

Нысанға-бағытталған бағдарламалауға графикалық интерфейсті құру кезінде модульдерді қолдану ерекшеліктері

¹*АБИШЕВА Айгуль Амантаевна, т.ғ.м., аға оқытушы, Aigul_abisheva@mail.ru,

¹УСПАНОВА Асель Исимхановна, т.ғ.м., аға оқытушы, esim78@mail.ru,

¹Қазақ экономика, қаржы және халықаралық сауда университеті, Қазақстан, 010005, Нұр-Сұлтан, Жұбанов көш., 7,

*автор-корреспондент.

Аңдатпа. Мақалада Python ортасында объектіге бағытталған бағдарламалауға ерекше назар аударылады. Бағдарламалау тілінің сипаттамалық белгілері ерекшеленеді және сипатталады. Ыңғайлы қосымшаны құру үшін бағдарламалау тілінің негізгі модульдері сипатталған. Шын мәнінде, белгілі бір сенімділікпен Python қысқа мерзімді тактикалық мәселелерді шешу үшін де, ұзақ мерзімді Стратегиялық жобаларды жасау үшін де бағдарламалық жасақтаманы жасайтын кез-келген үлкен ұйымда қолданылады деп айтуға болады. Бүгінгі таңда адамзаттың бағдарламалық өнімдерге деген қажеттіліктерін қамтамасыз ету үшін жүздеген бағдарламалау тілдері мен орталары қолданылады. Мақала бағдарламалық қосымшаны жасау кезінде Python платформасының құрылымына, механизміне және қолданылуына арналған.

Кілт сөздер: бағдарламалау тілі, нысанға-бағытталған бағдарламалау, әдіс, класс, модуль, қосымша, бағдарламалар, интерфейс.

Кіріспе

Python – бұл бағдарламаларды жасау үшін де, әр түрлі қолдану салаларында қолданбалы сценарийлер жасау үшін де қолданылатын танымал бағдарламалау тілі. Бұл – портативті, қолдануға оңай және еркін таратылатын тіл. Сізге басқа тілдерде жазылған бағдарламалар мен құралдарды оңай үйлесетін тиімді жобалар жасауға мүмкіндік береді. Тілді әмбебап болғандықтан, әртүрлі салаларда қолдануға болады.

Python – бұл әзірлеушінің өнімділігі мен кодты оқуға бағытталған жоғары деңгейлі бағдарламалау тілі.

Бағдарламалау тілінің стандартты кітапханасы көптеген пайдалы функцияларды қамтиды.

Python құрылымдық, объектіге бағытталған, функционалды, императивті және аспектке бағытталған бағдарламалауды қолдайды. Тілдің дизайны объектіге бағытталған бағдарламалау моделінің айналасында жасалған. Бағдарламалау әртүрлі әдістемелерге мүмкіндік береді, бірақ ол объектілік тәсілге негізделген, сондықтан Python-да объектіге бағытталған бағдарламалау стилінде жұмыс істеу өте оңай және тиімді [1,2].

Python кластарды, функцияларды немесе деректерді бөлек файлға орналастыруға және оларды басқа бағдарламаларда пайдалануға мүмкіндік береді. Мұндай файл модуль деп аталады. Модульден алынған нысандарды басқа модуль-

дерге импорттауға болады. Файл атауы .py кеңейту модулінің атауына қосу арқылы жасалады. Модуль – бұл компоненттер жиынтығы. Модуль – бұл аттар кеңістігі, – namespace және модульдегі барлық атаулар атрибуттар деп те аталады – мысалы, функциялар мен айнымалылар.

Кластар – бұл мәліметтер құрылымы мен процедураларын бір жерде топтастыруға мүмкіндік беретін объектілер. Кластарды дұрыс қолданған кезде олар сіздің жазған кодыңызды оңай, оқылатын және түсінікті етеді.

Атрибут – бұл класс элементі.

Әдіс – бұл класс ішінде орналасқан және белгілі бір жұмысты орындайтын функция.

Кластар барлық мүмкіндіктері бар объектілер болып табылады:

- Мұрагерлік, оның ішінде, көпше түрінде.
- Полиморфизм (барлық функциялар виртуалды).
- Инкапсуляция (екі деңгей – жалпыға қолжетімді және жасырын әдістер мен өрістер).
- Нысанның өмірлік циклін басқаратын арнайы әдістер: конструкторлар, деструкторлар, жады таратқыштар.
- Кластық және статикалық әдістер, сынып өрістері.

