4. Course structure

Semantic modules and topics	Hours					
	full-time					
	total	among them				
		lect	pract	lab	ind	self
1	2	3	4	5	6	7
Module 1						
Semantic module 1. Fundamentals of programming in Python for solving engineering problems in the CS						
THEME 3. Working with class objects, files and developing graphical user interface for visualization of function graphs	47	12	0	12	0	23
THEME 4. Libraries for working with photo and video images	27	4	0	8	0	15
Modular control	2	0	0	0	0	2
Total for semantic module 2	***76***	***16***	***0***	***20***	***0***	***40***
Semantic module 2. Use of object-oriented approach and Python libraries for working with images in technical vision CS						
THEME 3. Working with class objects, files and developing graphical user interface for visualization of function graphs	47	12	0	12	0	23
THEME 4. Libraries for working with photo and video images	27	4	0	8	0	15
Modular control	2	0	0	0	0	2
Total for semantic module 2	***76***	***16***	***0***	***20***	***0***	***40***
Module 2						
Semantic module 3. Coursework						
THEME 5. Image processing in control systems with technical vision	60	0	16	0	0	44
Total for semantic module 3	60	***0***	***16***	***0***	***0***	***44***
Course total	210	**32**	**16**	**32**	**0**	**130**

5. Topics of seminar classes

№ a/o	Topic name	Hours
1	Not appointed	–
	Total hours	–

6. Topics of practice classes

№ a/o	Topic name	Hours
1	Using the OpenCV library for loading images from files	2
2	Image geometric transformations	2
3	Image processing	2
4	Image color conversion	2
5	Image filtration	2
6	Graphic user interface implementation for the image processing application	2
7	Making a report for the course work	2
8	Making course presentation and speech	2
	Total hours	16

7. Topics of laboratory trainings

№ a/o	Topic name	Hours
1	Development of programs for mathematical calculations in Python	4
2	Development of structured programs for branching and repetition	4
3	Structuring programs and working with sequences	4
4	Realization of a class and work with objects	4
5	Development of graphical interface for calculation tasks and plotting	8
6	Development of graphical interface for downloading and processing raster images	8
	Module 1 lab classes total	32

8. Self-study (unaided works)

№ a/o	Topic name	Hours
1	Theme 1. Install Python interpreter, study documentation for standard libraries	25
2	Theme 2. Working with one-dimensional and multidimensional arrays, studying library documentation numpy	21
3	Theme 3. Elements of the graphical user interface, studying documentation for libraries Tkinter, matplotlib	15
4	Theme 4. Connect Pillow library, study library documentation. Connect OpenCV library, study library documentation	25
5	Theme 5. Examining documentation for the OpenCV library, running and analyzing script examples, experimenting with different methods settings	44
	Total hours	130

9. Individual assignments

№ a/o	Topic name	Hours
1	Not appointed	–
	Total hours	–

10. Teaching methods

Lectures delivering, laboratory training reports submission, individual consultations (if necessary), independent work of students with the tutorials books, on-line documentation.

11. Forms of control

Current test points that score submitted lab reports and individual assignments, evaluation (grades) of semantic topics, final examination.

12. Appointment of grade points obtaining by a student (credit points)

Semester 3

Components of the study work	Grades for one lesson (task)	Number of lessons (tasks)	Total grades
Semantic modulus 1			
Lecture activity	0…1	8	0…8
Laboratory work implementation and report submission	2...10	3	0...30
Current tests	2..5	1	0..5
Modular tests	0…7	1	0…7
Semantic modulus 2			
Lecture activity	0…1	8	0…8
Laboratory work implementation and report submission	2..10	3	0..30
Current tests	2..5	1	0..5
Modular and current tests	0…7	1	0…7
Total			**0…100**
	Semester 4		
Semantic modulus 3			
Current tests	0...5	8	0...40
Course work report	0…20	1	0…20
Software demonstration	0…10	1	0…10
Presentation	0…10	1	0…10
Speech	0…10	1	0…10
Answering a questions	0…10	1	0…10
Total			**0…100**

Grades scale: Ukrainian and ECTS

Grades	Marks	
	Examination	Pass
90 – 100	excellent	passed
75 – 89	good	
60 -74	satisfied	
0 – 59	unsatisfied	not passed

13. Methodical support

All methodical support is electronically located on a cloud storage and is open to all users. The author of the developments is the Associate professor of the dept. 301 Havrylenko O.V. Link for viewing and downloading:

https://drive.google.com/open?id=1O8iyos4Y642KrRYVM2iDZBhzNuITYsMI

1. Summary of lectures on discipline " Object-Oriented Programming ". 2021

2. Slides with presentations of lecture materials on discipline Object-Oriented Programming ". 2021

3. Methodical instructions and tasks for laboratory work on discipline " Object-Oriented Programming ". 2021

4. Methodical instructions and tasks for implementation of course work on discipline “Object-Oriented Programming”. 2021

14. Recommended reading

basic

1. Summerfield, Mark. Programming in Python 3 : a complete introduction to the Python language – Pearson Education, Inc. – 2010.
2. Alexander Mordvintsev & Abid K. OpenCV-Python Tutorials Documentation. Release 1 – 2017.

complementary reading

1. Joseph Howse. OpenCV Computer Vision with Python –Packt Publishing – 2013.
2. Jan Erik Solem. Programming Computer Vision with Python – Creative Commons – 2012.

15. Information resources

1. The official Python site: http://python.org
2. Python tutorial: https://docs.python.org/3/tutorial/index.html
3. Pillow documentation: http: //pillow/readthedocs.org/
4. OpenCV documentation: http://docs.opencv.org
5. Python+OpenCV tutorial https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html