Python-да бағдарламалық жасақтамасы (қосымша немесе кітапхана) модульдер түрінде жасалады, олар өз кезегінде пакеттерге жиналуы

мүмкін [2,3].

Модульдерді каталогтарда да, ZIP архивтерінде де орналастыруға болады.

Модульдер екі түрлі болуы мүмкін: «таза» Python-да жазылған модульдер және басқа бағдарламалау тілдерінде жазылған кеңейту модульдері (extension modules). Модуль жеке файл түрінде, ал пакет бөлек каталог түрінде жасалады.

Модульді бағдарламаға қосуды import операторы жүзеге асырады. Модуль іске қосылғаннан кейін жеке нысанмен ұсынылады және модуль аттар кеңістігіне қол жеткізуге мүмкіндік береді.

Бағдарламаны орындау барысында модульді reload () функциясымен қайта жүктеуге болады.

Нысанға бағытталған парадигманың бірнеше принциптері бар:

- Деректер нысандар түрінде құрылымдалған, олардың әрқайсысының белгілі бір түрі бар, яғни кез-келген класқа жатады;

- Кластар – шешілетін мәселені ресімдеудің нәтижесі, оның негізгі аспектілерін бөліп көрсету;

- Бағдарламадағы нысандар бір-бірімен өзара әрекеттеседі, сұраулар мен жауаптар алмасады. Бұл жағдайда бір типтегі нысандар бірдей сұрауныстарға ұқсас жауап береді;

- Нысандарды неғұрлым күрделі құрылымдармен ұйымдастыруға болады.

Tkinter – бұл терезе интерфейсі бар бағдарламалар жасауға мүмкіндік беретін графикалық кітапхана. Бұл кітапхана танымал бағдарламалау тілінің интерфейсі және tcl/tk графикалық қосымшаларын құру құралдары болып табылады.

Tkinter, tcl/tk сияқты, платформалық кітапхана болып табылады және оны көптеген жалпы операциялық жүйелерде (Windows, Linux, Mac OS X және т.б.) пайдалануға болады.

Зерттеу нәтижелері

GUI құру үшін Tkinter-ді практикалық қолдануы

Бағдарлама қосымшасын құру үшін біз Tkinter графикалық кітапханасын қолданамыз. Import пен from арасындағы басты айырмашылық – import бүкіл модульді бір объект ретінде жүктейді, ал from модульден белгілі бір қасиеттер мен функцияларды жүктейді. From операторының көмегімен аттарды импорттау тікелей импорттаушы модульде импортталған объектінің атын шақырмай-ақ қолданыла алады [3].

Tkinter кітапханасы Python-да стандартты модуль ретінде орнатылған. Бағдарлама қосымшаны Tkinter графикалық кітапханасын іске қосу арқылы бастаймыз (1-суретте көрсетілгендей).

Self – бұл класстың ағымдағы нысанына сілтеме, self арқылы сіз оның ішіндегі атрибуттар мен класс әдістеріне қол жеткізе аласыз және ол кез-келген әдістің алғашқы аргументі. Self әдісін шақырған кезде әдіс шақырылған экземплярға сілтеме беріледі.

Main класы таңдау мәзірі орналасқан бағдарламаның негізгі терезесін жасайды. Бұл әдіс нысан кластан жасалған кезде шақырылады, бұл класқа кластың атрибуттарын баптауға мүмкіндік береді. Main класы 2-ші суретте көрсетілген.

Main_menu әдісі негізгі терезеде PyGame кітапханасымен жүктелген тесті таңдау және фондық музыканы басқару батырмаларын жасайды.

Кластарда _init_ деп аталатын арнайы әдіс бар. _init_ әдісі бір рет шақырылады және бағдарлама ішінде қайта шақырылмайды. _init_ әдісі .png файлында орналасқан, негізгі терезеге жүктелетін қажетті ажыратымдылықты, фонды, қаріптерді (шрифт) орнатады. Класс экземплярін құру кезін-

```
1 import tkinter as tk
2 from os import listdir
3 from PIL import Image, ImageTk
4 from pydub import AudioSegment
5 from pydub.playback import play
6 import localization
7 from localization import get_text as _
8
```

1-сурет – Tkinter кітапханасы

```
1 class Main(tk.Frame):
2     def __init__(self, root):
3         super().__init__(root)
4         self.init_main(root)
5         self.buttons = list()
6     def init_main(self, root):
7         self.root = root
8
9
```

2-сурет – Main класы

де шақырылатын әдіс, ол класс айнымалыларын жариялайды, сондай-ақ, осы әдістің көмегімен басқа да әрекеттерді орындауға болады. `__init__()` әдісін қолдана отырып, класс атрибуты үшін мәндерді тағайындауға болады. 3-ші суретте Main класы `__init__` әдісі көрсетілген.

Батырмалар `button` класының көмегімен жасалады, оның атрибуттарына атау, шрифт, түсі, өлшем және басқа параметрлерде көрсетіледі.

`Pygame.mixer.music.load` әдісі бағдарлама каталогынан тағайындалған аудио файлды жүктейді.

Python-да нысанды құруға және оны жоюға да болады. Ол үшін арнайы әдіс деструктор `_del_` арналған, мысалы.

```
class Main Class(object):
    def __init__(self, name):
        self.name = name
    def __del__(self):
        print()
    {} класс Main Class'.format(self.name))
```

Тест сұрақтары қосымшаның барлық тақырыбы бойынша құрастырылған.

`Open_m`, `open_h`, `open_f`, `open_g` кластары арқылы тест терезелерін құрамыз.

Кластардың бәрі бірдей, олардың әрқайсысында есепті жүргізетін `score` айнымалысы және экранда нәтижені көрсететін `label` элементі бар.

`Score_plus` және `score_minus` әдістері жауаптың дұрыстығына байланысты ұпай санын

өзгертеді.

`Destroy ()` – бұл әмбебап виджет әдісі, яғни біз бұл әдісті кез-келген виджетпен, сондай-ақ, `tkinter` негізгі терезесімен қолдана аламыз.

`Destroy_game` әдісі алдыңғы сұрақтың виджеттерін жояды. Жауап батырмасын басқан кезде орындалатын `clickCorrerctButton` және `clickWrongButton` әдістер ұпайларды өзгертеді және келесі сұрақ әдісін шақырады (4-суретте көрсетілгендей).

Сондай-ақ, модульден суреттері мен жауап нұсқалары бар сұрақтар көрсетіледі. Бір модуль ішіндегі код басқа модульдерден оқшауланған, бұл бағдарлама ішіндегі атаулардың бір-біріне кедергі жасауын азайтады. `PhotoImage` класының көмегімен жүктеледі (5-суретте көрсетілгендей).

Әрбір тест тақырыбы бойынша әр сұрақтың негізгі функциясы мен әдістерін қамтитын тиісті модуль бар. 6-суретте логиканы жеке компоненттерге бөлу арқылы модульдер бағдарламаны құрылымдайтыны көрсетілген.

Бағдарламалық қосымшаның интерфейсі модульдерді қолдана отырып GUI құру үшін `Tkinter` көмегімен жасалған (7-суретте көрсетілген).

Қорытынды

Қорытындылай келе, Python платформасында бағдарламалар қосымшасын құру қиын емес деп айтқым келеді. Бағдарламалық қосымша жасау үшін сіз тілдің синтаксисін білуіңіз керек.

```
9 class Main(tk.Frame):
10     def __init__(self, root):
11         super().__init__(root)
12         # Фон главного окна, Нужно создать канвас для того чтобы сделать фон текста прозрачным
13         self.canvas_pn = tk.Canvas(width=800, height=600)
14         self.canvas_pn.pack()
15         self.bg = tk.PhotoImage(file='img\Bg.png')
16         # позиция на которой будет background.png
17         self.canvas_pn.create_image(400, 300, image=self.bg)
18         # Название программы в главном меню, создал в канвасе чтобы был прозрачный фон!
19         self.lab = self.canvas_pn.create_text(392, 60, text="Welcome to PyTester", font="ljk_Downcome 40", fill="white")
20         # Снизу в углу строка о создателях
21         self.lab2 = self.canvas_pn.create_text(748, 588, text="by IS-324-2019", font="mr_ITCserifGothic 10",
22                                                 fill="white")
23         self.main_menu()
```

3-сурет – Main Класы `__init__` әдісі

```
1 def clickCorrrectButton():
2     destroy_game()
3     level_4() # келесі деңгейге өту
4     self.score_plus() # дұрыс жауапқа балл қосады
5 def clickWrongButton():
6     destroy_game()
7     level_4()
8     self.score_minus()
9
```

4-сурет – Дұрыс жауаптың функциясы

Модульдерді импорттау – питондағы бағдарламалық архитектураның негізі. Python интерфейстері көптеген басқа тілдерге қарағанда басқаша өңделеді және олардың дизайнының

күрделілігінде әртүрлі болуы мүмкін.

Осы материалдан сіз Python жобаларын жасау кезінде әдістер мен кластарды қалай қолдану керектігін білдіңіз.

```
1 self.add_img = tk.PhotoImage(file="img\games\g2.png")
2 imgFrame = tk.Frame(self)
3 imgFrame.pack(pady=30)
4 img_label = tk.Label(imgFrame, image=self.add_img)
5 img_label.pack(side=tk.LEFT)
6
```

5-сурет – PhotoImage класы

```
1 self.question = tk.Label(self, text='?', font="mr_ITC SerifGothic 20")
2 self.question.pack(side=tk.TOP)
3 self.question.config(bg='#1D0C33', fg='white')
4 btn_top = tk.Frame(self)
5 btn_bottom = tk.Frame(self)
6
```

6-сурет – Негізгі сұрақтарды қосу функциясы



7-сурет – Бағдарламалық қосымшаның интерфейсі

ӘДЕБИЕТТЕР ТІЗІМІ

1. Федоров Д.Ю. Основы программирования на примере языка Python: учеб. пособие. М., 2016.
2. Буч Г. Объектно-ориентированный анализ и проектирование / Пер. с англ. – 2-е изд. – М.; СПб.: Бинот: Нев. диалект, 2008. – 558 с.
3. Прохоренко Н., Дронов В. Python 3 и PyQt 5. Разработка приложений. БХВ-Петербург, 2013.

**Особенности использования модулей при создании графического интерфейса
в объектно-ориентированном программировании**

¹*АБИШЕВА Айгуль Амантаевна, м.т.н., старший преподаватель, Aigul_abisheva@mail.ru,

¹УСПАНОВА Асель Исимхановна, м.т.н., старший преподаватель, esim78@mail.ru,

¹Казахский университет экономики, финансов и международной торговли, Казахстан, 010005, Нур-Султан, ул. Жубанова, 7,

*автор-корреспондент.

Аннотация. В статье особое внимание уделено объектно-ориентированному программированию в среде Python. Выделяются и описываются характерные особенности языка программирования. Описаны основные модули языка программирования для создания удобного приложения. Фактически с определенной долей уверенности можно утверждать, что Python используется практически в каждой достаточно крупной организации, занимающейся разработкой программного обеспечения как для решения краткосрочных тактических задач, так и для разработки долгосрочных стратегических проектов. Сегодня сотни языков программирования и сред используются для обеспечения потребностей человечества в программных продуктах. Статья посвящена структуре, механизму и использованию платформы Python при разработке программного приложения.

Ключевые слова: язык программирования, объектно-ориентированное программирование, метод, класс, модуль, приложение, кнопки, интерфейс.

Features of Using Modules when Developing a Graphical Interface in Object-oriented Programming

¹*ABISHEVA Aigul, Mast. Tech. Sci., Senior Lecturer, Aigul_abisheva@mail.ru,

¹USPANOVA Aseel, Mast. Tech. Sci., Senior Lecturer, esim78@mail.ru,

¹Kazakh University of Economics, Finance and International Trade, Kazakhstan, 010005, Nur-Sultan, Zhubanov str., 7,

*corresponding author.

Abstract. The article focuses on object-oriented programming in the Python environment. The characteristic features of the programming language are highlighted and described. The main modules of the programming language for creating a convenient application are described. In fact, it can be argued with a certain degree of confidence that the Python is used by almost every fairly large organization engaged in software development, both for solving short-term tactical problems and for developing long-term strategic projects. Today hundreds of programming languages and environments are used to meet the human needs for software products. The article is devoted to the structure, mechanism and use of the Python platform when developing a software application.

Keywords: programming language, object-oriented programming, method, class, module, application, buttons, interface.

REFERENCES

1. Fedorov D.Yu. Osnovy programmirovaniya na primere yazyka Python: ucheb. posobie. M., 2016.
2. Buch G. Obektno-orientirovannyi analiz i proektirovanie / Per. s angl. – 2-e izd. – M.; SPb.: Binom: Nev. dialekt, 2008. – 558 s.
3. Prokhorenok N., Dronov V. Python 3 i PyQt 5. Razrabotka prilozhenii. BKhV-Peterburg, 2013.